



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΕΤΑΦΟΡΩΝ & ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΥΠΟΔΟΜΗΣ

Επιβλέπων Καθηγητής: Γ. Κανελλαΐδης

Συνεπιβλέπων : Φ.Μερτζάνης

Ονοματεπώνυμο: Θεοφανίδης Αχιλλέας – Μάριος

ΣΧΕΔΙΑΣΜΟΣ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ ΟΔΟΥ ΜΕ ΧΡΗΣΗ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ ΤΥΠΟΥ “SPLINES”

ΙΟΥΛΙΟΣ 2008

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ	3
ΣΥΝΟΨΗ	4
ABSTRACT	5
ΠΕΡΙΛΗΨΗ	6
ΚΕΦΑΛΑΙΟ 1	8
ΕΙΣΑΓΩΓΗ	8
ΚΕΦΑΛΑΙΟ 2	10
ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	10
2.1 ΑΝΑΓΚΗ ΧΡΗΣΗΣ ΑΡΙΘΜΗΤΙΚΩΝ ΜΕΘΟΔΩΝ	10
2.2 ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΙΣ ΚΑΜΠΥΛΕΣ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ	13
2.3 Η ΠΡΟΟΠΤΙΚΗ ΧΡΗΣΗΣ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ ΣΤΗΝ ΟΔΟΠΟΙΑ	15
2.4 ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΤΩΝ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ	16
2.5 ΜΑΘΗΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΩΝ B-SPLINES ΣΤΟ ΣΧΕΔΙΑΣΜΟ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ ΟΔΟΥ	18
2.6 ΕΦΑΡΜΟΓΗ ΤΩΝ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ (B-SPLINES) ΣΤΗΝ ΟΔΟΠΟΙΑ	26
2.6.1 Ακτίνα Οριζοντιογραφικής Καμπύλης	28
2.6.2 Έλεγχος Ακτίνας Καμπυλότητας	29
2.6.4 ΟΡΑΤΟΤΗΤΑ ΚΑΙ ΣΤΟΙΧΕΙΑ ΜΕΛΕΤΗΣ ΤΗΣ	31
ΚΕΦΑΛΑΙΟ 3	43
ΠΑΡΟΥΣΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ “DROMOSSPLINES”	43
3.1 ΛΟΓΙΣΜΙΚΟ “DROMOSSPLINES”	43
3.2 ΠΕΡΙΓΡΑΦΗ & ΛΟΓΙΚΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ «DROMOSSPLINES»	44
3.2.1 ΚΟΡΥΦΕΣ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ	44
3.2.2 ΧΑΡΑΞΗ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ	46
3.2.3 ΚΑΘΟΡΙΣΜΟΣ ΜΟΡΦΟΛΟΓΙΑΣ ΕΛΑΦΟΥΣ	47
3.2.4 ΜΟΡΦΟΛΟΓΙΚΗ ΚΑΤΑΤΑΞΗ ΤΟΥ ΔΡΟΜΟΥ	48
3.2.5 ΈΛΕΓΧΟΣ ΑΚΤΙΝΑΣ ΚΑΜΠΥΛΟΤΗΤΑΣ ΚΑΤΑ ΟΜΟΕ	49
3.2.5 ΈΛΕΓΧΟΣ ΟΡΑΤΟΤΗΤΑΣ ΚΑΤΑ ΟΜΟΕ	50
ΚΕΦΑΛΑΙΟ 4	52
ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΕΣ ΕΞΙΣΩΣΕΙΣ	52
4.1 ΕΞΙΣΩΣΗ ΓΩΝΙΩΝ	52
4.2 ΕΞΙΣΩΣΗ ΤΟΜΗΣ	54
4.3 ΕΞΙΣΩΣΗ ΕΠΙΠΕΔΟΥ	56
4.4 ΕΞΙΣΩΣΗ ΕΓΚΛΩΒΙΣΜΟΥ	58
ΚΕΦΑΛΑΙΟ 5	62
ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ	62
5.1 ΚΑΤΑΣΚΕΥΗ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ	62

5.2 ΜΟΡΦΟΛΟΓΙΚΗ ΚΑΤΑΤΑΞΗ ΤΟΥ ΔΡΟΜΟΥ	68
5.3 ΕΛΕΓΧΟΣ ΑΚΤΙΝΑΣ ΚΑΜΠΥΛΟΤΗΤΑΣ ΚΑΤΑ ΟΜΟΕ	75
5.4 ΕΥΡΕΣΗ ΤΗΣ ΟΡΑΤΟΤΗΤΑΣ ΓΙΑ ΚΑΘΕ ΣΗΜΕΙΟ ΤΟΥ ΔΡΟΜΟΥ ΚΑΙ ΚΑΤΟΠΙΝ ΕΛΕΓΧΟΣ 80	
5.5 ΧΑΡΑΞΗ ΔΙΑΓΡΑΜΜΑΤΩΝ ΟΡΑΤΟΤΗΤΑΣ	84
5.6 ΕΛΕΓΧΟΣ ΠΙΣΤΟΤΗΤΑΣ ΧΑΡΑΞΗΣ ΟΔΟΥ	94
ΚΕΦΑΛΑΙΟ 6	96
ΣΥΓΚΡΙΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ ΧΑΡΑΞΗΣ	96
6.1 ΣΥΓΚΡΙΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ ΧΑΡΑΞΗΣ	96
ΣΥΜΠΕΡΑΣΜΑΤΑ	108
ΒΙΒΛΙΟΓΡΑΦΙΑ	111

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 3.1	45
ΕΙΚΟΝΑ 3.2	46
ΕΙΚΟΝΑ 3.3	47

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 2.1	30
ΠΙΝΑΚΑΣ 2.6.1	33
ΠΙΝΑΚΑΣ 2.6.1	35
ΠΙΝΑΚΑΣ 2.6.3	40

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

ΣΧΗΜΑ 2.1	11
ΣΧΗΜΑ 2.2	14
ΣΧΗΜΑ 2.3	27
ΣΧΗΜΑ 2.4	35
ΣΧΗΜΑ 2.5	37
ΣΧΗΜΑ 2.6	38
ΣΧΗΜΑ 2.7	39
ΣΧΗΜΑ 2.8	42
ΣΧΗΜΑ 3.1	48
ΣΧΗΜΑ 3.2	50
ΣΧΗΜΑ 3.3	51
ΣΧΗΜΑ 4.1	53
ΣΧΗΜΑ 4.2	55
ΣΧΗΜΑ 4.3	59
ΣΧΗΜΑ 4.4	60
ΣΧΗΜΑ 5.1	86
ΣΧΗΜΑ 5.2	88
ΣΧΗΜΑ 5.3	90
ΣΧΗΜΑ 5.4	92

Πρόλογος

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε κατά τη διάρκεια του 10^{ου} εξαμήνου των σπουδών μου στο τμήμα Πολιτικών Μηχανικών, στον Τομέα Μεταφορών και Συγκοινωνιακής Υποδομής του Εθνικού Μετσοβίου Πολυτεχνείου. Σημαντική πηγή έμπνευσης αποτέλεσαν τα μαθήματα των Αεροδρομίων και των Ειδικών Θεμάτων Σχεδιασμού Οδών του 9^{ου} εξαμήνου.

Ευχαριστώ θερμά τον υπεύθυνο καθηγητή κ. Κανελαΐδη για την ανάθεση αυτής της διπλωματικής, καθώς επίσης και τον κ. Φώτη Μερτζάνη, υπό την επίβλεψη του οποίου πραγματοποιήθηκε η εργασία, για την πολύτιμη βοήθειά του και την σημαντική καθοδήγηση που μου προσέφερε.

Αθήνα 2008

Σύνοψη

Στην διπλωματική αυτή εργασία απώτερος σκοπός είναι η δημιουργία ενός λογισμικού σχεδιασμού και ελέγχου Οριζοντιογραφίας με βάση καμπύλες μορφής Splines. Το λογισμικό είναι γραμμένο σε γλώσσα Visual Basic, υπό μορφής πρόσθετου προγράμματος (add-on) στο σχεδιαστικό πρόγραμμα Microstation 95 SE της Bentley Corp.

Αντικείμενα του λογισμικού που αναπτύχθηκε, είναι η χάραξη οριζοντιογραφίας οδού και ο έλεγχός της σε επάρκεια ακτίνας καμπυλότητας και ορατότητας για στάση και προσπέραση με βάση τους κανονισμούς ΟΜΟΕ 2001.

Αποτελείται από 16 υποπρογράμματα όλα γραμμένα σε V.Basic που χωρίζουν ουσιαστικά το πακέτο στις εξής ενότητες.

1. Σχεδιασμός Οριζοντιογραφίας
2. Συσχετισμός της Οριζοντιογραφίας με το μορφολογικό ανάγλυφο.
3. Έλεγχος Ακτίνας Καμπυλότητας κατά ΟΜΟΕ 2001
4. Έλεγχος Ορατότητας για στάση και προσπέραση κατά ΟΜΟΕ 2001
5. Διαγραμματική απεικόνιση Ορατότητας κάθε σημείου.

Το λογισμικό σχεδιάστηκε με σκοπό την κατά το δυνατόν ελάχιστη συμβολή του χρήστη και την ελάχιστη απαίτηση σε υπολογιστική ισχύ.

Abstract

The main objective on this thesis, is the creation of a computer program which designs and tests the Horizontal Alignment of a road by using a given function of the Splines family. The software's code is written in Visual Basic as an "add on" to Microstation 95 SE of Bentley Corporations.

The software can be used to design the Horizontal Alignment of a road and evaluate the Horizontal Radius of Curvature and sight distance (both stopping and passing sight distance) based on the rules of OMOE 2001.

It's consisted of 16 subprograms, all developed in V. Basic which divide the package in five main parts.

1. Horizontiography Design
2. Horizontiography and terrain morphology association
3. Radius of Curvature test (based on OMOE 2001)
4. Visibility of stop and overtake test (based on OMOE 2001)
5. Diagrammatic presentation of Visibility for every head- spot of the road

The software was designed with an effort to minimize the user's interaction and the minimum hardware demands.

Keywords: Spline curves, 2d road design, road visibility check, road design software

Περίληψη

Με τον γεωμετρικό σχεδιασμό των οδών που χρησιμοποιείται σήμερα για την χάραξη των οδών στον χώρο, λόγω της γεωμετρικής και όχι αριθμητικής προσέγγισης της χάραξης αλλά και λόγω της τμηματικής παραγωγής της λύσης της οδού (Οριζοντιογραφία, Μηκοτομή, κτλ.), παρουσιάζονται ασυμβατότητες σε σύγχρονα σχετικά συστήματα και λογισμικά.

Το πρότυπο επικοινωνίας των σχεδίων που έχει επικρατήσει σήμερα, είναι μέσω των συστημάτων σχεδίασης και κατασκευής με χρήση H/Y (cad/cam). Όμως για να παραχθεί ένα αποτέλεσμα διαχειρίσιμο και από τα λογισμικά στατικών υπολογισμών αλλά και από τα Συστήματα Ευφών Μεταφορών (ITS) πρέπει να αποτελείται από αριθμητικά προσδιοριζόμενες μορφές με ενιαίο χαρακτήρα και όχι από ένα σύνολο γεωμετρικών σχημάτων ανηγμένων στον χώρο που παράγεται από τα λογισμικά σχεδιασμού των οδών σήμερα.

Η λύση σε αυτήν την ανάγκη που εξετάζεται σε αυτήν την εργασία είναι η προσέγγιση του σχεδιασμού των οδών με χρήση καμπύλων ελεύθερης μορφής spline . Οι καμπύλες αυτές ενσωματώνονται σε όλα τα σύγχρονα συστήματα σχεδιασμού με χρήση H/Y, άρα ικανοποιείται η βασική προϋπόθεση της συμβατότητας.

. Η πλήρης και άμεση σύγκριση της spline χάραξης με την συμβατική χάραξη δεν είναι θεωρητικά εφικτή, αφού υπάρχει διαφορετική θεώρηση της χάραξης και οι ισχύοντες κανονισμοί, σε ένα μεγάλο μέρος τους δεν μπορεί να εφαρμοστούν στην spline χάραξη. Κάποιες βασικές αρχές όμως, όπως ο περιορισμός της μέγιστης ακτίνας κυκλικού τόξου μπορεί να εφαρμοστεί υπολογίζοντας την καμπυλότητα της καμπύλης spline.

Για να υπολογιστούν τα δεδομένα που χρειάζονται για την επεξεργασία αλλά και για την αξιολόγηση μιας spline χάραξης, στην παρούσα διπλωματική εργασία σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή ηλεκτρονικού υπολογιστή σε γλώσσα προγραμματισμού Microstation Basic η οποία επικοινωνεί με το σχεδιαστικό πρόγραμμα Microstation.

Τα δεδομένα που εισάγονται στην εφαρμογή αυτή είναι στοιχεία για τη μορφολογία του εδάφους της περιοχής, το βήμα του αλγορίθμου Splines, το πλάτος της Οδού και τα σημεία από τα οποία είναι επιθυμητό να διέρχεται η οδός. Αφού εισαχθούν τα παραπάνω στοιχεία παράγεται η οριζοντιογραφία της οδού, γίνεται έλεγχος καμπυλότητας και ορατότητας για στάση και προσπέραση σχεδιάζοντας και τα αντίστοιχα διαγράμματα.

Με χρήση της παραπάνω εφαρμογής η spline χάραξη συγκρίνεται με την συμβατική χάραξη για την περίπτωση μιας πραγματικής εγκεκριμένης αναγνωριστικής μελέτης οδού. Σχεδιάζοντας την spline χάραξη να ταυτίζεται ποιοτικά με την συμβατική χάραξη γίνεται εμφανές ότι στην οριζοντιογραφία είναι δυνατό να επιτευχθεί ποιοτική ταύτιση λύσεων ενώ τα διαγράμματα παρουσιάζουν παρόμοιες τιμές με ουσιαστική διαφορά ότι η μεταβολές της spline είναι καμπυλόγραμμες ενώ της συμβατικής γραμμικές.

Συμπερασματικά φαίνεται ότι η spline χάραξη παρουσιάζει ενδιαφέρον για την χάραξη των οδών γιατί ενώ διευκολύνει και δίνει πολλές δυνατότητες στις εφαρμογές που έπονται της οδοποιίας παρουσιάζει και ποιοτικώς ίδιο αποτέλεσμα με τον γεωμετρικό σχεδιασμό των οδών με συμβατικές μεθόδους. Περαιτέρω έρευνα θεωρείται απαραίτητη, ώστε να διαπιστωθεί σε ποιο βαθμό η spline χάραξη είναι δυνατόν μελλοντικά να αντικαταστήσει την συμβατική χάραξη.

Κεφάλαιο 1

Εισαγωγή

Οι μέχρι τώρα χρησιμοποιούμενες μέθοδοι για το σχεδιασμό οριζοντιογραφίας οδού βασίζονταν κατά κύριο σε σχεδίαση με διαδοχικές ευγραμμίες, κυκλικά τόξα και καμπύλες συναρμογής (κλωθοειδείς καμπύλες).

Στον τομέα της οδοποιίας οι ηλεκτρονικοί υπολογιστές έχουν βοηθήσει πολύ τον μελετητή, βελτιώνοντας τον χρόνο που χρειάζεται για την σχεδίαση μιας χάραξης. Όμως οι αρχές και η μεθοδολογία που χρησιμοποιεί, δηλαδή του γεωμετρικού σχεδιασμού των οδών, είναι η ίδια που θα χρησιμοποιούσε και στον σχεδιασμό της οδού με γραφικά μέσα. Αυτό δημιουργεί κάποια προβλήματα στην χρήση των δεδομένων της οδού σε άλλες εφαρμογές, λόγω της γεωμετρικής και όχι αριθμητικής προσέγγισης της οδού.

Η ανάγκη χρήσης των δεδομένων της οδού σε άλλες εφαρμογές παρουσιάζεται σήμερα περισσότερο από ποτέ γιατί η κατασκευή ενός οδικού τμήματος σήμερα, και ακόμη περισσότερο στο μέλλον, είναι ένα σύνολο έργων που σκοπό έχουν βέλτιστη μεταφορική ικανότητα του έργου αλλά και τις ελάχιστες επιπτώσεις στην κοινωνία και το περιβάλλον. Έτσι κατασκευάζονται μεγάλα τεχνικά έργα όπως γέφυρες και σήραγγες για να μειωθούν οι αποστάσεις, πολλά μικρότερης κλίμακας τεχνικά έργα όπως μικρές γέφυρες και οχετοί για την εναρμόνιση του έργου με το περιβάλλον, αλλά και σχεδιάζονται διάφορα Συστήματα Ευφών Μεταφορών (**ITS, Intelligent Transportation Systems**) για την καλύτερη και ασφαλέστερη αξιοποίηση του έργου. Για όλα αυτά τα σύγχρονα συστήματα παρουσιάζεται σήμερα η ανάγκη, να αναζητηθούν λύσεις που θα οδηγήσουν σε συμβατές εφαρμογές κατά στάδιο της μελέτης.

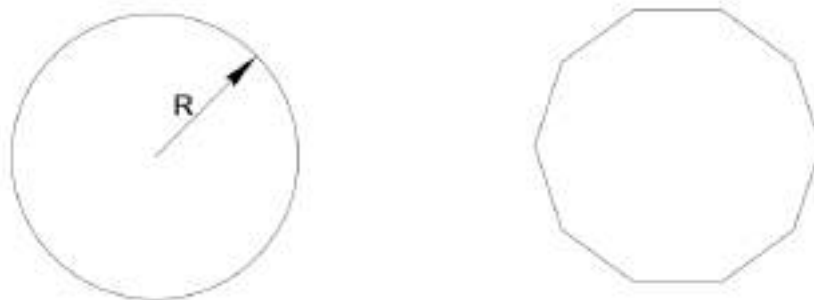
Στο πλαίσιο αυτής της ανάγκης, η παρούσα εργασία ασχολείται με την δυνατότητα χρήσης αριθμητικών μεθόδων για τον σχεδιασμό οριζοντιογραφίας οδών και ειδικότερα με την χρήση των καμπύλων ελεύθερης μορφής (Splines). Οι καμπύλες αυτές τα τελευταία χρόνια έχουν βρει πολλές εφαρμογές σε τομείς της επιστήμης και της τεχνολογίας, όπως στην αυτοκινητοβιομηχανία και την αεροναυπηγική που ήταν οι πρωτοπόροι που χρησιμοποίησαν και ανέπτυξαν τις καμπύλες Splines, αλλά και στην αρχιτεκτονική, τον σχεδιασμό προϊόντων, την τοπογραφία, την γεωλογία, κ.α. Επίσης σε ερευνητικό επίπεδο μελετάται η χρήση τους στην ιατρική, τα πεπερασμένα στοιχεία, την στατιστική, κ.α. Ο λόγος που η χρήση τους άργησε να επεκταθεί ευρέως είναι ότι για τον υπολογισμό τους χρειάζεται μεγάλος αριθμός αριθμητικών πράξεων και έτσι καθίσταται απαραίτητη η χρήση ηλεκτρονικού υπολογιστή για τον προσδιορισμό τους.

Κεφάλαιο 2

Βιβλιογραφική Ανασκόπηση

2.1 ΑΝΑΓΚΗ ΧΡΗΣΗΣ ΑΡΙΘΜΗΤΙΚΩΝ ΜΕΘΟΔΩΝ

Κατά τη συνήθη πρακτική του γεωμετρικού σχεδιασμού των οδών με χρήση σχετικού λογισμικού ηλεκτρονικού υπολογιστή, η οριζοντιογραφία της οδού μελετάται χωριστά από τη μηκοτομή, και στον χώρο μπορεί μόνο να αναχθεί το αποτέλεσμα (η χάραξη), ως ένα σύνολο από μικρά τρισδιάστατα ευθύγραμμα τμήματα. Σε μια τέτοια λογική όσο αυξάνεται η ακρίβεια τόσο αυξάνεται και ο όγκος των δεδομένων αλλά και η πολυπλοκότητα των υπολογισμών. Επίσης ένα ακόμα σύνολο από πληροφορίες που χρειάζονται για την περιγραφή της οδού, όπως οι ακτίνες των κυκλικών τόξων και οι παράμετροι των τόξων συναρμογής, πρέπει να καταγράφονται χωριστά γιατί κατά την μετατροπή της χάραξης σε ευθύγραμμα τμήματα οι πληροφορίες που θα μπορούσαν να είναι αποθηκευμένες στον ίδιο τον άξονα χάνονται. Παραδείγματος χάριν (για να γίνει κατανοητή η απώλεια της πληροφορίας), αν σε ένα πρόγραμμα σχεδίασης σχεδιαστεί ένας κύκλος, τότε το σχεδιαστικό πρόγραμμα αποθηκεύει τις απαραίτητες πληροφορίες για το αντικείμενο, δηλαδή τις συντεταγμένες του κέντρου του και την ακτίνα του. Αν όμως το ίδιο σχήμα χαραχθεί ως ένα σύνολο από ευθύγραμμα τμήματα όπως φαίνεται στο σχήμα 1.1, τότε θα αποθηκευτούν από το πρόγραμμα οι συντεταγμένες τις αρχής και του τέλους κάθε ευθύγραμμου τμήματος και οι πληροφορίες για την ακτίνα και το κέντρο του κύκλου θα χαθούν και δεν θα μπορούν και να ανακτηθούν.



Σχ. 2.1 Σχεδίαση κύκλου δίνοντας ακτίνα και κέντρο και σχεδίαση με ευθύγραμμα τμήματα

Οι εφαρμογές που χρειάζονται μια **ενιαία** και **ακριβή** περιγραφή των οδικών αξόνων και του καταστρώματος της οδού στον **τριδιάστατο χώρο** είναι πολλές και παρουσιάζουν μεγάλο ενδιαφέρον.

Μια πρώτη ομάδα από αυτές της εφαρμογές είναι τα στατικά και γεωτεχνικά προγράμματα ηλεκτρονικού υπολογιστή, τα οποία παρέχουν την δυνατότητα να εισάγονται απ' ευθείας οι καμπύλες επιφάνειες και γραμμές εφόσον αυτές είναι αριθμητικά προσδιορισμένες. Αυτό δίνει την δυνατότητα να επιλύονται πιο γρήγορα αλλά και πιο σωστά, πολύπλοκα στατικά συστήματα όπως γέφυρες πάνω σε κυκλικό τόξο και τόξο συναρμογής κ.α. Εδώ αξίζει να αναφερθεί ότι οι αριθμητικές μέθοδοι προσδιορισμού των επιφανειών και των καμπύλων και η επεξεργασία τους με ηλεκτρονικό υπολογιστή έχουν δώσει την δυνατότητα στους μηχανικούς να σχεδιάσουν αλλά και να κατασκευάσουν τεχνικά έργα που αλλιώς θα ήταν αδύνατο. Ένα γνωστό παράδειγμα εφαρμογής των παραπάνω είναι το μουσείο Guggenheim στο Μπιλμπάο της Ισπανίας που σχεδιάσθηκε εξ' ολοκλήρου με επιφάνειες ελεύθερης μορφής. Με την βοήθεια προηγμένων συστημάτων σχεδίασης και κατασκευής με χρήση ηλεκτρονικού υπολογιστή (**cad/cam**), έγινε εφικτό να επιλυθεί στατικά αλλά και να κατασκευαστεί ακριβώς ως είχε η μελέτη.

Μια δεύτερη ομάδα από εφαρμογές είναι τα Συστήματα Ευφυών Μεταφορών (**ITS, Intelligent Transportation Systems**), που υποσύνολο τους είναι τα Σύγχρονα Συστήματα

Υποστήριξης Οδηγού (**ADAS**-Advanced Driver Assistance Systems) τα οποία περιλαμβάνουν εκτός άλλων την μέθοδο Προειδοποίησης Αλλαγής Λωρίδας (**LDW** - Lane Departure Warning), το Σύστημα Προσαρμόσιμου Ελέγχου Ταχύτητας (**ACC** - Adaptive Cruise Control), το Σύστημα Προειδοποίησης Σύγκρουσης (**CWS** - Collision Warning System), την μέθοδο Προειδοποίησης Ελέγχου Ταχύτητας σε Καμπύλη (**CSW** - Curve Speed Warning), κ.α. Για τα περισσότερα από αυτά τα Σύγχρονα Συστήματα Υποστήριξης Οδηγού και ιδιαίτερα για την εφαρμογή της Προειδοποίησης Αλλαγής Λωρίδας είναι απαραίτητο να έχουμε πληροφορίες για την ακριβή θέση του οχήματος αλλά και για την οδό. Για το πρώτο μπορούμε να έχουμε τις απαραίτητες πληροφορίες τοποθετώντας στο όχημα έναν δέκτη GPS (Global Positioning System) ο οποίος με την σημερινή τεχνολογία, παρέχει ακρίβεια 10 cm σε πραγματικό χρόνο που είναι αποδεκτό μέγεθος. Επίσης στα επόμενα ένα με δυο χρόνια αναμένεται να ενεργοποιηθεί και το ευρωπαϊκό σύστημα εντοπισμού θέσης Galileo που σε συνεργασία με το αμερικανικό GPS θα παρέχουν ακόμα μεγαλύτερης ακρίβειας μετρήσεις. Για τις πληροφορίες της οδού που χρειάζονται οι εφαρμογές αυτές, αναφέρθηκε ότι ο γεωμετρικός σχεδιασμός των οδών δεν μπορεί να τις προσφέρει, οπότε πρέπει να αναζητηθούν λύσεις σε αριθμητικές μεθόδους. Σε εργασία που παρουσιάστηκε στο πέμπτο Ελβετικό συνέδριο μεταφορών από τους H. Gontran, P.-Y. Gilliéron, J. Skaloud, [9] χρησιμοποιώντας μια μέθοδο του K. Atkinson [10] για την προσομοίωση οδού στον χώρο με spline παρεμβολή, παρουσιάστηκε ένα ερευνητικό πρόγραμμα κατά το οποίο, με την βοήθεια ενός ειδικά εξοπλισμένου οχήματος (Photobus), το οποίο διαθέτει εξοπλισμό GPS και ψηφιακές φωτογραφικές μηχανές, παρέχεται η δυνατότητα να προσομοιώσης υπάρχουσων οδών με καμπύλες ελεύθερης μορφής (splines). Το όχημα αυτό έχει δυνατότητα να δίνει στοιχεία σε πραγματικό χρόνο και για τον άξονα της οδού αλλά και για τις λωρίδες κυκλοφορίας με αποτέλεσμα να παράγεται μία ενιαία τρισδιάστατη χάραξη που μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σύστημα

2.2 ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΤΙΣ ΚΑΜΠΥΛΕΣ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ

Πριν την εποχή των υπολογιστών, οι αρχιτέκτονες, οι μηχανικοί, οι σχεδιαστές και οι καλλιτέχνες σχεδίαζαν με μολυβί ή πενάκι και διάφορα αλλιά εργαλεία ώστε να αποδώσουν τα σχέδια τους στο χαρτί. Τα εργαλεία αυτά ήταν ο χάρακας και το ταυ για την σχεδίαση των ευθειών, ο διαβήτης για τους κύκλους και τα κυκλικά τόξα και το τρίγωνο και το μοιρογνώμονιο για την σχεδίαση γωνιών. Φυσικά, πολλές ενδιαφέρουσες μορφές δεν ήταν δυνατόν να σχεδιαστούν με αυτά τα απλά εργαλεία, γιατί είχαν πολύπλοκα τμήματα που δεν αποτελούσαν ούτε τμήμα κύκλου ούτε έλλειψης. Συχνά, μια καμπύλη έπρεπε να περνά ομαλά από διάφορα προκαθορισμένα σημεία, απαίτηση ιδιαίτερα συχνή στη ναυπηγική. Αν και ένας εξειδικευμένος σχεδιαστής θα μπορούσε σχεδιάσει τέτοιες καμπύλες με ελεύθερο χέρι, οι κατασκευαστές ναυπηγοί έπρεπε συχνά να κάνουν σχέδια κλίμακας 1 προς 1, όπου το μεγάλο μήκος των καμπυλών έκανε το σχέδιο στο χέρι αδύνατο. Λόγω του μεγάλου μεγέθους τους, τέτοια σχέδια γίνονταν συχνά στην σοφίτα ενός μεγάλου κτηρίου, από έναν ειδικό σχεδιαστή γνωστό ως *loftsmen* (*loft* = σοφίτα) . Για να καταφέρει να σχεδιάσει τις καμπύλες ο *loftsmen* χρησιμοποιούσε μακριές, λεπτές, εύκαμπτες λουρίδες ξύλου ή πλαστικού ή μετάλλου που ονομάζονται *splines*. Οι *splines* σταθεροποιούνταν με βάρη μολύβδου, που ονομάζονται *πάπιες* λόγω της ομοιότητάς τους με το ομώνυμο πτηνό.

Η πρώτη αναφορά του όρου *spline* ως μαθηματική καμπύλη έγινε σε άρθρο του 1946 από τον Ρουμάνο μαθηματικό Isaac Jacob Schoenberg (1903-1990) και το πρώτο βιβλίο αφιερωμένο στις συναρτήσεις *splines* είναι το “The theory of splines and their applications” από τους Ahlberg, Nilson and Walsh το 1967 [1] (σχ.2.2). Στο βιβλίο αυτό παρουσιάζονται οι πρώτες μη παραμετρικές συναρτήσεις *splines* με βασική την κυβική *spline* (cubic spline) όπου είναι μια σειρά από πολυώνυμα 3^{ου} βαθμού. Η πρώτη προσέγγιση στη σχεδίαση καμπύλων και επιφανειών με μη αναλυτική περιγραφή έγινε από τον Fergusson το 1960 στην Boeing. Ο



Σχ. 2.2 Το πρώτο βιβλίο αφιερωμένο στις συναρτήσεις *splines*.

Fergusson εισήγαγε την έννοια της παραμέτρου ορισμού της καμπύλης και την παραμετρική διανυσματική αναπαράσταση των καμπύλων. Τα

τμήματα Fergusson είναι κυβικά πολυωνυμικά τμήματα, που προσδιορίζονται από τα διανύσματα θέσης και τα εφαπτόμενα διανύσματα στα άκρα του τμήματος, χρησιμοποιώντας Hermite παρεμβολή. Ο Ferguson ήταν ο πρώτος που έκανε επίδειξη των δυνατοτήτων της παραμετρικής αναπαράστασης έναντι της αναλυτικής περιγραφής. Στη συνέχεια, η παραμετρική αναπαράσταση έγινε η αποκλειστική μέθοδος περιγραφής καμπύλων και επιφανειών στα συστήματα σχεδίασης με χρήση ηλεκτρονικού υπολογιστή (**CAD, Computer Aided Design**). Λίγα χρόνια αργότερα ο Bezier στη Renault εισήγαγε μια μέθοδο περιγραφής καμπύλης από το χαρακτηριστικό πολύγωνο ή πολύγωνο ελέγχου, χρησιμοποιώντας τα πολυώνυμα Bernstein ως συναρτήσεις μείξης. Η μέθοδος αυτή παρέχει ευκολία ελέγχου της μορφής της καμπύλης αλλά παρουσιάζει προβλήματα σύνδεσης των διάφορων τμημάτων μεταξύ τους, δημιουργεί πολυωνυμικές καμπύλες μεγάλου βαθμού και το τελικό μοντέλο αποτελείται από πολλά επί μέρους τμήματα και επιφανειακά μπαλώματα. Για μεγάλο χρονικό διάστημα ήταν η βασική

μορφή αναπαράστασης καμπυλών και επιφανειών ελεύθερης μορφής σε πολλά συστήματα CAD. Οι Gordon και Riesenfeld χρησιμοποίησαν τις συναρτήσεις Basis ως συναρτήσεις μείξης της καμπύλης και δημιούργησαν τις B-Splines. Για τον ορισμό τους χρησιμοποιείται το χαρακτηριστικό πολύγωνο, οι καμπύλες έχουν ιδιότητες ανάλογες με τις καμπύλες Bezier αλλά επιπλέον έχουν και τη δυνατότητα τοπικού ελέγχου της καμπύλης.

Όλες οι παραπάνω μέθοδοι δεν μπορούν να περιγράψουν με ακρίβεια κύκλους, ελλείψεις και κωνικές τομές εν γένει, μορφές που δεν περιγράφονται με συνήθη πολυώνυμα. Στην περίπτωση αυτή χρησιμοποιούνται ρητές (rational) καμπύλες, με βασικές τις καμπύλες NURBS (Non-Uniform Rational B-Splines).

2.3 Η ΠΡΟΟΠΤΙΚΗ ΧΡΗΣΗΣ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ ΣΤΗΝ ΟΔΟΠΟΙΑ

Είναι γεγονός ότι η έρευνα πάνω στις αριθμητικές μεθόδους χάραξης οδού είναι μέχρι αυτή τη στιγμή, μια εντελώς νέα προσέγγιση στον τομέα της οδοποιίας και γι' αυτό η βιβλιογραφία πάνω σε αυτήν είναι ελάχιστη και καθαρά σε ερευνητικό επίπεδο. Εκτός από την εργασία αυτή που προαναφέρθηκε των H. Gontran, P.-Y. Gilliéron, J. Skaloud, στο 3^ο διεθνές συμπόσιο γεωμετρικού σχεδιασμού των οδών παρουσιάστηκαν δύο εργασίες που προσέγγιζαν τις καμπύλες spline. Στην πρώτη του Kuhn Wolfgang [11] αναφέρεται ότι η κυβική καμπύλη spline μπορεί να προσομοιώσει την γεωμετρική χάραξη εφόσον χρησιμοποιηθεί μια σειρά από ξεχωριστά τρισδιάστατα τμήματα ενωμένα μεταξύ τους και γίνεται μια παρουσίαση της μαθηματικής έννοιας της καμπύλης αυτής. Στην δεύτερη των Tilger Klaus και Appelt Veit [12] περιγράφονται οι δυνατότητες που δίνονται στον μελετητή χρησιμοποιώντας επιφάνειες spline για τον σχεδιασμό των οδών και παρουσιάζονται κάποια παραδείγματα σε υπάρχουσες

εφαρμογές ηλεκτρονικού υπολογιστή. Οι εργασίες αυτές όμως αναφέρονται σε προσομοίωση είτε υπάρχουσας οδού είτε γεωμετρικής χάραξης με καμπύλες spline, και όχι σε εξ' αρχής ανεξάρτητη χάραξη με καμπύλες spline.

Μια αναφορά ακόμα στις καμπύλες αυτές γίνεται στον νέο Ιταλικό κανονισμό σχεδιασμού οδικών έργων (“Norme funzionali e geometriche per la costruzione delle strade”, D.M. 5.11.2001) [13]. Στον κανονισμό αυτόν αναφέρεται ότι επιτρέπεται η χρήση πολυωνυμικών καμπύλων για τον σχεδιασμό των οδών εφόσον αυτές προσομοιωθούν με μια συμβατική χάραξη που είναι αποδεκτή από τους κανονισμούς. Η απαίτηση για προσομοίωση με συμβατική χάραξη υπάρχει γιατί παρότι από τον κανονισμό επιτρέπεται η χρήση των καμπύλων αυτών, δεν υπάρχει καμία αναφορά σε περιορισμούς και μεθοδολογία χρήσης τους, αλλά ούτε και σε επιτρεπόμενα όρια με βάση αυτές τις καμπύλες. Για να είναι δηλαδή αποδεκτή μια χάραξη με πολυωνυμικές καμπύλες από τον Ιταλικό κανονισμό πρέπει να υπάρχει μια ποιοτικά ίδια χάραξη με συμβατικές μεθόδους. Στον κανονισμό αυτόν αναφέρεται επίσης ότι η χρήση των καμπύλων αυτών μπορεί να γίνει είτε σε δυο διαστάσεις, δηλαδή χάραξη της οριζοντιογραφίας με πολυωνυμικές καμπύλες και μετά χάραξη μηκοτομής με ίδιου τύπου καμπύλες, αλλά χωρίς να είναι μία αυτοτελής καμπύλη στον χώρο, είτε κατευθείαν χάραξη στον χώρο με μία ενιαία τρισδιάστατη καμπύλη.

Στην παρούσα διπλωματική εξετάζεται η πρώτη περίπτωση.

2.4 ΜΑΘΗΜΑΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΤΩΝ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ.

Ο σχεδιασμός επιπέδων καμπύλων καθώς και επιφανειών στο χώρο παίζει έναν αρκετά σημαντικό ρόλο στην κατασκευή διαφόρων προϊόντων. Κατά τα τελευταία

χρόνια της δεκαετίας του πενήντα με την εξέλιξη των ηλεκτρονικών υπολογιστών άρχισε η μελέτη σχεδιασμού των επιφανειών ή καμπύλων με νέες μεθόδους και ιδιαίτερα με θεωρητικές και ίσως πιο πολύπλοκες συναρτήσεις απ' ότι χρησιμοποιούνταν μέχρι τότε. Έχουν διαμορφωθεί δύο φιλοσοφίες για το σχεδιασμό επιφανειών :

- A) Αρχίζοντας από υπάρχουσες φόρμες
- B) Σχεδιασμός θεωρητικά των καμπύλων

Η πρώτη έχει καθιερωθεί σαν Coon's και η δεύτερη σαν Bezier. Από τους Gordon και Riesenfeld έχει αποδειχθεί ότι οι B-SPLINES είναι κατάλληλες για την εφαρμογή των καμπύλων Bezier.

Η συνάρτηση B-SPLINE ορίζεται ως εξής :

$$S(u) = \sum_i d_i N_i^n(u)$$

$$\text{όπου } \dots < u_0 < u_1 < u_2 < \dots$$

$$\sum_i N_i^n(u) = 1$$

$$N_i^n(u) \geq 0$$

$$N_i^n(u) = 0 \quad \text{αν } u \notin [u_i, u_{i+n+1}]$$

$N_i^n(u)$ είναι (n-1) φορές συνεχώς παραγωγίσιμη

$$N_i^r(u) = (u - u_i) \frac{N_i^{r-1}(u)}{u_{i+r} - u_i} + (u_{i+r+1} - u) \frac{N_{i+1}^{r-1}(u)}{u_{r+i+1} - u_{i+1}}$$

$$\text{όπου } N_i^0 = \begin{cases} 1 & \text{αν } u \in [u_i, u_{i+1}] \\ 0 & \text{αν } u \notin [u_i, u_{i+1}] \end{cases}$$

Αρχικά, μια επιφάνεια αναλύεται σαν σειρά από ένα μεγάλο αριθμό σημείων με δεδομένες τις συντεταγμένες τους. Δίνοντας στον υπολογιστή ορισμένα σημεία, από τα οποία θέλουμε να διέλθει η καμπύλη και τις προδιαγραφές που θα πρέπει να πληρεί η επιφάνεια ή επίπεδη καμπύλη, υπολογίζονται από το πρόγραμμα τα σημεία της αντίστοιχης επιφάνειας.

Μέχρι τώρα οι συναρτήσεις SPLINES έχουν χρησιμοποιηθεί στη βιομηχανία, στην τοπογραφία κ.τ.λ. Στη βιομηχανία χρησιμοποιήθηκαν για το σχεδιασμό διαφόρων προϊόντων όπως αυτοκινήτων, πλοίων, αεροσκαφών καθώς και για το σχήμα πετρώγων, ελίκων πλοίων, σχήμα υποδημάτων, μπουκαλιών κ.τ.λ. αλλά και στην περιγραφή γεωλογικών, φυσικών ακόμα και ιατρικών φαινομένων.

Επίσης, οι SPLINES χρησιμοποιούνται για προσαρμογή μιας επιφάνειας ή επίπεδης καμπύλης σε δεδομένα που έχουν ληφθεί από μετρήσεις. Τυπικό παράδειγμα αποτελεί ο υπολογισμός της κατανομής των θερμοκρασιών σε μια περιοχή εάν υπάρχουν μετρήσεις θερμοκρασιών σε διάφορα σημεία της περιοχής.

2.5 ΜΑΘΗΜΑΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΩΝ B-SPLINES ΣΤΟ ΣΧΕΔΙΑΣΜΟ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ ΟΔΟΥ

Οι καμπύλες ελεύθερης μορφής στο εν λόγω πρόβλημα καλούνται να περιγράψουν τον άξονα μιας οδού. Ο άξονας που πρόκειται να χαραχτεί ορίζεται από συγκεκριμένα σημεία με καθορισμένες συντεταγμένες. Με το σύνολο αυτών των σημείων και των συνοριακών συνθηκών, μια συνάρτηση SPLINE μπορεί να ορισθεί ως εξής:

Έστω ότι θέλουμε η συνάρτηση να διέρχεται από τα σημεία P_i όπου $i=0,1,...,N$. Εισάγεται ο ακόλουθος συνδυασμός:

$$h_i = x_i - x_{i-1} \quad i = 1, ..., N$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \quad i = 1, \dots, N-1$$

$$\mu_i = 1 - \lambda_i \quad i = 1, \dots, N-1$$

Έτσι, η μη περιοδική συνάρτηση SPLINE καθορίζεται από το ακόλουθο σύστημα εξισώσεων:

$$\begin{bmatrix} 2 & \lambda_0 & 0 & 0 & \cdot & \cdot & \cdot \\ \mu_1 & 2 & \lambda_1 & 0 & \cdot & \cdot & \cdot \\ 0 & \mu_2 & 2 & \lambda_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 2 & \lambda_{N-2} & 0 \\ \cdot & \cdot & \cdot & \cdot & \mu_{N-1} & 2 & \lambda_{N-1} \\ \cdot & \cdot & \cdot & \cdot & 0 & \mu_N & 2 \end{bmatrix} \circ \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \cdot \\ \cdot \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

$$\text{όπου } d_i = 6 \cdot \left[\left[(y_{i+1} - y_i) / h_{i+1} \right] - \left[(y_i - y_{i-1}) / h_i \right] \right] / (h_i + h_{i+1})$$

Η εξίσωση των συναρτήσεων SPLINE έχει την ακόλουθη μορφή:

$$\begin{aligned} y = S(x) &= M_{i-1} \cdot \frac{(x_i - x)^3}{6h_i} + M_i \cdot \frac{(x - x_{i-1})^3}{6h_i} + \left(y_{i-1} - \frac{M_{i-1} \cdot h_i^2}{6} \right) \cdot \frac{x_i - x}{h_i} + \\ &+ \left(y_i - \frac{M_i \cdot h_i^2}{6} \right) \cdot \frac{x - x_{i-1}}{h_i} \\ y' = S'(x) &= M_{i-1} \cdot \frac{(x_i - x)^2}{2h_i} + M_i \cdot \frac{(x - x_{i-1})^2}{2h_i} + \frac{y_i - y_{i-1}}{h_i} - \frac{M_i - M_{i-1}}{6} \cdot h_i \end{aligned}$$

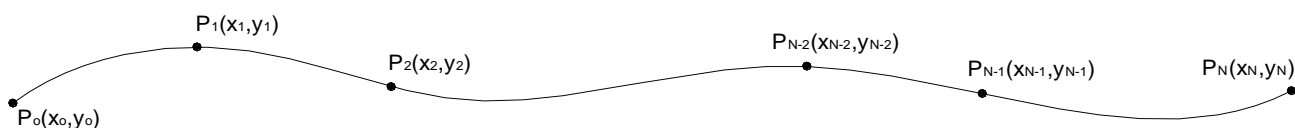
$$y'' = S''(x) = M_{i-1} \cdot \frac{x_i - x}{h_i} - M_i \cdot \frac{x - x_{i-1}}{h_i}$$

Το παραπάνω σύστημα εξισώσεων υπολογίζει τις τιμές M_i με $i=0,1,\dots,N$. Παρά το ότι M_i δεν είναι ροπές στα σημεία P_i συνηθίζεται να γράφονται ως M .

Γενικά, μπορούμε να ορίσουμε την κλίση ή την καμπυλότητα της συνάρτησης SPLINE στα συνοριακά σημεία. Για την εφαρμογή, ωστόσο, στην περίπτωση αυτή είναι αρκετά ικανοποιητικό να θεωρηθεί ότι $M_0=M_N=0$ που αντιστοιχεί στο να τοποθετηθούν απλές στηρίξεις στα συνοριακά σημεία της συνάρτησης SPLINE ή στα σημεία κοντά στο σύνορο.

Έτσι, το σύστημα που πρέπει να επιλυθεί έχει την ακόλουθη μορφή:

$$\begin{bmatrix} 2 & \lambda_0 & 0 & 0 & \cdot & \cdot & \cdot \\ \mu_1 & 2 & \lambda_1 & 0 & \cdot & \cdot & \cdot \\ 0 & \mu_2 & 2 & \lambda_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 2 & \lambda_{N-2} & 0 \\ \cdot & \cdot & \cdot & \cdot & \mu_{N-1} & 2 & \lambda_{N-1} \\ \cdot & \cdot & \cdot & \cdot & 0 & \mu_N & 2 \end{bmatrix} \cdot \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \cdot \\ \cdot \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$



Επιθυμείται η γραφική παράσταση της συνάρτησης να περάσει από τα σημεία, όπου $i=1,2,\dots,N$. Εισάγεται ο ακόλουθος συμβολισμός:

$$h_i = x_i - x_{i-1} \quad i = 1, \dots, N$$

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \quad i = 1, \dots, N-1$$

$$\mu_i = 1 - \lambda_i \quad i = 1, \dots, N-1$$

Έτσι, η μη περιοδική συνάρτηση SPLINE καθορίζεται από το ακόλουθο σύστημα εξισώσεων, όπου d_i δίνεται από την έκφραση:

$$d_i = 6 \cdot \left[\left[(y_{i+1} - y_i) / h_{i+1} \right] - \left[(y_i - y_{i-1}) / h_i \right] \right] / (h_i + h_{i+1})$$

$$\begin{bmatrix} 2 & \lambda_0 & 0 & 0 & . & . & . \\ \mu_1 & 2 & \lambda_1 & 0 & . & . & . \\ 0 & \mu_2 & 2 & \lambda_2 & . & . & . \\ 0 & 0 & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & 2 & \lambda_{N-2} & 0 \\ . & . & . & . & \mu_{N-1} & 2 & \lambda_{N-1} \\ . & . & . & . & 0 & \mu_N & 2 \end{bmatrix} \circ \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ . \\ . \\ M_{N-2} \\ M_{N-1} \\ M_N \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ . \\ . \\ d_{N-2} \\ d_{N-1} \\ d_N \end{bmatrix}$$

Αλλά έχει θεωρηθεί ότι $M_0=M_N=0$, άρα πρέπει να επιλυθεί το σύστημα:

$$\begin{bmatrix} 2 & \lambda_1 & 0 & . & . & . & . \\ \mu_2 & 2 & \lambda_2 & . & . & . & . \\ 0 & \mu_3 & 2 & \lambda_3 & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & \mu_{N-3} & 2 & \lambda_{N-3} \\ . & . & . & . & 0 & \mu_{N-2} & 2 \\ . & . & . & . & 0 & 0 & \mu_{N-1} \end{bmatrix} \circ \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ . \\ . \\ M_{N-3} \\ M_{N-2} \\ M_{N-1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ . \\ . \\ d_{N-3} \\ d_{N-2} \\ d_{N-1} \end{bmatrix}$$

Αρχικά, πρέπει να υπολογιστούν τα στοιχεία των μητρώων με την ακόλουθη αντιστοιχία:

h_i ----- $H(i)$

μ_i ----- $P(i)$

2 ----- $Q(i)$

λ_i ----- $R(i)$

d_i ----- $S(i)$

$$H_i = X(i+1) - X(i) \quad i = 1, \dots, N-1$$

$$R_i = \frac{H(i+1)}{H(i-1) + H(i)} \quad i = 1, \dots, N-2$$

$$P(i) = 1 - R(i) \quad i = 1, \dots, N-2$$

$$Q(i) = 2 \quad i = 1, \dots, N-1$$

$$S(i) = 6 \cdot \frac{(Y(i+1) - Y(i))/H(i) - (Y(i) - Y(i-1))/H(i-1))}{H(i-1) + H(i)} \quad i = 1, \dots, N-2$$

Αφού έχουν υπολογιστεί τα στοιχεία των πινάκων, στη συνέχεια πρέπει να επιλυθεί το σύστημα που έχει τη γενική εξίσωση:

$$P(i) \cdot XM(i-1) + 2 \cdot XM(i) + R(i) \cdot XM(i+1) = S(i) \quad i = 1, \dots, N-2$$

$$\text{όπου } R(N-2) = 0 \text{ και } P(1) = 0$$

Η επίλυση γίνεται με τη μέθοδο απαλοιφής του Gauss. Προκειμένου να επιλυθεί το σύστημα $[A] [X] = [B]$, η μέθοδος απαλοιφής του Gauss γίνεται σε δύο κύριες φάσεις. Εν προκειμένω ισχύει:

$$[A] = \begin{bmatrix} 2 & R_1 & 0 & \cdot & \cdot & \cdot & \cdot \\ P_2 & 2 & R_2 & \cdot & \cdot & \cdot & \cdot \\ 0 & P_3 & 2 & R_3 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & P_{N-3} & 2 & R_{N-3} \\ \cdot & \cdot & \cdot & \cdot & 0 & P_{N-2} & 2 \\ \cdot & \cdot & \cdot & \cdot & 0 & 0 & P_{N-1} \end{bmatrix} \quad [B] = \begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ S_{N-2} \\ S_{N-1} \end{bmatrix}$$

1^η φάση: Τριγωνοποίηση του πίνακα $[A]$ του συστήματος. Εκτελείται με διαδοχικές εφαρμογές πράξεων. Έτσι ο πίνακας $[A]$ μετατρέπεται σε άνω-τριγωνικό.

2^η φάση: Πίσω αντικατάσταση, όπου υπολογίζονται οι άγνωστοι x_n, \dots, x_1 κατά σειρά με διαδοχικές αντικαταστάσεις στις εξισώσεις, από την τελευταία ως την πρώτη.

Τριγωνοποίηση:

Στο πρώτο βήμα κρατείται αμετάβλητη η πρώτη εξίσωση, πολλαπλασιάζεται επί $m_2 = -\frac{a_{21}}{a_{11}}$ και προστίθεται στη δεύτερη εξίσωση. Στη συνέχεια πολλαπλασιάζεται η πρώτη εξίσωση επί $m_3 = -\frac{a_{31}}{a_{11}}$ και την προστίθεται στην τρίτη και ούτω καθεξής μέχρι την εξίσωση n . Απαλείφεται έτσι ο άγνωστος x_1 από τις εξισώσεις 2 έως n και καταλήγουμε στο ισοδύναμο σύστημα με νέους συντελεστές και δεύτερα μέλη.

$$a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \dots + a_{1n}^{(1)} x_n = b_1^{(1)}$$

$$a_{22}^{(2)} x_2 + \dots + a_{2n}^{(2)} x_n = b_2^{(2)}$$

.

.

.

$$a_{n2}^{(n)} x_2 + \dots + a_{nn}^{(n)} x_n = b_n^{(n)}$$

$$\text{όπου} \quad a_{22}^{(2)} = a_{22} + m_2 a_{12} \quad \text{κ.τ.λ.}$$

$$b_2^{(2)} = b_2 + m_2 b_1 \quad \text{κ.τ.λ.}$$

Στο δεύτερο βήμα επαναλαμβάνουμε τις πράξεις στο υποσύστημα $(n-1) \times (n-1)$ που αποτελείται από τις εξισώσεις 2 μέχρι n , όπου εμπλέκονται οι άγνωστοι x_2, \dots, x_n μόνο και ούτω καθεξής, $n-1$ φορές. Το τελικό σύστημα γίνεται έτσι άνω-τριγωνικό, ισοδύναμο με το πρώτο και έχει την ακόλουθη μορφή:

$$a_{11}^{(1)} x_1 + a_{12}^{(1)} x_2 + \dots + a_{1n}^{(1)} x_n = b_1^{(1)}$$

$$a_{22}^{(2)} x_2 + \dots + a_{2n}^{(2)} x_n = b_2^{(2)}$$

.

$$a_{nn}^{(n)} x_n = b_n^{(n)} .$$

Πίσω αντικατάσταση:

Προκειμένου να επιλυθεί πλέον το νέο σύστημα υπολογίζουμε τον άγνωστο $x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}}$ από την εξίσωση n, τον αντικαθιστούμε στην εξίσωση n-1 για να υπολογίσουμε τον x_{n-1} και ούτω καθεξής μέχρι τον x_1 . Στην περίπτωση μας οι άγνωστοι είναι οι λεγόμενες ροπές $M(i)$ που στο πρόγραμμα συμβολίζονται με $XM(i)$.

Αφού έχει ολοκληρωθεί ο υπολογισμός των ροπών, ο οποίος είναι απαραίτητος για τον υπολογισμό της συνάρτησης SPLINE, ακολουθεί ο υπολογισμός των συντεταγμένων κάθε σημείου δοθείσης της τετμημένης. Επίσης υπολογίζονται, όπως αναφέρθηκε και παραπάνω, η πρώτη παράγωγος της συνάρτησης $y'(x)$ και η δεύτερη παράγωγος $y''(x)$.

2.6 ΕΦΑΡΜΟΓΗ ΤΩΝ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ (B-SPLINES) ΣΤΗΝ ΟΔΟΠΟΙΙΑ

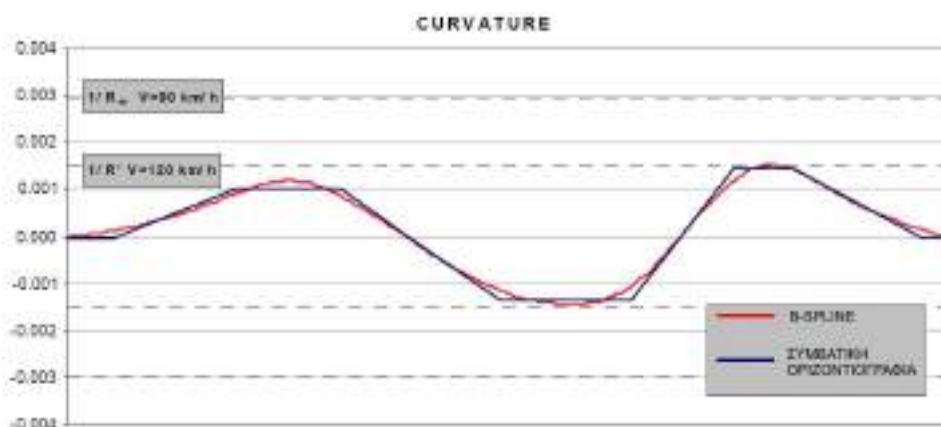
Μια οριζοντιογραφική χάραξη σχεδιασμένη με συμβατικές μεθόδους αποτελείται από ευθύγραμμα τμήματα, κυκλικά τόξα και καμπύλες συναρμογής. Ο συνδυασμός των τριών αυτών γεωμετρικών σχημάτων έχει αποτέλεσμα μια καμπύλη που παρουσιάζει συνέχεια C^2 δηλαδή συνέχεια καμπυλότητας. Η χρήση μάλιστα του τόξου συναρμογής γίνεται για να αποκτήσει η καμπύλη συνέχεια C^2 από C^1 που θα είχε εάν ενώνονταν απ' ευθείας τα ευθύγραμμα τμήματα με τα κυκλικά τόξα. Η έννοια της συνέχειας έχει μεγάλη πρακτική σημασία στην συμπεριφορά του οδηγού. Διατρέχοντας μία χάραξη χωρίς τόξα συναρμογής, δηλαδή με συνέχεια C^1 , ο οδηγός τη στιγμή που μπαίνει στο κυκλικό τόξο πρέπει να στρέψει το τιμόνι του ακαριαία κάποιες μοίρες, ώστε να μπει στο κυκλικό τόξο. Επειδή όμως η ακαριαία στροφή είναι αδύνατη το όχημα εκτρέπεται ελαφρά από την πορεία του και επανέρχεται μετά από μερικά μέτρα. Η εφαρμογή των τόξων συναρμογής λύνει αυτό το πρόβλημα παρεμβάλλοντας μεταξύ του ευθύγραμμου τμήματος και του κυκλικού τόξου, μια κλωθοειδή καμπύλη με μεταβαλλόμενη καμπυλότητα γραμμικά από 0 σε R, δίνοντας στην συνολική καμπύλη συνέχεια C^2 . Έτσι στην συμβατική οριζοντιογραφία ο οδηγός έχει σταθερό το τιμόνι στην ευθεία και όταν εισέρχεται στο τόξο συναρμογής το στρέφει με σταθερή γωνιακή ταχύτητα ώσπου να εισέλθει στο κυκλικό τόξο όπου πάλι το κρατάει σταθερό κτλ.

Η ασυνέχεια που παρουσιάζεται σε αυτήν την περίπτωση είναι ότι ενώ ο οδηγός κρατάει σταθερό το τιμόνι, ξαφνικά αρχίζει να το στρέφει με μία σταθερή γωνιακή ταχύτητα ω . Η ανάπτυξη σταθερής ταχύτητας χωρίς να έχει προηγηθεί επιτάχυνση είναι αδύνατη, άρα και σε αυτήν την περίπτωση υπάρχει μια πολύ μικρή εκτροπή του οχήματος από την πορεία του, που όμως είναι αμελητέα γιατί έτσι κι αλλιώς ο οδηγός

είναι αδύνατο να κάνει κινήσεις τόσο μεγάλης ακρίβειας. Η συνέχεια που πρέπει να έχει μία καμπύλη ώστε ο οδηγός να στρέφει το τιμόνι του με επιταχυνόμενη κίνηση και όχι με σταθερή ταχύτητα είναι η C^3 , η οποία δεν αναφέρθηκε παραπάνω γιατί είναι καθαρά μαθηματική συνέχεια και όχι γεωμετρική.

Η συνέχεια που παρουσιάζει μια καμπύλη B-Spline είναι $k-2$ όπου k είναι η τάξη της καμπύλης, άρα μια καμπύλη B-Splines 5^{ης} τάξης παρουσιάζει συνέχεια C^3 , δηλαδή θεωρητικά μπορεί ένα όχημα να ακολουθήσει τον άξονα της με απόλυτη ακρίβεια. Γεωμετρικά η συνέχεια C^3 είναι εμφανής στο διάγραμμα καμπυλότητας, όπου για να έχει συνέχεια μεγαλύτερη ή ίση από C^3 μια καμπύλη, πρέπει η γραφική παράσταση της καμπυλότητας να έχει συνέχεια C^1 , δηλαδή να μην παρουσιάζει γωνίες.

Στο σχήμα 2.3 φαίνεται σε ένα διάγραμμα η γραφική παράσταση της καμπυλότητας μιας συμβατικής οριζοντιογραφίας με μπλε και μιας B-Spline χάραξης 5^{ης} τάξης με κόκκινο. Είναι εμφανές ότι η καμπυλότητα για την συμβατική χάραξη, στα σημεία συναρμογής της κλωθοειδούς με την ευθυγραμμία αλλά και με το κυκλικό τόξο, παρουσιάζει θλάση (μαθηματικά τα σημεία αυτά ονομάζονται ιδιαίτερα (singular)), ενώ για την B-Spline χάραξη παρουσιάζει συνέχεια C^1 .



Σχ. 2.3 Διάγραμμα καμπυλότητας για παρόμοια χάραξη με καμπύλη B-Spline και με συμβατική χάραξη [14]

2.6.1 Ακτίνα Οριζοντιογραφικής Καμπύλης

Όπως φαίνεται στο σχήμα 2.3, η μεταβολή της καμπυλότητας στην B-Spline χάραξη είναι συνεχής, που σημαίνει ότι δεν υπάρχει ευθυγραμμία (η ευθυγραμμία στο διάγραμμα καμπυλότητας φαίνεται όπου υπάρχει σταθερή μηδενική καμπυλότητα) και αυτή είναι μια από τις σημαντικότερες διαφορές οριζοντιογραφικά με την συμβατική χάραξη. Η καμπύλη παρουσιάζει μεγαλύτερη συνέχεια αλλά παντού υπάρχει κάποια, έστω και μικρή καμπυλότητα. Επίσης τα καμπύλα τμήματα της οδού, δηλαδή τα τόξα συναρμογής και το κυκλικό τόξο αντικαθίστανται από την συνεχή χάραξη B-Spline.

Στο τεύχος των OMOE-X αναφέρονται κάποιοι βασικοί περιορισμοί που πρέπει να λαμβάνονται υπ' όψη από τον μελετητή κατά την χάραξη της οριζοντιογραφίας. Οι περιορισμοί αυτοί σε μια B-Spline χάραξη δεν μπορούν να εφαρμοστούν πλήρως αλλά μια ποιοτική σύγκριση είναι δυνατή.

Ο βασικός περιορισμός των ελαχίστων ακτίνων καμπυλότητας μπορεί να τηρηθεί, αφού για κάθε σημείο της καμπύλης η ακτίνα καμπυλότητας είναι γνωστή. Μάλιστα η μέγιστη καμπυλότητα είναι στιγμιαία άρα οι οριακές τιμές των ακτίνων μπορεί να χρησιμοποιηθούν με μεγαλύτερη ασφάλεια (στους Αμερικανικούς κανονισμούς τονίζεται ότι η χρήση της ελάχιστης επιτρεπόμενης ακτίνας καμπυλότητας θα πρέπει να αποφεύγεται, γιατί αντιστοιχεί σε οριακή κατάσταση). Επίσης μπορεί να τηρηθεί και ο περιορισμός της αλληλουχίας των διαδοχικών καμπύλων, λαμβάνοντας σαν διαδοχικές καμπύλες τα τμήματα που παρουσιάζουν ελάχιστο κάτω από μια μεγάλη τιμή, π.χ. 2000μ.

2.6.2 Έλεγχος Ακτίνας Καμπυλότητας

Σύμφωνα με τις οδηγίες ΟΜΟΕ 2001, η ακτίνα καμπυλότητας θα πρέπει να επιλέγεται κατάλληλα με κριτήρια που θα πληρούν τις παρακάτω προϋποθέσεις.: (Παρατίθεται τμήμα του ΟΜΟΕ 2001 που αφορά στους ελέγχους ακτίνας καμπυλότητας)

- κατά κανόνα συνολικώς μικρά μήκη καμπύλων,
- επαρκή μήκη ορατότητας για προσπέραση,
- αρμονία και συνέχεια στην οδική συμπεριφορά.

Παράλληλα με τους προαναφερθέντες σκοπούς, η επιλογή των ακτινών θα πρέπει να είναι τέτοια, ώστε :

- η οδός να προσαρμόζεται κατά μορφή και μέγεθος με το ανάγλυφο του εδάφους και το τοπίο,
- όσον αφορά τα μεγέθη και την αλληλουχία των καμπύλων, να εξασφαλίζεται η συμβατότητα μεταξύ οριζοντιογραφίας και μηκοτομής, και να επιτυγχάνεται η καλή ανάπτυξη της οδού στο χώρο.
- να εξασφαλίζεται η αρμονική σχέση μεταξύ ταχύτητας μελέτης V_e και λειτουργικής ταχύτητας V_{85} σύμφωνα με την Περίπτωση 1 του Κριτηρίου Ασφαλείας Ι (βλ. Πίνακα 2.1).

Επιτρέπεται η θλάση του άξονα της χάραξης χωρίς εφαρμογή οριζόντιας καμπύλης όταν η γωνία θλάσης γ είναι μικρότερη από την τιμή που προκύπτει (ανάλογα με την ταχύτητα μελέτης V_e) από τις σχέσεις :

- όταν $V_e < 70\text{km/h}$, $\gamma = \arctan (1,6/V_e)$
- όταν $V_e \geq 70\text{km/h}$, $\gamma = \arctan (155/V_e^2)$

Τυπικές και οριακές τιμές

Για τις οδούς των ομάδων Α και Β οι ελάχιστες ακτίνες των καμπύλων R_{\min} δίδονται στον παρακάτω πίνακα. Οι τιμές αυτές εξαρτώνται από την ταχύτητα μελέτης V_e , τον βαθμό εκμετάλλευσης του συντελεστή εγκάρσιας τριβής n , και τις οριακές τιμές της επίκλισης). Για λόγους καλύτερης προσαρμογής στις κλιματολογικές και τις τοπογραφικές συνθήκες της Ελλάδας, η επιλογή των τιμών R_{\min} εξαρτάται εκτός από την ομάδα, στην οποία ανήκει η οδός, και από την κατηγορία του εδάφους (πεδινό, λοφώδες ή ορεινό).

R_{\min} [m]						
Ομάδα οδών Α					Ομάδα οδών Β	
V_e	πεδινά εδάφη			λοφώδη και ορεινά εδάφη		όλες οι κατηγορίες εδαφών
[km/h]	$q_{\max}=8$ (9)%	$q_{\min}=2,5\%$	$q_{\max}=7\%$	$q_{\min}=2,5\%$	$q_{\max}=6\%$	$q_{\min}=2,5\%$
$n=45\%$	$n=10\%$		$n=40\%$	$n=10\%$	$n=60\%$	$n=30\%$
1	2	3	4	5	6	7
50	80	325	95	325	70	150
60	125 (120)	490	140	490	110	230
70	180 (170)	700	200	700	160	335
80	250 (235)	960	280	960	220	470
90	330 (310)	1.260	370	1.260	300	630
100	420 (400)	1.620	480	1.620	—	—
110	530 (500)	2.020	600	2.020	—	—
120	650 (620)	2.470	740	2.470	—	—
(130)	790 (740)	2.970	890	2.970	—	—

Πίνακας 2.1 Ελάχιστες ακτίνες καμπύλων για οδούς των ομάδων Α και Β. Οι τιμές σε παρένθεση εφαρμόζονται σε εξαιρετικές περιπτώσεις[6]

2.6.4 Ορατότητα και στοιχεία μελέτης της.

Μήκος ορατότητας

. Το μήκος ορατότητας ορίζεται από το τμήμα της οδού που εκτίθεται στο οπτικό πεδίο του οδηγού σε κάθε χρονική στιγμή. Όσο μικρότερο είναι το μήκος ορατότητας, τόσο λιγότερη οπτική πληροφορία είναι διαθέσιμη για αξιολόγηση, με αποτέλεσμα ένας οδηγός να χρειάζεται πιο συχνά να ανανεώνει τις προβλέψεις του. Όταν το μήκος ορατότητας περιορίζεται, η σημασία της εμπειρίας του οδηγού με την οδό αυξάνεται. Η εμπειρία με την οδό σχετίζεται με τον αριθμό διελεύσεων του οδηγού από ένα συγκεκριμένο τμήμα της οδού και την ομοιότητα της οδού ε άλλες τις οποίες αυτός έχει ήδη εμπειρία. Ένας οδηγός προσδοκά ότι η πορεία ή η γεωμετρία της οδού θα έχει ομοιογένεια και ότι είναι προβλέψιμη ακόμη και όταν το μήκος ορατότητας περιορίζεται. Η ομοιογένεια της γεωμετρίας της οδού επιτρέπει στον οδηγό να προβλέπει με ακρίβεια τη σωστή πορεία, ενώ παράλληλα να αφιερώνει την ικανότητα του σε επεξεργασία μόνο μικρού όγκου οπτικής πληροφορίας. Δηλαδή η ομοιογένεια στη γεωμετρία της οδού επιτρέπει στην προσοχή και στην ικανότητα του οδηγού να αφιερώνεται περισσότερο στην αποφυγή απροσδόκητων εμποδίων και στην οδήγηση.

Η ασφάλεια της κυκλοφορίας και η ποιότητα της κυκλοφοριακής ροής απαιτούν την ύπαρξη ελάχιστων μηκών ορατότητας, προκειμένου να είναι δυνατή η έγκαιρη ακινητοποίηση ενός οχήματος (απαιτούμενο μήκος ορατότητας για στάση), η ασφαλής προσπέραση (απαιτούμενο μήκος ορατότητας για προσπέραση) καθώς και η ασφαλής εξέλιξη της απόφασης του οδηγού για αλλαγή πορείας (μήκος ορατότητας για απόφαση).

Το απαιτούμενο μήκος ορατότητας για στάση είναι καθοριστικής σημασίας για την αξιολόγηση των συνθηκών ορατότητας σε όλες τις κατηγορίες οδών. Το ίδιο ισχύει και για το απαιτούμενο μήκος ορατότητας για προσπέραση, το οποίο χρησιμοποιείται άμεσα στην αξιολόγηση των συνθηκών ορατότητας σε οδούς με ενιαία επιφάνεια

κυκλοφορίας της ομάδας Α. Για τις οδούς της ομάδας Β το μήκος ορατότητας για προσπέραση έχει δευτερεύουσα σημασία.

Απαιτούμενο μήκος ορατότητας για στάση S_h

Το απαιτούμενο μήκος ορατότητας για στάση S_h είναι το μήκος, που χρειάζεται ένας οδηγός κινούμενος με την ταχύτητα V_{85} , για να ακινητοποιήσει το όχημα του πριν από ένα απροσδόκητο ακίνητο εμπόδιο στο οδόστρωμα. Το μήκος ορατότητας για στάση είναι το άθροισμα του μήκους που διανύει το όχημα κατά τη διάρκεια του χρόνου αντίληψης, του χρόνου αντίδρασης, και του μήκους πέδησης

Σε όλο το μήκος των οδών πρέπει να διατίθεται κατ' ελάχιστο το μήκος ορατότητας στάσης που ορίζεται από την ταχύτητα

Όταν συνεχώς το πεδίο ορατότητας του οδηγού περιορίζεται στις ελάχιστες τιμές της ορατότητας στάσης, τότε ο οδηγός βρίσκεται σε διαρκή ένταση που προκαλεί κόπωση. Γι' αυτό το λόγο επιβάλλεται να εξασφαλίζονται συνολικά στο 70% της οδού μήκη ορατότητας κατά 1,3 φορές μεγαλύτερα από τα ελάχιστα απαιτούμενα [6]. Επίσης συνιστάται οι ελάχιστες τιμές να μη χρησιμοποιούνται σε περιοχές κόμβων. Το απαιτούμενο μήκος ορατότητας για στάση είτε λαμβάνεται από το διάγραμμα Ο1 που ακολουθεί σε συνάρτηση με την ταχύτητα V_{85} και την κατά μήκος κλίση της οδού s , είτε από τους τύπους που ακολουθούν.

$$S_h = S_1 + S_2$$

$$S_1 = (V_{85} / 3,6) \cdot t_r$$

$$S_2 = (V_{85} / 3,6)^2 / [2 \cdot (d + g \cdot s)]$$

όπου :

S_h [m] = μήκος ορατότητας για στάση

S_1 [m] = διανυόμενο μήκος κατά τη διάρκεια του χρόνου αντίληψης-αντίδρασης

S_2 [m] = μήκος πέδησης

V_{85} [km/h] = λειτουργική ταχύτητα 85%

t_r [s] = χρόνος αντίληψης-αντίδρασης
= (λαμβάνεται ως 2 δευτερόλεπτα)

g [m/s²] = επιτάχυνση βαρύτητας

d [m/s²] = συντελεστής εξαρτώμενος από την ταχύτητα (η τιμή του λαμβάνεται από τον Πίνακα 10-1)

s [m/m] = κατά μήκος κλίση
[θετική (+) : ανωφέρεια,
αρνητική (-) : κατωφέρεια]

Συντελεστής d υπολογισμού του μήκους ορατότητας στάσης S_h

V_{85}	[km/h]	50	60	70	80	90	100	110	120	130
d	[m/s ²]	4,4	4,2	4,0	3,8	3,6	3,4	3,3	3,1	3,0

Πίνακας 2.6.1 Στοιχεία Ορατότητας στάσης S_h [6]

Οι ελάχιστες τιμές της ακτίνας οριζοντίων καμπυλών των χαράξεων υπολογίζονται με βάση τη δυναμική της κίνησης των οχημάτων σε συνθήκες υγρού οδοστρώματος. Η χρήση αυτών των ελάχιστων τιμών των ακτίνων υπόκειται και στον έλεγχο για επαρκή ορατότητα για στάση ενώπιον ακίνητου εμποδίου. Ο έλεγχος αυτός υποχρεωτικά γίνεται για:

- τις εξωτερικές πλευρές των οδών οπότε είτε επιβεβαιώνεται η ικανοποίηση της απαιτούμενης συνθήκης πλευρικού ελεύθερου οπτικών εμποδίων χώρου, είτε οδηγεί στην ανάγκη διεύρυνσης του πλευρικού χώρου π.χ. με πρόσθετη εκσκαφή των πρανών ορυγμάτων. Διευκρινίζεται ότι η εκσκαφή γίνεται από τη στάθμη του οδοστρώματος,
- την πλευρά της κεντρικής νησίδας αυτοκινητοδρόμου,
- την πλευρά της διαχωριστικής νησίδας μεταξύ δύο οδών
- την εξωτερική πλευρά που τοποθετούνται στηθαία (επιχώματα, γέφυρες, βάθρα γεφυρών)

Δηλαδή ακόμη και η θέση των στηθαίων ασφαλείας πρέπει να ελέγχεται ότι αφήνει τον απαιτούμενο πλευρικό ελεύθερο χώρο για να υπάρχει το απαιτούμενο μήκος ορατότητας για στάση.

Εφόσον ο προβλεπόμενος από την τυπική διατομή ελεύθερος χώρος στην πλευρά της κεντρικής νησίδας μεταξύ της οριογραμμής κυκλοφορίας και της όψης του στηθαίου ασφαλείας, δεν είναι επαρκής για να εξασφαλίσει το απαιτούμενο μήκος ορατότητας για στάση, τότε πρέπει να λαμβάνονται πρόσθετα μέτρα. Τα μέτρα αυτά είναι : είτε η μετατόπιση του στηθαίου προς την κοίλη πλευρά της νησίδας, είτε η διαπλάτυνση του ελεύθερου χώρου, σύμφωνα με διάγραμμα ορατότητας που πρέπει να σχεδιάζεται ειδικά γι' αυτές τις περιπτώσεις. Εφόσον αυτά τα μέτρα δεν είναι εφικτά τότε είναι δυνατόν να αποφασίζεται η τοποθέτηση ειδικών πινακίδων με διπλό όριο ταχύτητας. Το ένα όριο θα αφορά συνθήκες στεγνού και το άλλο συνθήκες υγρού οδοστρώματος. Για το όριο ταχύτητας με υγρό οδόστρωμα μπορεί να χρησιμοποιείται η πινακίδα Πρ-6 του ΚΟΚ. Επειδή αυτή η πινακίδα στη συγκεκριμένη περίπτωση κρίνεται μη ικανοποιητική λόγω μεγέθους, συνίσταται η υιοθέτηση μιας παρόμοιας από ΚΟΚ άλλης χώρας όπως π.χ. της Γαλλίας. Σύμφωνα με τα προαναφερόμενα, η τελικά επιτρεπόμενη ταχύτητα κυκλοφορίας σε αυτοκινητοδρόμους, καθορίζεται από συνδυασμό των παραμέτρων που είναι:

- το πλάτος πλευρικού ελεύθερου οπτικών εμποδίων χώρου μεταξύ της οριογραμμής κυκλοφορίας και της όψης του στηθαίου ασφαλείας που ορίζεται στην τυπική διατομή,
- η ακτίνα της οριζόντιας καμπύλης της χάραξης,
- η κατά μήκος κλίση της χάραξης.

Η σχέση μεταξύ της ακτίνας τόξου R για δεδομένο πλάτος M του ελευθέρου εμποδίων πλευρικού χώρου, που προσφέρει μήκος ορατότητας στάσης S_h στο μέσο της ελεγχόμενης λωρίδας κυκλοφορίας, ορίζεται από τις ακόλουθες εξισώσεις [6] (βλ. Σχήμα 2.6.1):

- όταν $S_h \leq L$ $S_h = 2 \cdot R \cdot \cos^{-1} (1 - M/R)$
ή $S_h = 2 \cdot (2 \cdot R \cdot M - M^2)^{0.5}$
- όταν $S_h > L$ $S_h = 4RM/L + L/2$

Όπου :

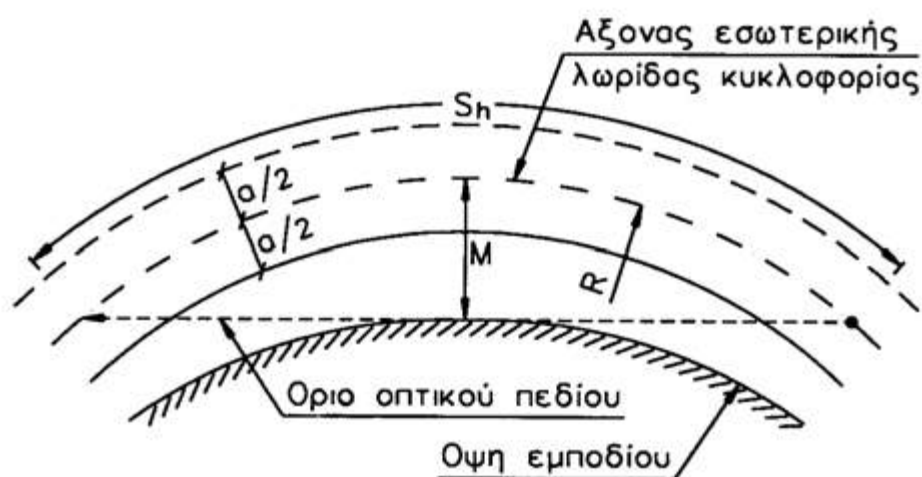
S_h [m]: Το μήκος ορατότητας στάσης μετρούμενο επί της καμπύλης στον άξονα που χαράζεται στο μέσο της λωρίδας κυκλοφορίας

R [m]: Η ακτίνα της καμπύλης του άξονα της λωρίδας κυκλοφορίας

M [m]: Το πλάτος του ελεύθερου χώρου μετρούμενο από το μέσο της εξεταζόμενης λωρίδας κυκλοφορίας

L [m]: Το μήκος της καμπύλης μετρούμενο στον άξονα της εξεταζόμενης λωρίδας κυκλοφορίας

Πίνακας 2.6.2 Στοιχεία Ορατότητας στάσης S_h [6]



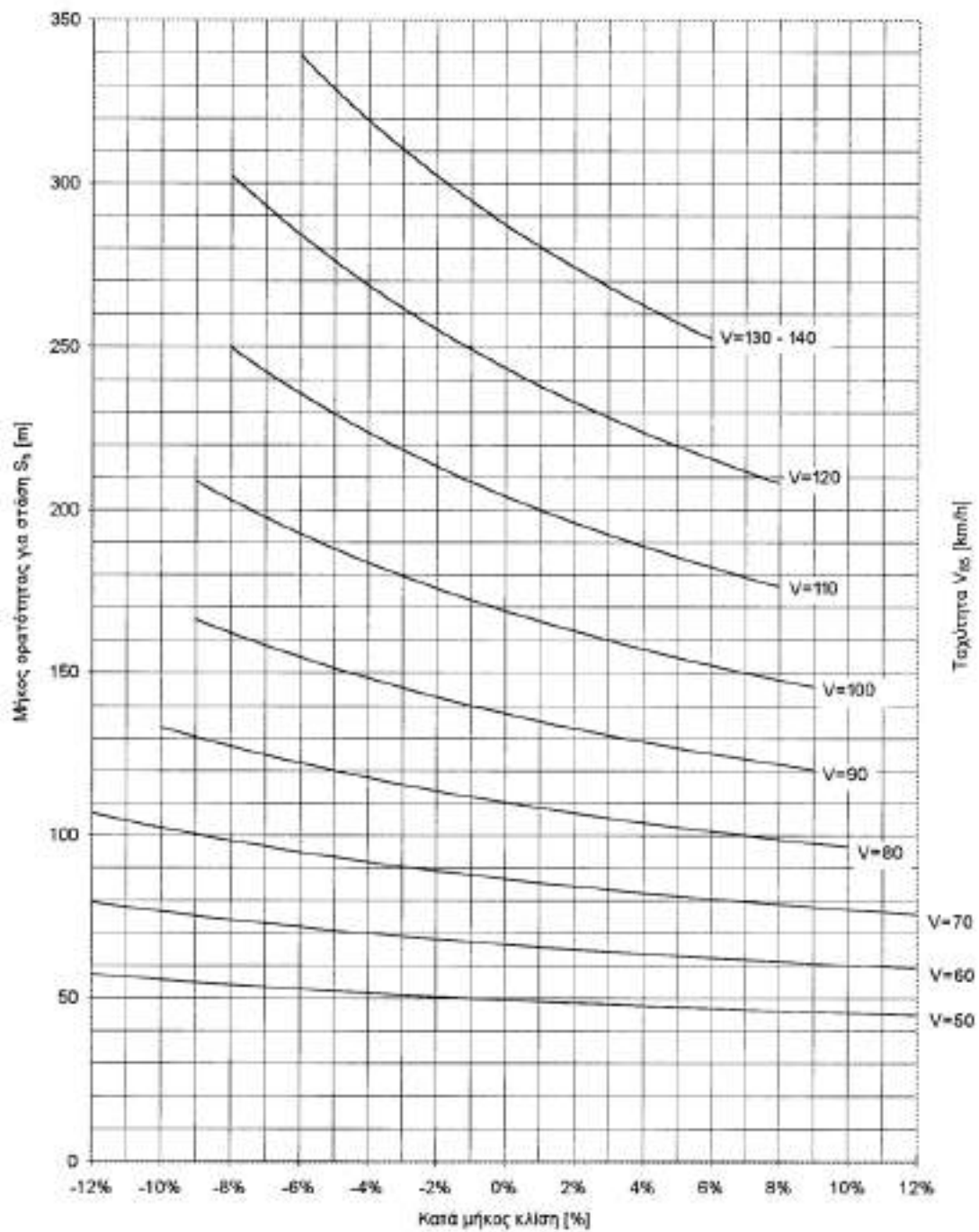
Σχ. 2.4 Στοιχεία Ελέγχου ορατότητας από τον πλευρικό ελεύθερο οπτικών εμποδίων χώρο.[6]

Η σχέση των τριών στοιχείων S_h , R και M αναπαρίσταται στον πίνακα 2.6.2 ο οποίος μπορεί να χρησιμοποιείται ως εξής :

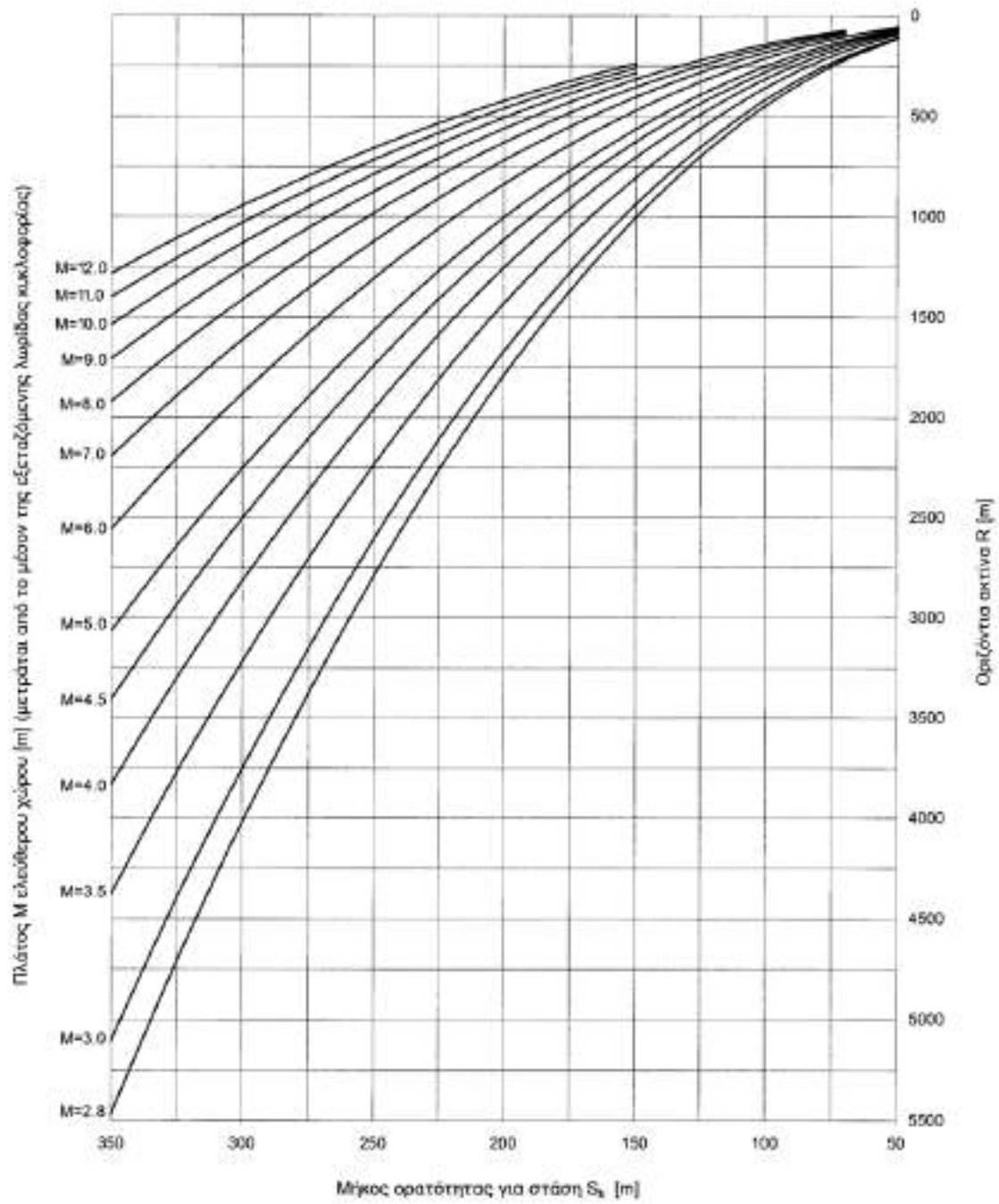
A. από το διάγραμμα ορατότητας (σχήμα 2.6.2) προσδιορίζεται το απαιτούμενο μήκος ορατότητας για στάση S_h για την κατά μήκος κλίση της εξεταζόμενης θέσης με βάση την ταχύτητα V_{856}

B. από το σχήμα 2.6.3 με εισαγωγή του μήκους ορατότητας για στάση S_h που προσδιορίσθηκε, ανευρίσκεται η απαιτούμενη ακτίνα R για δεδομένο πλάτος M πλευρικού ελεύθερου εμποδίων χώρου. Αντίστοιχα μπορεί από το διαθέσιμο πλάτος M και την ακτίνα R της χάραξης να προσδιορίζεται η μέγιστη επιτρεπόμενη ταχύτητα από άποψη ορατότητας για στάση και μόνο.

Έτσι, ορίζεται το επιτρεπόμενο όριο ταχύτητας (πινακίδα) από την τιμή της, στρογγυλευμένη στη δεκάδα που ανήκει (δηλ. όταν προσδιορίζεται μέγιστη επιτρεπόμενη ταχύτητα $V=86\text{km/h}$, τότε αναγράφεται όριο πινακίδας υπό βροχή 80).



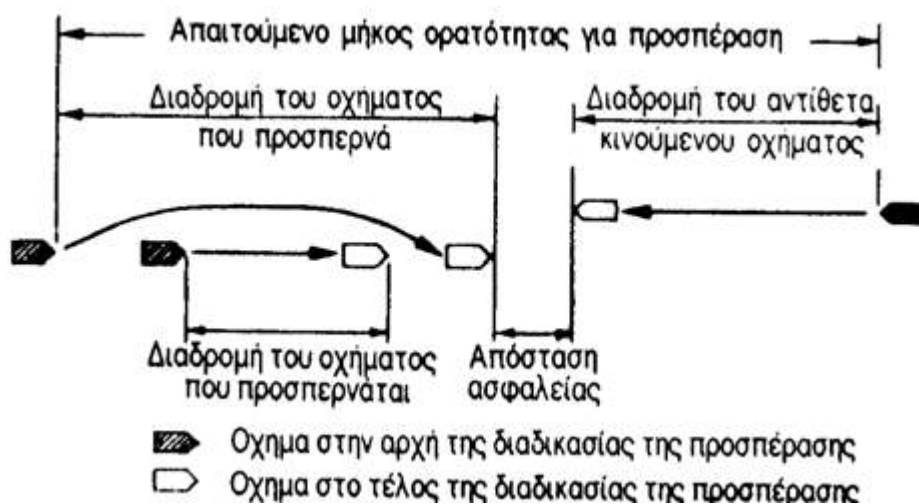
Σχ. 2.5 Απαιτούμενο μήκος ορατότητας για στάση Sh σε υγρό οδόστρωμα [6]



Σχ. 2.6 Σχέση μήκους ορατότητας για στάση S_h με πλάτος ελεύθερου χώρου M[6]

Απαιτούμενο μήκος ορατότητας για προσπέραση S_u

Το μήκος ορατότητας για προσπέραση S_u είναι το μήκος, που απαιτείται για την ασφαλή διαδικασία προσπέρασης εμποδίου ή βραδυπορούντος οχήματος και είναι συνάρτηση της λειτουργικής ταχύτητας V_{85} . Συνίσταται δε από το μήκος που διανύει το όχημα κατά τη διαδικασία της προσπέρασης, το μήκος που διανύει στο ίδιο χρονικό διάστημα το αντίθετα κινούμενο όχημα, και την απόσταση ασφαλείας μεταξύ των δύο αντίθετα κινουμένων οχημάτων στο τέλος της διαδικασίας της προσπέρασης



Σχ. 2.7 Απεικόνιση τυπικής διαδικασίας προσπέρασης προπορευμένου οχήματος

Το απαιτούμενο μήκος ορατότητας για προσπέραση στις οδούς της ομάδας Α παρέχεται από τον πίνακα 2.6.3. Στις οδούς της ομάδας Β πέρα από τις τιμές του πίνακα 2.6.3, πρέπει να ελέγχεται και το μήκος ορατότητας σε ισόπεδους κόμβους.

Σε οδούς των κατηγοριών Α και Β με ενιαία επιφάνεια κυκλοφορίας πρέπει να διατίθενται συνθήκες ορατότητας για προσπέραση συνολικά τουλάχιστον 20-25% του μήκους της οδού με ομοιόμορφη κατανομή των τμημάτων αυτών στο μήκος της οδού[6].

Στα οδικά τμήματα που δεν διαθέτουν επαρκές μήκος ορατότητας για προσπέραση, η διαχωριστική γραμμή μεταξύ των δύο κατευθύνσεων πρέπει να διαμορφώνεται με διπλή συνεχή γραμμή.

V_{85} [km/h]	S_u [m]
60	475
70	500
80	525
90	575
100	625
110	675

Πιν. 2.6.3 Απαιτούμενα μήκη ορατότητας για προσπέραση S_u σε οδούς της ομάδας Α και Β.[4]

Προσδιορισμός υφιστάμενων μηκών ορατότητας

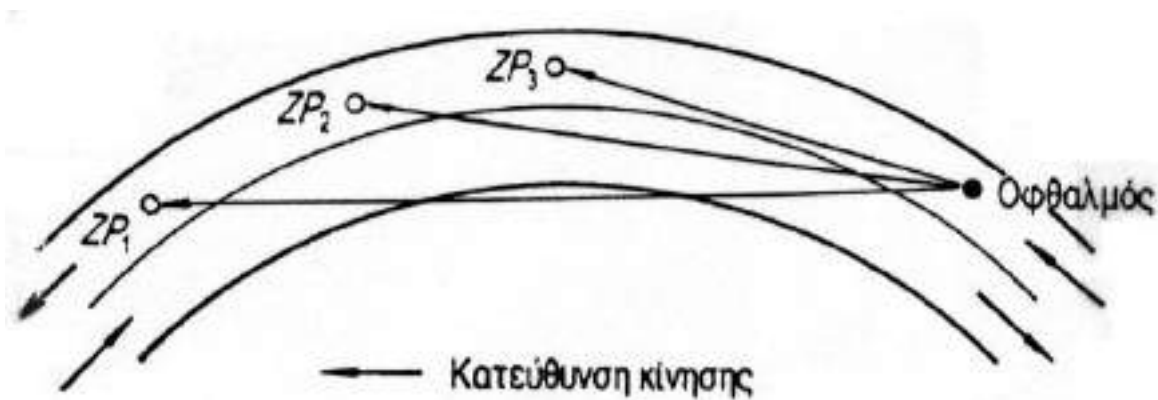
Τα πραγματικά υφιστάμενα μήκη ορατότητας για στάση, συνάντηση και προσπέραση υπολογίζονται σε σχέση με την οριζοντιογραφία, την μηκοτομή, την διατομή και τον περιβάλλοντα χώρο της οδού.

Επειδή η ικανότητα παρατήρησης του ανθρώπινου οφθαλμού είναι περιορισμένη, τα εμπόδια πάνω στην οδό πρέπει να έχουν ένα ελάχιστο μέγεθος, προκειμένου να είναι αντιληπτά και κατανοητά από απόσταση ίση με το μήκος ορατότητας για στάση. Αυτό το ελάχιστο μέγεθος του εμποδίου εξαρτάται από την απόδοση της ανθρώπινης όρασης, τα οπτικά χαρακτηριστικά του εμποδίου και του οδοστρώματος, καθώς επίσης και από τις συνθήκες φωτισμού και τις καιρικές συνθήκες. Τα εμπόδια που είναι αρκετά μεγάλα, ώστε να μην είναι δυνατή η διέλευση των οχημάτων πάνω από αυτά, εν τούτοις είναι δυνατόν να είναι πολύ μικρά, για να αναγνωρίζονται έγκαιρα από τους οδηγούς που κινούνται με την λειτουργική ταχύτητα V_{85} . Επομένως δε μπορεί να προσδιορισθεί ένα εμπόδιο με συγκεκριμένες διαστάσεις και οπτικά χαρακτηριστικά για τον υπολογισμό του μήκους ορατότητας για στάση. Για αυτόν τον λόγο ορίζεται ως ύψος εμποδίου το

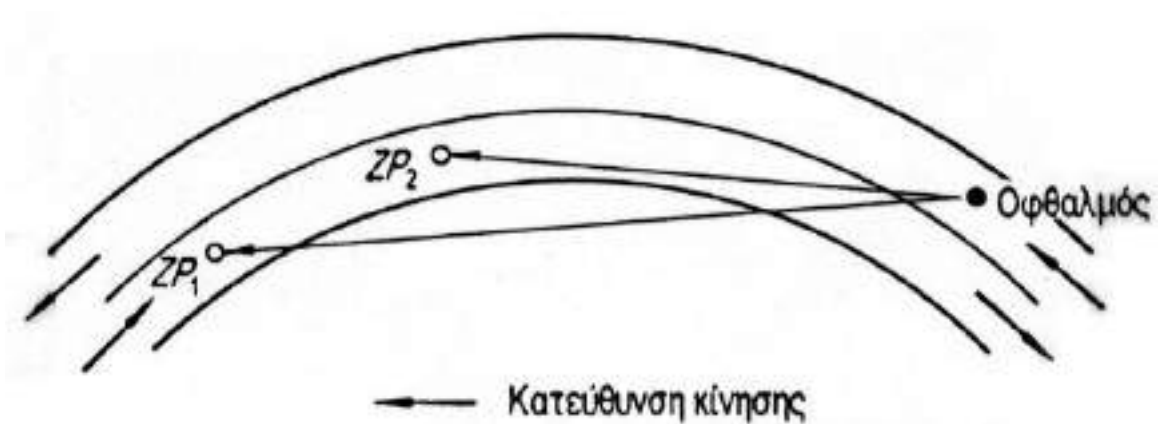
ύψος που προκύπτει με σταθερή οπτική γωνία 5 πρώτα λεπτά της μοίρας για την ταχύτητα V_{85} [6].

Για τον προσδιορισμό των υφιστάμενων μηκών ορατότητας ισχύουν οι παρακάτω κανόνες.

- Ο προσδιορισμός γίνεται σε σχέση με τον οδικό χώρο. Επιπλέον κατά τον προσδιορισμό του μήκους ορατότητας λαμβάνονται υπόψη όλα τα αντικείμενα που υπεισέρχονται στον οδικό και στον παρόδιο χώρο, (όπως πινακίδες κλπ.) καθώς και οι υφιστάμενες και οι προβλεπόμενες φυτεύσεις.
- Ο προσδιορισμός γίνεται για κάθε τύπο μήκους ορατότητας και πάντοτε χωριστά για κάθε κατεύθυνση κυκλοφορίας
- Ο προσδιορισμός των μηκών ορατότητας βασίζεται στα δεδομένα του σχήματος 2.6.2 και στις τιμές του Πίνακα 2.6.3
- Οι θέσεις του οφθαλμού και του εμποδίου αναπαρίστανται στο Σχήμα 2.6.4, τόσο για το μήκος ορατότητας για στάση, όσο και για το μήκος ορατότητας για προσπέραση.



Δυνατή πορεία των οπτικών ακτινών από τον οφθαλμό του οδηγού σε εμπόδιο για τον προσδιορισμό του μήκους ορατότητας για στάση



Δυνατή πορεία των οπτικών ακτινών από τον οφθαλμό του οδηγού σε εμπόδιο για τον προσδιορισμό του μήκους ορατότητας για προσπέραση

ZP1,ZP2,ZP3: Δυνατές θέσεις εμποδίων

Σχ. 2.8 Θέση του οφθαλμού και του εμποδίου για τον προσδιορισμό του μήκους ορατότητας για στάση και του μήκους ορατότητας για προσπέραση

Κεφάλαιο 3

Παρουσίαση Λογισμικού “DromosSplines”

3.1 ΛΟΓΙΣΜΙΚΟ “DROMOSSPLINES”

Το λογισμικό «DromosSplines» που δημιουργήθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας, είναι μία ομάδα προγραμμάτων γραμμένων σε γλώσσα προγραμματισμού Visual Basic και βασισμένων στο σχεδιαστικό πακέτο για ηλεκτρονικό υπολογιστή (CAD) Microstation SE V.5 της Bentley Systems. Το λογισμικό «DromosSplines » ενσωματώνεται και λειτουργεί στο περιβάλλον του Microstation πρόσθετες εντολές-προγράμματα (commands) που εμφανίζονται στον κεντρικό κατάλογο (menu bar) κάτω από την ονομασία «DromosSplines».

Κίνητρο για τη δημιουργία αυτού του πακέτου προγραμμάτων ήταν η επιθυμία για σχεδιασμό ,με βοήθεια του ηλεκτρονικού υπολογιστή και χρήση της εναλλακτικής μεθόδου των καμπύλων ελεύθερης μορφής («SPLINES») τη οριζοντιογραφία μίας οδού. Βασικός στόχος ήταν και ο κατά το δυνατόν περιορισμός στο ελάχιστο των απαιτήσεων για εισαγωγή δεδομένων και παραμέτρων από πλευράς του χρήστη.

Έγινε, επομένως, μία προσπάθεια να είναι όσο το δυνατόν πιο αυτοματοποιημένη η όλη διαδικασία και ο χρήστης να παρεμβαίνει και να αλληλεπιδρά με το λογισμικό μόνο όταν είναι πραγματικά απαραίτητο. Ιδιαίτερη σημασία δόθηκε και στην φιλικότητα του προγράμματος προς το χρήστη, έτσι ώστε να είναι δυνατόν να εποπτεύει εύκολα και

απλά τα αποτελέσματα, προκειμένου να βγάλει τα συμπεράσματά του και να προβαίνει, όταν χρειάζεται, στις κατάλληλες επεμβάσεις.

Οι εντολές-προγράμματα που αποτελούν το πακέτο «DromoSplines » και που εμφανίζονται ως κατάλογος εντολών «DromoSplines» στο Microstation είναι οι παρακάτω:

- Κορυφές οριζοντιογραφίας
- Χάραξη οριζοντιογραφίας
- Καθορισμός μορφολογίας εδάφους
- Μορφολογική κατάταξη δρόμου
- Έλεγχος ακτίνας καμπυλότητας κατά ΟΜΟΕ
- Έλεγχος ορατότητας

3.2 ΠΕΡΙΓΡΑΦΗ & ΛΟΓΙΚΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ «DromoSplines»

3.2.1 Κορυφές Οριζοντιογραφίας

Αρχίζοντας το λογισμικό, ο μηχανικός καλείται να εισάγει τα σημεία από τα οποία επιλέγει να περνάει ο δρόμος. Η διαδικασία γίνεται με την εισαγωγή των καρτεσιανών συντεταγμένων που αντιστοιχούν στις κορυφές της επικείμενης οριζοντιογραφίας οι οποίες αφορούν σε σημεία της καμπύλης Splines και δεν ταυτίζονται με τις κορυφές της πολυγωνικής.

Όνομα	X	Y
1	1.000000	12.000000
2	53.000000	4.000000
3	128.000000	-12.000000
4	156.000000	7.000000
5	351.000000	10.000000
6	370.337900	2.099600
7	396.870000	-36.920000
8	457.220000	-54.060000
9	602.143700	35.680000
10	762.870000	-49.500000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000
	0.000000	0.000000

Επόμενη

Προηγούμενη

1η

Τελευταία

Παραβολή

Συγκεκριμένη

Πληροφορίες

Πριν Απο Την Κορυφή

Όνομασία Κορυφής

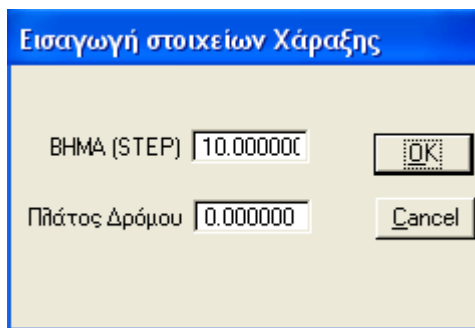
OK

Εξοδος

Εικ. 3.1 Κορυφές Οριζοντιογραφίας

Πατώντας το κουμπί OK το λογισμικό αποθηκεύει στο αρχείο **fm.hor** τις κορυφές και είναι έτοιμο να συνεχίσει στην χάραξη της οδού.

3.2.2 Χάραξη οριζοντιογραφίας



Εικ. 3.2 Εισαγωγή Στοιχείων Χάραξης

Το λογισμικό αυτό θα μπορούσε να χαρακτηριστεί ως ένα από τα κυριότερα προγράμματα, ίσως το βασικότερο όλων, του πακέτου προγραμμάτων «DromosSplines », επειδή είναι ακριβώς αυτό το οποίο τελικώς σχεδιάζει στην οθόνη του υπολογιστή τη οριζοντιογραφία ης οδού. Αρχικά, διαβάζει το αρχείο (fm.hor) το οποίο περιέχει τις συντεταγμένες των κορυφών οριζοντιογραφίας που έχει δώσει νωρίτερα ο χρήστης. Στη συνέχεια, καλεί τον χρήστη να πληκτρολογήσει το επιθυμητό βήμα (STEP) για τον υπολογισμό όλων των σημείων και το σχεδιασμό της οριζοντιογραφίας του δρόμου καθώς και το επιθυμητό πλάτος της οδού και τα καταχωρεί στο αρχείο (XLSTEP.dat). Το βήμα αυτό θα καθορίσει το πλήθος των συνολικών σημείων της οριζοντιογραφίας που θα υπολογιστούν από το λογισμικό. Όσο πιο μεγάλο είναι το βήμα, τόσο λιγότερα σημεία θα υπολογιστούν και επομένως τόσο πιο γρήγορα θα εκτελεστούν όλοι οι υπολογισμοί και για το σχεδιασμό αλλά και για την πραγματοποίηση των ελέγχων αργότερα. Από την άλλη, όσο πιο μικρό είναι το βήμα, τόσο πιο λεπτομερής και ακριβής θα είναι η οριζοντιογραφία του δρόμου. Αν για παράδειγμα δοθεί ως επιθυμητό βήμα η μονάδα, το λογισμικό θα υπολογίσει σημεία ανά ένα μέτρο για το σχεδιασμό της οριζοντιογραφίας του δρόμου. Όταν το λογισμικό υπολογίσει όλα τα σημεία και σχεδιάσει την οριζοντιογραφία με καμπύλη SPLINE, καταχωρεί τα δεδομένα που έχουν υπολογισθεί για όλα τα σημεία της οριζοντιογραφίας (την τετμημένη των σημείων x, την τεταγμένη των σημείων y, την πρώτη παράγωγο $y'(x)$, τη δεύτερη παράγωγο $y''(x)$) στο αρχείο (sp3.dat).

3.2.3 Καθορισμός μορφολογίας εδάφους

Αφού σχεδιάστηκε η οριζοντιογραφία, ο χρήστης καλείται να δηλώσει στο λογισμικό, δημιουργώντας πολύγωνα που περικλείουν τις λοφώδεις-ορεινές περιοχές, τη μορφολογία του εδάφους στην περιοχή του έργου. Η δημιουργία του πολυγώνου μορφολογίας γίνεται εισάγοντας στον πίνακα των Κορυφών Πολυγώνου Μορφολογίας Εδάφους (morf1.bat) τις συντεταγμένες του επιθυμητού πολυγώνου.

Προσοχή, οι συντεταγμένες της πρώτης και της τελευταίας κορυφής πρέπει να είναι ίδιες για να αναγνωρίσει το λογισμικό το κλείσιμο του εκάστοτε πολυγώνου.

Όνομα	X	Y
1	127.00000	44.00000
2	225.00000	-23.00000
3	88.00000	-63.00000
4	127.00000	44.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000
	0.00000	0.00000

Επόμενη
Προηγούμενη
1η
Τελευταία
Παραμετοποίηση

OK

Εξοδος

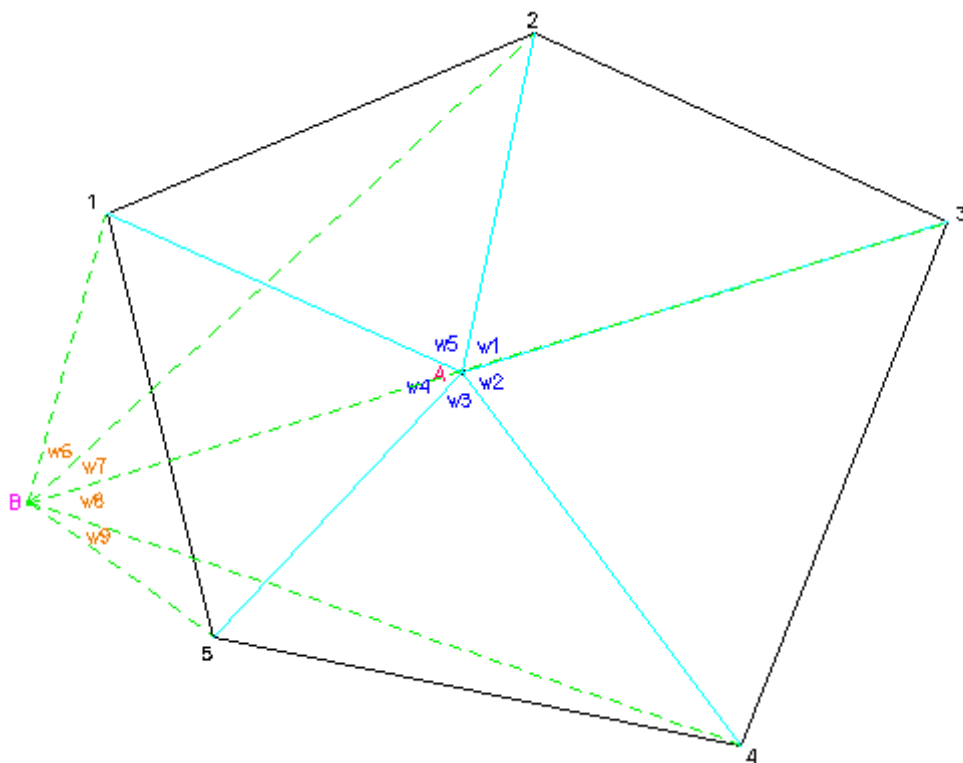
Προσοχή η πρώτη με την τελευταία κορυφή πρέπει να έχουν ίδιες συντεταγμένες

Εικ. 3.3 Κορυφές Πολυγώνου Μορφολογίας Λοφώδους Ορεινού Εδάφους

3.2.4 Μορφολογική κατάταξη του δρόμου

Ο χρήστης «τρέχοντας» το υποπρόγραμμα morfgraf.bas το λογισμικό χαράσσει από όλα τα κορυφαία σημεία του δρόμου ευθείες προς όλες τις κορυφές του μορφολογικού πολυγώνου που κατασκευάστηκε πριν. Στην συνέχεια τρέχοντας το tomi.bas το λογισμικό μετρώντας το άθροισμα των γωνιών που δημιουργούνται μεταξύ των προαναφερθεισών ευθειών αντιλαμβάνεται, αν το εκάστοτε σημείο ανήκει εντός του μορφολογικού πολυγώνου.

Γεωμετρική παραπομπή: Αν ένα σημείο βρίσκεται εντός ενός πολυγώνου, τότε το άθροισμα των γωνιών που προκύπτουν από την ένωση με ευθείες του σημείου με όλες τις κορυφές του πολυγώνου είναι ίσο με 360° .



Σχήμα 3.1

Εν προκειμένω A : $w1+w2+w3+w4+w5 = 360^\circ$

Ενώ B : $w6+w7+w8+w9 < 360^\circ$

Αφού λοιπόν το λογισμικό κατατάζει μορφολογικά τα σημεία του δρόμου, στη συνέχεια προσθέτει άλλη μια παράμετρο στο ήδη υπάρχον αρχείο `sp3.dat` που αφορά στη μορφολογία.

3.2.5 Έλεγχος ακτίνας καμπυλότητας κατά ΟΜΟΕ

Η διαδικασία που ακολουθήθηκε στο λογισμικό για να βρεθεί η ακτίνα καμπυλότητας σε κάθε σημείο του δρόμου προκύπτει από την εφαρμογή του τύπου:

$$\rho(t) = \frac{|1 + f'(t)^2|^{3/2}}{|f''(t)|} \quad (\Sigma 3.1)$$

Παραγωγίζεται η αρχική συνάρτηση του άξονα του δρόμου μέχρι τη δεύτερη παράγωγο και στη συνέχεια εφαρμόζεται η σχέση $\Sigma 3.1$.

Στη συνέχεια παρατίθεται το απόσπασμα του κώδικα στο οποίο φαίνεται ο υπολογισμός.

```
YL(j2)=XM(j-1)*(X(j)-XL(j2))^3/(6*H(j-1))+XM(j)*(XL(j2)-x(j-1))^3/(6*H(j-1))+  
(y(j-1)-(XM(j-1)*H(j-1)^2/6))*(x(j)-XL(j2))/H(j-1)+(y(j)-XM(j)*H(j-1)^2/6)*(XL(j2)-  
x(j-1))/H(j-1)
```

```
YDL(j2)=-XM(j-1)*(X(j)-XL(j2))^2/(2*H(j-1))+XM(j)*(XL(j2)-x(j-1))^2/(2*H(j-  
1))+(y(j)-y(j-1))/H(j-1)-(XM(j)-XM(j-1))*H(j-1)/6
```

```
YDDL(j2)=XM(j-1)*(X(j)-XL(j2))/H(j-1)-XM(j)*(XL(j2)-x(j-1))/H(j-1)
```

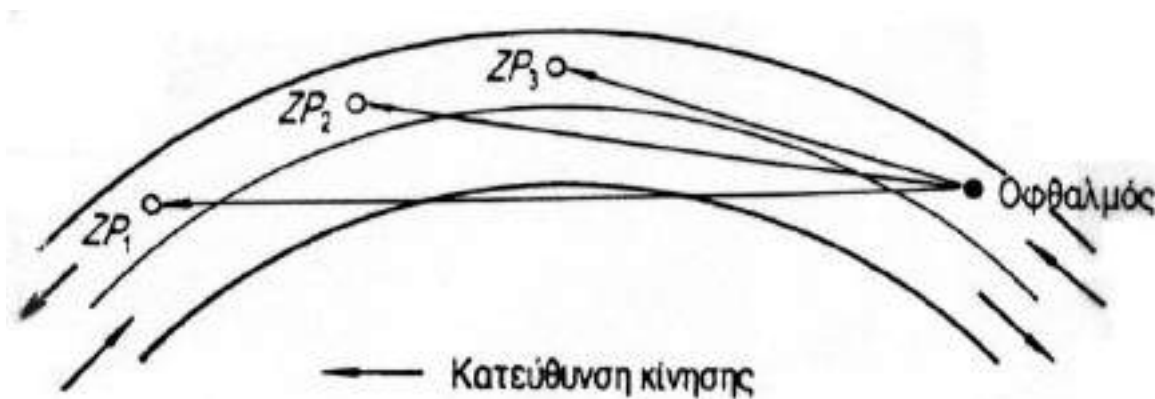
```
if YDDL(j2)<>0 then
```

$$\text{CURV}(j2)=(((1+\text{YDL}(j2)^2)^{1.5}/\text{YDDL}(j2))^2)^{(1/2)}$$

3.2.6 Έλεγχος ορατότητας κατά ΟΜΟΕ

Ο συγκεκριμένος έλεγχος πραγματοποιείται για ορατότητα στάσης και στάθμευσης. Η λογική της συγκεκριμένης μακροεντολής βασίζεται στην αναγνώριση της οδού ως ένα εκτεταμένο πολύγωνο. Το πολύγωνο αποτελείται από τα αριστερά και δεξιά όρια του δρόμου. Έχοντας ως δεδομένο αυτό, στη μέση κάθε ρεύματος κατεύθυνσης εντοπίζονται σημεία καθ' όλο το μήκος της οδού ανά ένα μέτρο.

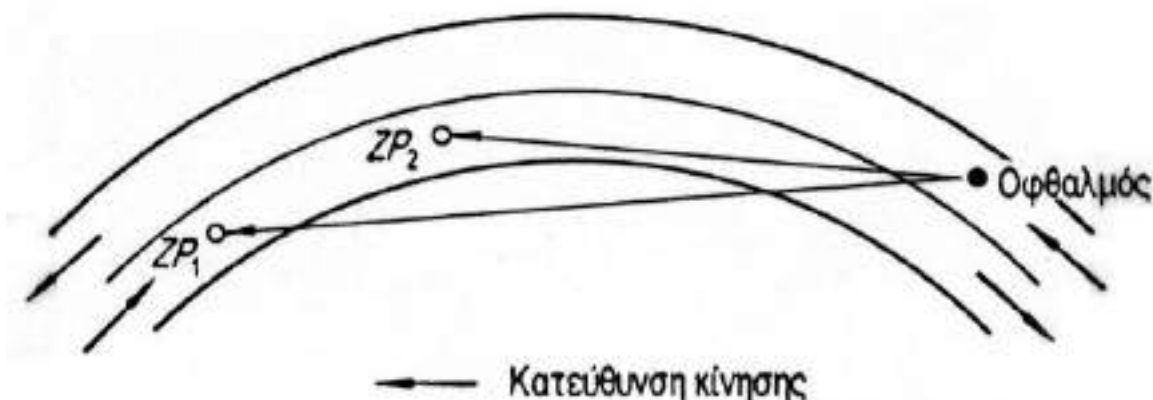
- **Ορατότητα για στάση.**



Σχήμα 3.2

Όπως φαίνεται στο παραπάνω σχήμα, στην περίπτωση της Ορατότητας σε στάση, το λογισμικό ενώνει με ευθεία το κάθε σημείο σε ένα ρεύμα κατεύθυνσης με όλα τα σημεία που το ακολουθούν χιλιομετρικά στο ίδιο ρεύμα, μέχρις ότου αντιληφθεί ότι κάποιο σημείο των ευθειών που δημιουργεί, βρίσκεται εκτός του πολυγώνου της οδού. Τότε αυτόματα η διαδικασία σταματά και το μήκος της αμέσως προηγούμενης γραμμής (που βρισκόταν εξ' ολοκλήρου μέσα στο πολύγωνο της οδού) καταγράφεται ως πραγματοποιούμενη Ορατότητα στάσης για το εκάστοτε σημείο.

- Ορατότητα για προσπέραση



Σχήμα 3.3

Όπως φαίνεται στο παραπάνω σχήμα, στην περίπτωση της Ορατότητας σε προσπέραση, το λογισμικό ενώνει με ευθεία το κάθε σημείο σε ένα ρεύμα κατεύθυνσης με όλα τα σημεία που το ακολουθούν χιλιομετρικά στο αντίθετο ρεύμα, μέχρις ότου αντιληφθεί ότι κάποιο σημείο των ευθειών που δημιουργεί, βρίσκεται εκτός του πολυγώνου της οδού. Τότε αυτόματα η διαδικασία σταματά και το μήκος της αμέσως προηγούμενης γραμμής (που βρισκόταν εξ' ολοκλήρου μέσα στο πολύγωνο της οδού) καταγράφεται ως πραγματοποιούμενη Ορατότητα προσπέρασης για το εκάστοτε σημείο.

Αφού βρεθεί η πραγματοποιούμενη ορατότητα για κάθε σημείο, κατασκευάζεται διάγραμμα στο οποίο φαίνονται όλες οι τιμές που προέκυψαν από την προηγούμενη διαδικασία, και ταυτόχρονα γίνεται και σ'θγκριση με τις απαιτούμενες κατά Ο.Μ.Ο.Ε.

Κεφάλαιο 4

Χρησιμοποιούμενες Εξισώσεις

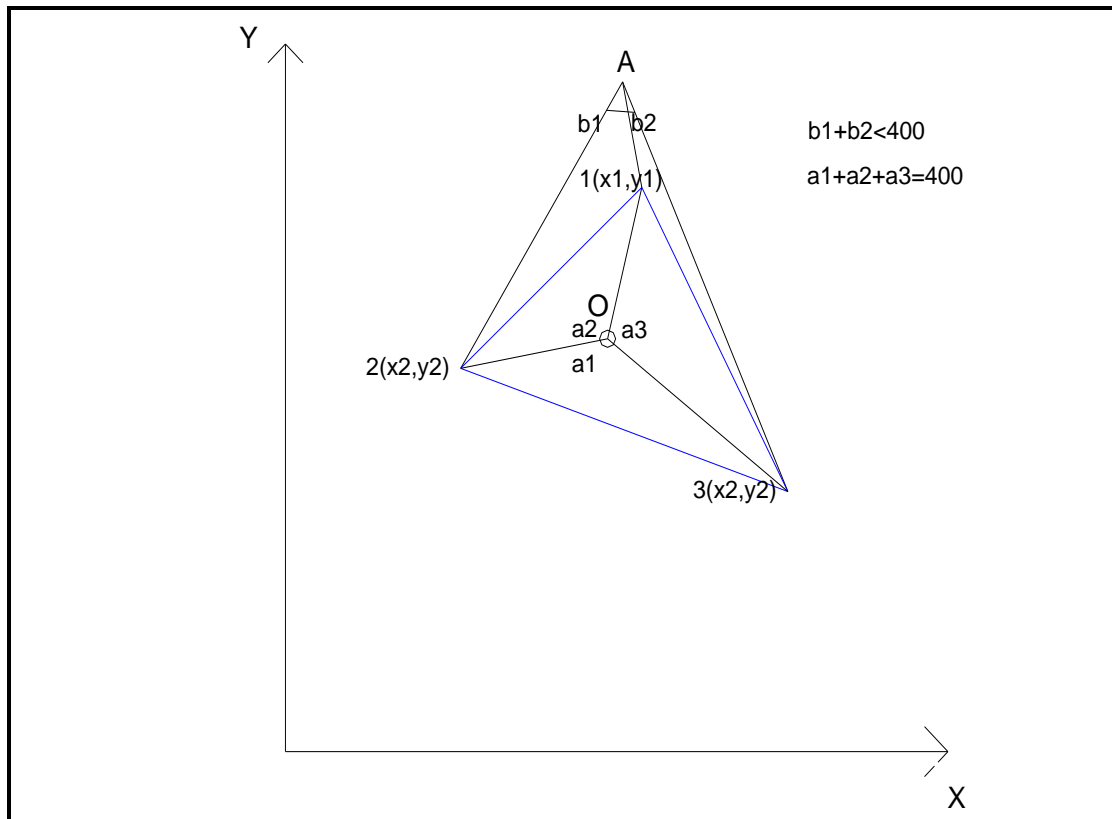
4.1 ΕΞΙΣΩΣΗ ΓΩΝΙΩΝ

Η εξίσωση των γωνιών είναι μια εξίσωση, η οποία κάνει την ίδια δουλειά με αυτή της εξίσωσης εγκλωβισμού, δηλαδή μας δίνει ποιο σημείο είναι εντός και ποιο είναι εκτός μιας περιοχής.

Ένα σημείο σχηματίζει με δύο άλλα σημεία κάποια γωνία, που είναι δυνατόν να υπολογιστεί με τη βοήθεια του νόμου των συνημίτονων αν είναι γνωστές οι συντεταγμένες των σημείων αυτών. Οι αποστάσεις μεταξύ των σημείων καθώς και οι πλευρές του τριγώνου υπολογίζονται με απλές μαθηματικές σχέσεις, οι οποίες παρουσιάζονται παρακάτω.

Στη συγκεκριμένη διπλωματική, έχουμε ένα πολύγωνο τριών κορυφών με αποτέλεσμα ένα τυχαίο σημείο σχηματίζει τρεις, διαδοχικές γωνίες με τις κορυφές του πολυγώνου αυτού

Ένα σημείο βρίσκεται εντός της περιοχής του πολυγώνου όταν το άθροισμα των γωνιών που σχηματίζει το σημείο αυτό με τις διαδοχικές κορυφές του πολυγώνου, είναι ίσο με $360^\circ (400^\circ)$. Αντιθέτως, ένα σημείο βρίσκεται εκτός της περιοχής του πολυγώνου όταν το άθροισμα των γωνιών που σχηματίζει το σημείο αυτό με τις διαδοχικές κορυφές του πολυγώνου, είναι μικρότερο από 360° .



Σχήμα. 4.1

Μεταβλητές:

- $s1, s2, s3$ τα μήκη των πλευρών του τριγώνου
- $d1, d2, d3$ οι απόσταση του σημείου από τις κορυφές
- $a1, a2, a3$ οι σχηματιζόμενες γωνίες
- xw, yw, zw οι συντεταγμένες του σημείου
- $x1, y1, z1$ οι συντεταγμένες των κορυφών

Σχέσεις:

$$s1 = \sqrt{(x1-x2)^2 + (y1-y2)^2}$$

$$s2 = \sqrt{(x2-x3)^2 + (y2-y3)^2}$$

$$s3 = \sqrt{(x3-x1)^2 + (y3-y1)^2}$$

$$d2 = \sqrt{(x1-xw)^2 + (y1-yw)^2}$$

$$d3 = \sqrt{(x2-xw)^2 + (y2-yw)^2}$$

$$d1 = \sqrt{(x3-xw)^2 + (y3-yw)^2}$$

$$a1 = \frac{(d2^2 + d3^2 - s1^2)}{(2 \times d2 \times d3)}$$

$$a2 = \frac{(d3^2 + d1^2 - s2^2)}{(2 \times d1 \times d3)}$$

$$a3 = \frac{(d1^2 + d2^2 - s3^2)}{(2 \times d1 \times d2)}$$

$$a = a1 + a2 + a3$$

4.2 ΕΞΙΣΩΣΗ ΤΟΜΗΣ

Η εξίσωση τομής αποτελείται ουσιαστικά από δύο εξισώσεις από τις οποίες εξάγονται οι συντεταγμένες της εκάστοτε τομής δύο ευθειών. Βασική προϋπόθεση είναι να είναι οι συντεταγμένες των κορυφών κάθε ευθείας γνωστές. Η εξίσωση αυτή χρησιμοποιείται στην περίπτωση τομής του άξονα του διαδρόμου με τα δημιουργημένα τρίγωνα-επιφάνειες, στην αντίστοιχη του εγκάρσιου άξονα στο διάδρομο και στην κατασκευή των ορυγμάτων-επιχωμάτων στις διατομές.

Έστω λοιπόν, ότι αναζητείται η τομή δύο ευθειών: (12) και (34). Γνωρίζοντας τις συντεταγμένες τους $x1, y1, x2, y2, x3, y3, x4, y4$ υπολογίζονται οι λόγοι $(y1 - y2) / (x1 - x2)$ και $(y3 - y4) / (x3 - x4)$ οι οποίοι αποτελούν ουσιαστικά την κλίση της ευθείας. Οι λόγοι είναι πάντα θετικοί γιατί αν άλογα με τη θέση των σημείων ως προς τον άξονα x

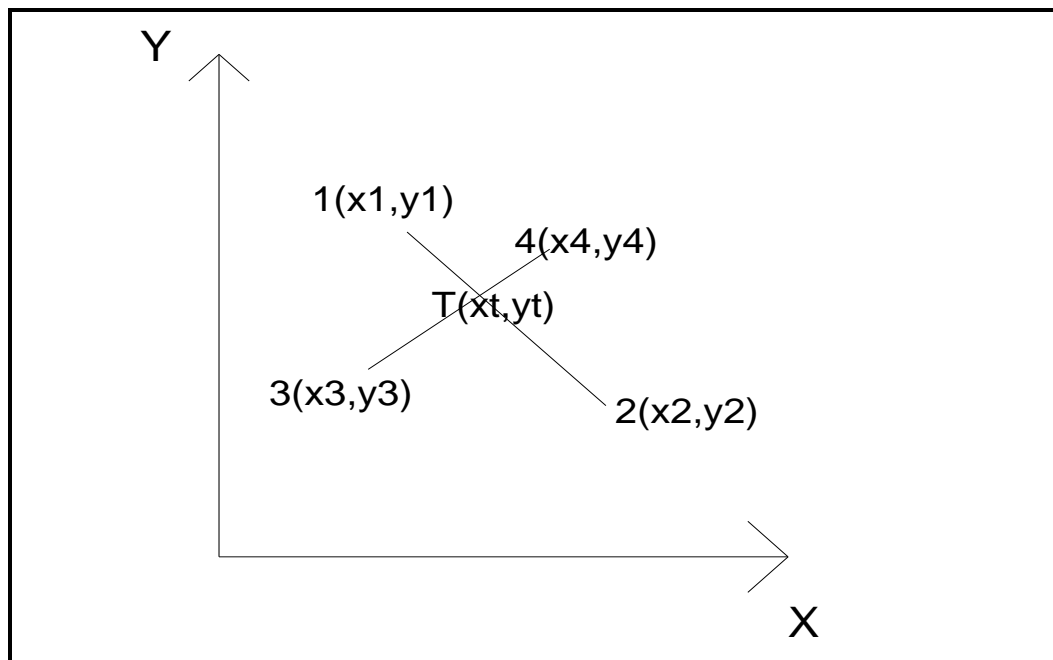
μεταβάλλονται αριθμητικά οι συντεταγμένες (δηλ. αν το $\chi_1 < \chi_2$ τότε $\chi_1 = \chi_1$ και $\chi_2 = \chi_2$ αλλιώς $\chi_1 = \chi_2$ και $\chi_2 = \chi_1$).

Έχοντας υπολογίσει τους λόγους οι συντεταγμένες της τομής δίνονται από τις σχέσεις:

$$x_t = \frac{y_1 - y_3 + a_2 \times x_3 - a_1 \times x_1}{(a_2 - a_1)}, \text{ όπου } a_1, a_2 \text{ οι λόγοι.}$$

$$y_t = y_1 + a_1 \times (x_t - x_1)$$

Έτσι υπολογίζονται οι συντεταγμένες της τομής. Συχνά βέβαια αναζητείται το υψόμετρο της τομής, το οποίο δίνεται με τη βοήθεια της εξίσωση επιπέδου όπως αυτή περιγράφεται παρακάτω.



Σχ. 4.2

4.3 ΕΞΙΣΩΣΗ ΕΠΙΠΕΔΟΥ

Ως γνωστόν, ένα επίπεδο ορίζεται από τουλάχιστον τρία σημεία. Η γενική μορφή της εξίσωσης του επιπέδου είναι :

$$z=ax+by+c \quad (*)$$

όπου a, β, ϵ σταθερές και χ, γ, ζ οι συντεταγμένες των σημείων του επιπέδου.

Αν υπάρχουν τρία σημεία $A(x_1, y_1, z_1)$, $B(x_2, y_2, z_2)$ και $\Gamma(x_3, y_3, z_3)$ τότε αυτά ορίζουν ένα επίπεδο, όταν επαληθεύουν την εξίσωση του επιπέδου (*). Παρακάτω θα προσπαθήσουμε να ορίσουμε τις σταθερές a, b, c με τα γενικά σημεία A, B και Γ και συνεπώς με τις αντίστοιχες συντεταγμένες που θα δίνονται με τη μορφή μεταβλητών.

Επομένως για τα σημεία A, B και Γ η εξίσωση (*) μας δίνει :

$$z_1=ax_1+by_1+c \quad (1)$$

$$z_2=ax_2+by_2+c \quad (2)$$

$$z_3=ax_3+by_3+c \quad (3)$$

Συνεπώς έχουμε :

$$(2) \rightarrow c = z_2 - ax_2 - by_2 \quad (4)$$

οπότε η (2) γίνεται :

$$(3) \xrightarrow{(4)} z_3 = ax_3 + by_3 + z_2 - ax_2 - by_2 \rightarrow$$

$$\xrightarrow{x_2=x_1} a = \frac{(z_2 - z_3) - b \times (y_2 - y_3)}{x_2 - x_3} \quad (5)$$

σε συνδυασμό με τις δύο παραπάνω σχέσεις και την (1), βρίσκουμε τη μεταβλητή b , οπότε :

$$\begin{aligned}
 (1) & \xrightarrow{x_2=x_1} z_1 = ax_1 + by_1 + z_2 - ax_2 - by_2 \rightarrow \\
 & \rightarrow (z_2 - z_1) + a \times (x_1 - x_2) + b \times (y_1 - y_2) \rightarrow \\
 & \xrightarrow{(5)} (z_2 - z_1) + \frac{(z_2 - z_3) - b \times (y_2 - y_3)}{x_2 - x_3} \times (x_1 - x_2) + b \times (y_1 - y_2) = 0 \rightarrow \\
 & \rightarrow (z_2 - z_1) \times (x_2 - x_3) + (z_2 - z_3) \times (x_1 - x_2) - \\
 & \quad - b \times (y_2 - y_3) \times (x_1 - x_2) + b \times (y_1 - y_2) \times (x_2 - x_3) = 0 \rightarrow \\
 & \rightarrow b \times [(y_1 - y_2) \times (x_2 - x_3) - (y_2 - y_3) \times (x_1 - x_2)] + \\
 & \quad + (z_2 - z_1) \times (x_2 - x_3) + (z_2 - z_3) \times (x_1 - x_2) = 0 \rightarrow \\
 & \rightarrow b = \frac{(z_1 - z_2) \times (x_2 - x_3) - (z_2 - z_3) \times (x_1 - x_2)}{(y_1 - y_2) \times (x_2 - x_3) - (y_2 - y_3) \times (x_1 - x_2)} \quad (1^*)
 \end{aligned}$$

Από τη σχέση (5) σε συνδυασμό με την (1*) προκύπτει το a , οπότε:

$$\begin{aligned}
 (5) \rightarrow a &= \frac{(z_2 - z_3) - b \times (y_2 - y_3)}{(x_2 - x_3)} \xrightarrow{(1^*)} \\
 & \rightarrow a = \frac{(z_2 - z_3) - \frac{(z_1 - z_2) \times (x_2 - x_3) - (z_2 - z_3) \times (x_1 - x_2)}{(y_1 - y_2) \times (x_2 - x_3) - (y_2 - y_3) \times (x_1 - x_2)} \times (y_2 - y_3)}{(x_2 - x_3)} \quad (2^*)
 \end{aligned}$$

Από τη σχέση (4) σε συνδυασμό με την (2*) προκύπτει το c , οπότε:

$$(4) \rightarrow c = z_2 - ax_2 - by_2 \xrightarrow{(1^*), (2^*)}$$

$$\rightarrow C=Z_1 - \frac{(Z_2-Z_3) - \frac{(Z_1-Z_2) \times (X_2-X_3) - (Z_2-Z_3) \times (X_1-X_2)}{(Y_1-Y_2) \times (X_2-X_3) - (Y_2-Y_3) \times (X_1-X_2)} \times (Y_2-Y_3)}{(X_2-X_3)} \times X_1 - \frac{(Z_1-Z_2) \times (X_2-X_3) - (Z_2-Z_3) \times (X_1-X_2)}{(Y_1-Y_2) \times (X_2-X_3) - (Y_2-Y_3) \times (X_1-X_2)} \times Y_1 \quad (3^*)$$

Οι σταθερές a, b, c της εξίσωσης του επιπέδου ορίστηκαν πλήρως από τις συντεταγμένες των σημείων που ορίζουν ένα επίπεδο. Συνεπώς, ένα τυχαίο σημείο $O(x_0, y_0, z_0)$ βρίσκεται στο επίπεδο όταν η εξίσωση του επιπέδου επαληθεύεται από τις συντεταγμένες του.

Η εξίσωση του επιπέδου συμβάλλει στην ανεύρεση των υψομέτρων των διαφόρων σημείων των επιφανειών.

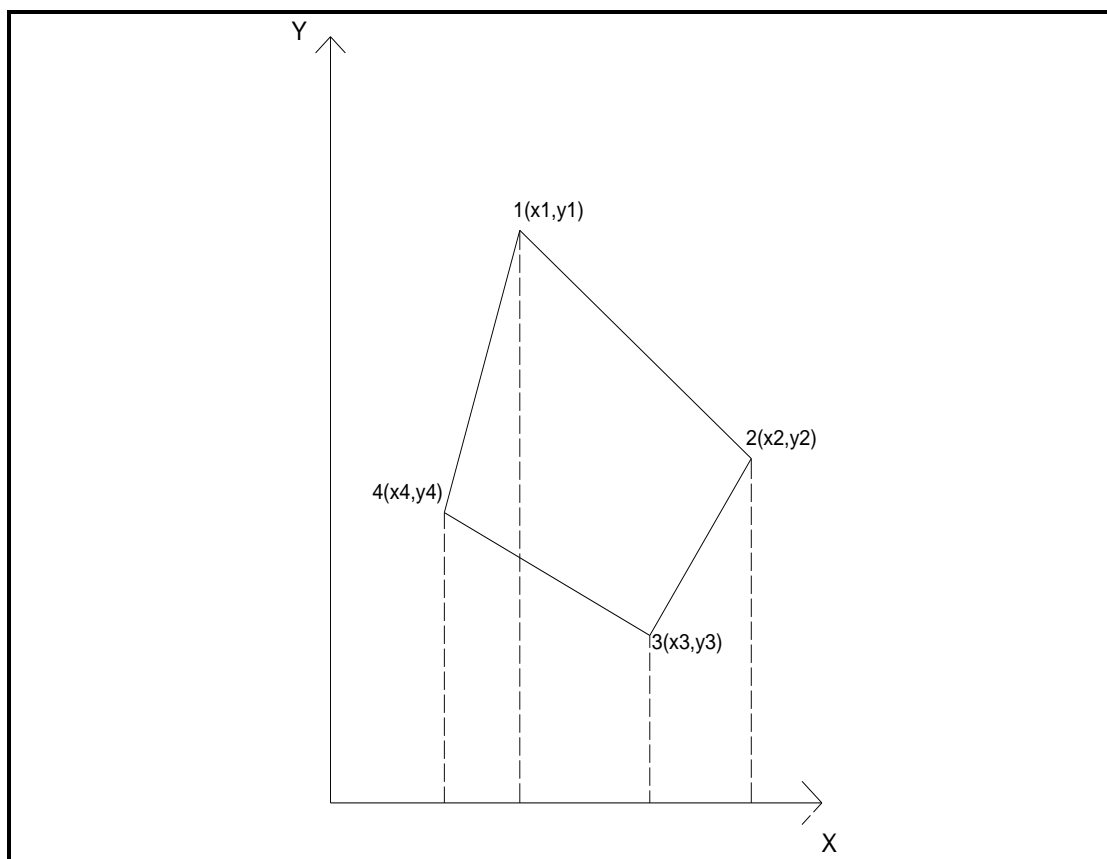
4.4 ΕΞΙΣΩΣΗ ΕΓΚΛΩΒΙΣΜΟΥ

Ένα πρόβλημα που χρήζει αντιμετώπισης στην πορεία του προγράμματος ήταν ο τρόπος μέσω του οποίου ανιχνεύεται πιο σημείο είναι μέσα σε μια περιοχή και πιο είναι εκτός περιοχής. Στη συγκεκριμένη διπλωματική εργασία το πρόβλημα υφίσταται σαρώνοντας τα σημεία της περιοχής στη οποία βρίσκονται οι υποψήφιες θέσεις του διαδρόμου όπως αυτές καθορίζονται από το χρήστη. Επίσης η ζητούμενη εξίσωση χρησιμοποιείται στην κατασκευή των ορυγμάτων, επιχωμάτων των διατομών. Δημιουργείται λοιπόν ένας κάνναβος στον οποίο σαρώνονται τα σημεία.

Η δισδιάστατη μορφή του κάνναβου είναι ένα ορθογώνιο, κάθε φορά, ψηφιοποιημένο με άπειρα σημεία σε συγκεκριμένη απόσταση (βήμα) μεταξύ τους. Στην διπλωματική αυτή εργασία δημιουργείται ο ίδιος κάνναβος για κάθε επιφάνεια ή ζώνη του αεροδρομίου.

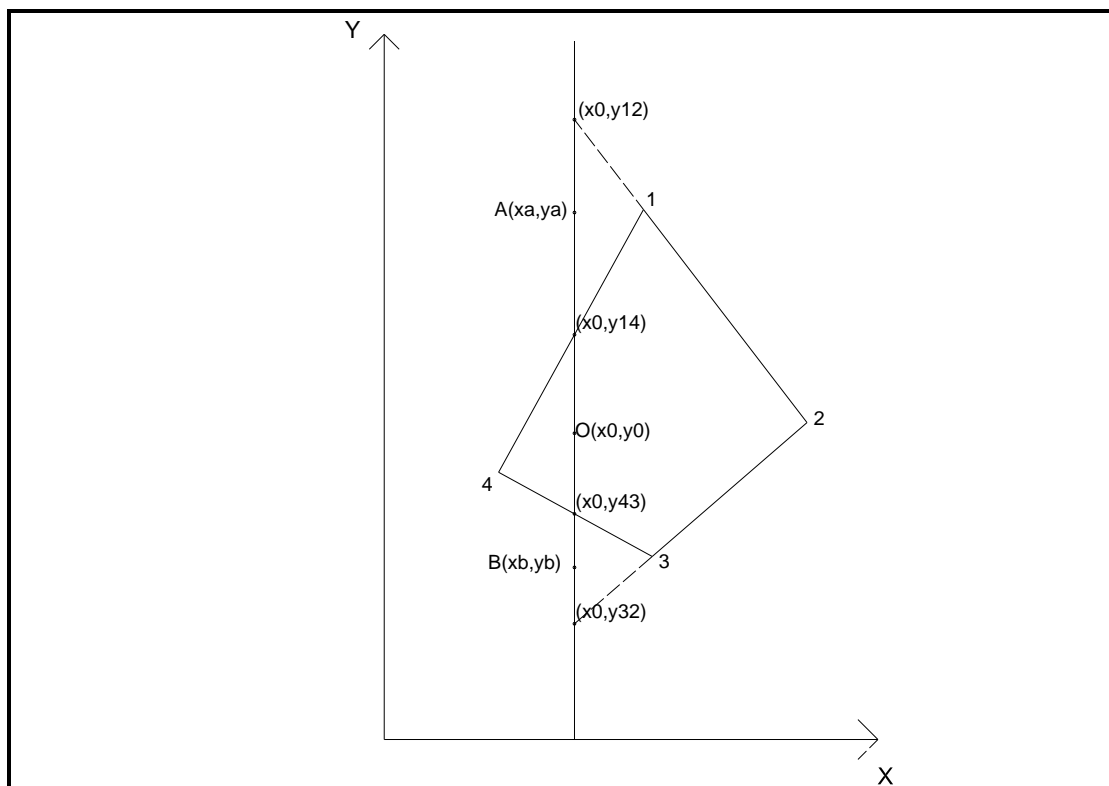
Συνεπώς, είναι απαραίτητη μία σχέση που να εξετάζει αν ένα συγκεκριμένο σημείο κάθε φορά, σε σχέση με τις συντεταγμένες του (x,y) στο χώρο, βρίσκεται μέσα σε μια ορισμένη περιοχή από τρία ή τέσσερα σημεία.

Σε ένα διαζονικό σύστημα συντεταγμένων (x,y) ένα πολυγωνικό σχήμα, με γνωστές τις συντεταγμένες των κορυφών του, έχει δεδομένη θέση ως προς στους άξονες συντεταγμένων.



Σχ. 4.3

Ένα τυχαίο σημείο στο χώρο με γνωστές τις συντεταγμένες του μπορεί να είναι μέσα στην επιφάνεια που ορίζει το πολύγωνο αυτό ή έξω από αυτήν. Όπως φαίνεται στο σχήμα (7.4.2), το τυχαίο σημείο $O(x_0,y_0)$ είναι μέσα στο πολύγωνο, ενώ τα σημεία $A(x_a,y_a)$ και $B(x_b,y_b)$ βρίσκονται έξω από αυτό, ενώ και τα τρία αυτά σημεία έχουν την ίδια συντεταγμένη x , $x_0=x_a=x_b$.



Σχ. 4.4

Το πρόβλημα συνίσταται στο ποιο από τα τρία σημεία είναι μέσα και ποια έξω από το πολύγωνο. Αν είναι σχεδιασμένα είναι εύκολο να τα βρούμε, αλλά ζητείται η σχέση εκείνη που με γνωστές τις συντεταγμένες των, υπό εξέταση, σημείων θα δίνει τα σημεία που είναι εντός ή εκτός του πολυγώνου.

Δεδομένου ότι είναι γνωστές οι συντεταγμένες των τεσσάρων (ή τριών) κορυφών του πολυγώνου εύκολα ορίζονται οι εξισώσεις των ευθειών των πλευρών του. Έχοντας γνωστή τη συντεταγμένη x των σημείων ($x_0 = x_a = x_b$), καταλήγουμε στην εξίσωση της ευθείας που είναι κάθετη στον άξονα των x και που ισχύει για όλα αυτά τα σημεία. Πρόκειται για την ευθεία με εξίσωση $x = x_0$.

Τα σημεία τομής της ευθείας του σημείου με τις ευθείες των πλευρών των κορυφών είναι γνωστά, αν στις εξισώσεις των ευθειών των πλευρών θέσουμε ως x το x_0 , που είναι τιμή.

Κάθε φορά ζητούνται δύο τομές που γίνονται με τις ευθείες των πλευρών. Οι ζητούμενες τομές κάθε φορά βρίσκονται με ελέγχους που γίνονται με βάση τη θέση των σημείων ως προς τον άξονα των x (βλέπε παραπάνω) Στο συγκεκριμένο παράδειγμα, οι τομές είναι αυτές των ευθειών 14 και 34, με συντεταγμένες αντίστοιχα, (x_0, y_{14t}) και (x_0, y_{43})

Η τελική σχέση έχει ως εξής :

- Βρίσκουμε για κάθε ένα εξεταζόμενο σημείο (O, A ή B) τις διαφορές των συντεταγμένων y των σημείων αυτών με τις συντεταγμένες y των σημείων τομής, δηλαδή, π.χ., για το σημείο O τα $(y_{14t} - y_0)$ και $(y_{43t} - y_0)$.
- Εν συνεχεία, κάνουμε το γινόμενο αυτών των δύο διαφορών, κάθε φορά, και ψάχνουμε να βρούμε αν είναι θετικό ή αρνητικό το αποτέλεσμα του γινομένου.
- Αν το γινόμενο είναι αρνητικό τότε το σημείο που εξετάσαμε είναι μέσα στην περιοχή του πολυγώνου, αν όμως είναι θετικό τότε το σημείο είναι εκτός της περιοχής που εξετάσαμε.

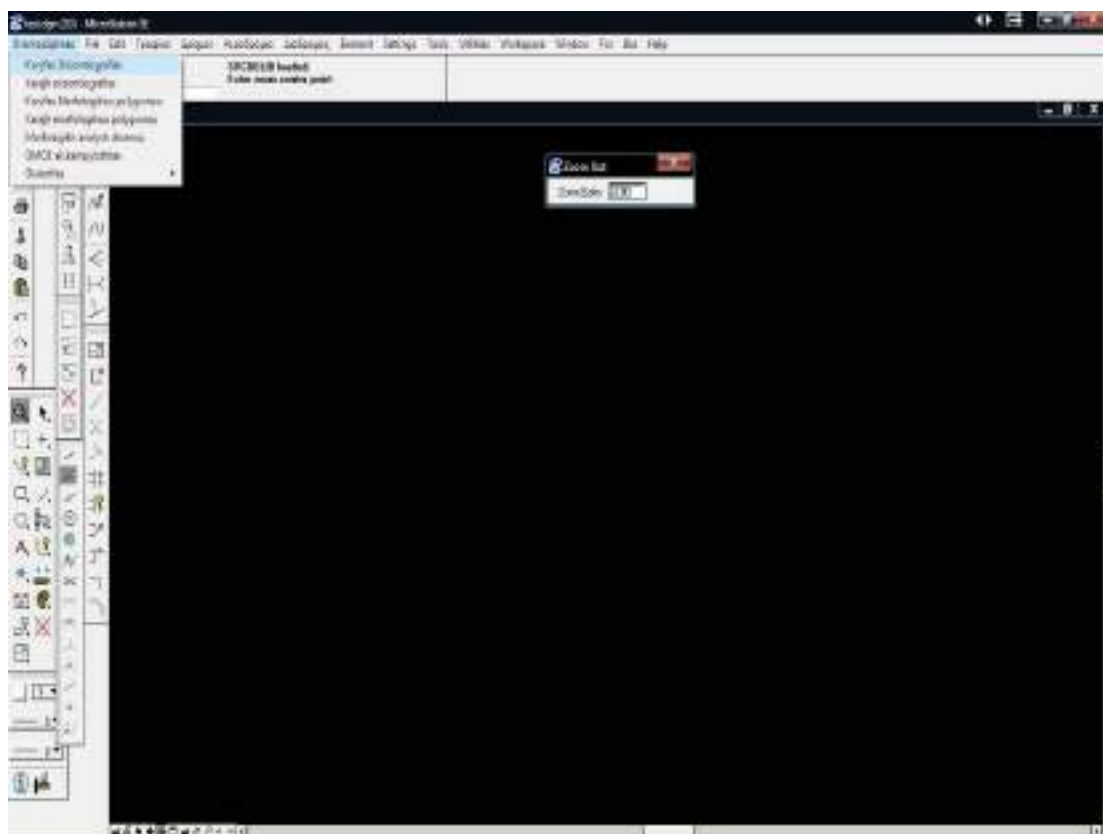
Το γινόμενο αυτών των διαφορών το ονομάσαμε εξίσωση εγκλωβισμού και την χρησιμοποιούμε για να βρίσκουμε ποια σημεία του κάνναβου είναι μέσα σε μια περιοχή, επιφάνεια, ζώνη ή ευθεία και ποια όχι.

Κεφάλαιο 5

ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

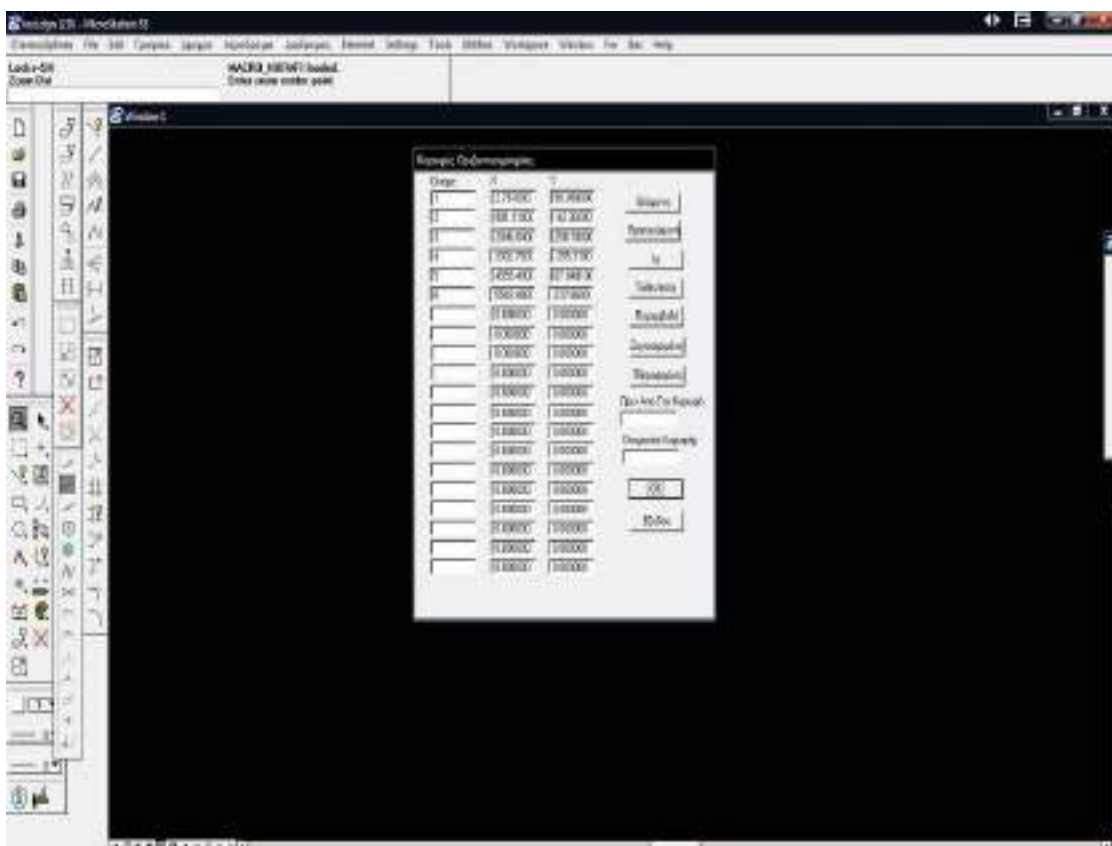
5.1 ΚΑΤΑΣΚΕΥΗ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ

Για να ξεκινήσουμε το λογισμικό, ξεκινάμε επιλέγοντας το button: “Koryfes Orizontiografias” από το tab “DromosSplines”.



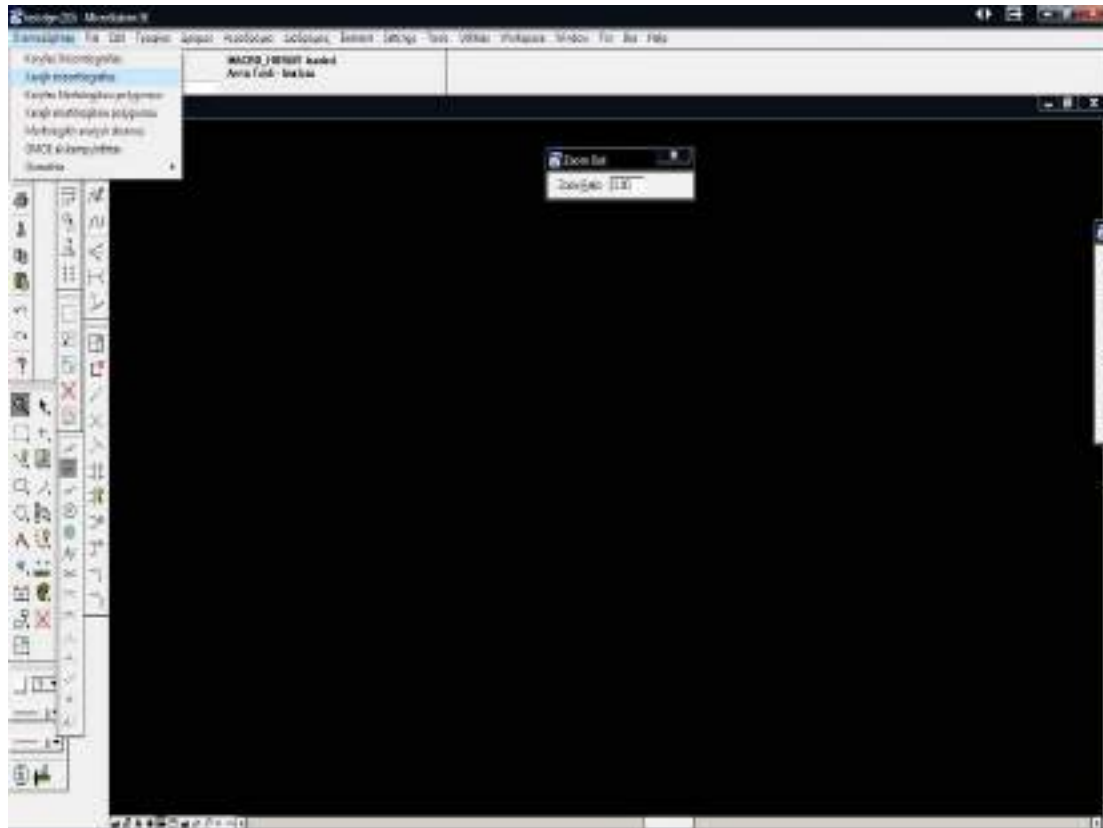
Με αυτή την ενέργεια αυτόματα ο υπολογιστής ανοίγει την macro “horafi.bas” στην οποία εισάγουμε τις κορυφές οριζοντιογραφίας, δηλαδή τα καίρια σημεία από τα οποία πρόκειται να περνάει ο δρόμος.

Συνεπώς ερχόμαστε στο Action Window που φαίνεται παρακάτω.

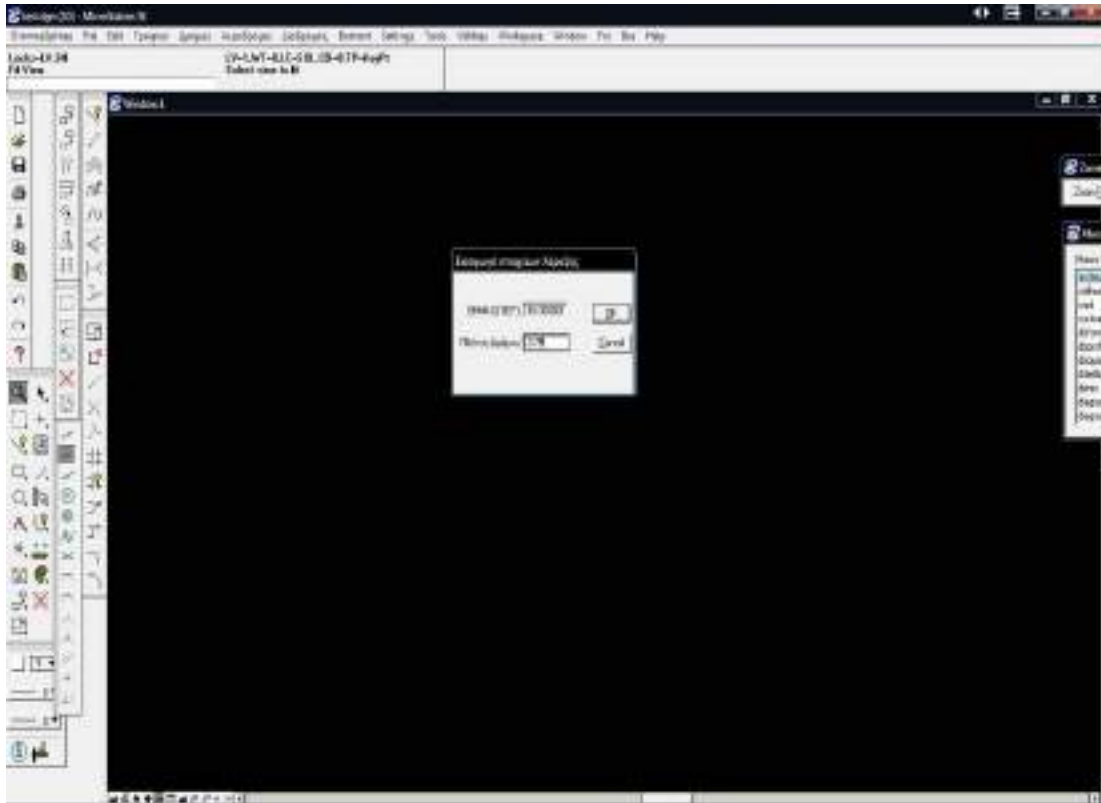


Στο παράθυρο αυτό, εισάγονται οι συντεταγμένες με τον αύξοντα αριθμό τους μπροστά και εν συνεχεία, πατώντας το κουμπί OK αποθηκεύονται στο αρχείο “fm.hor”.

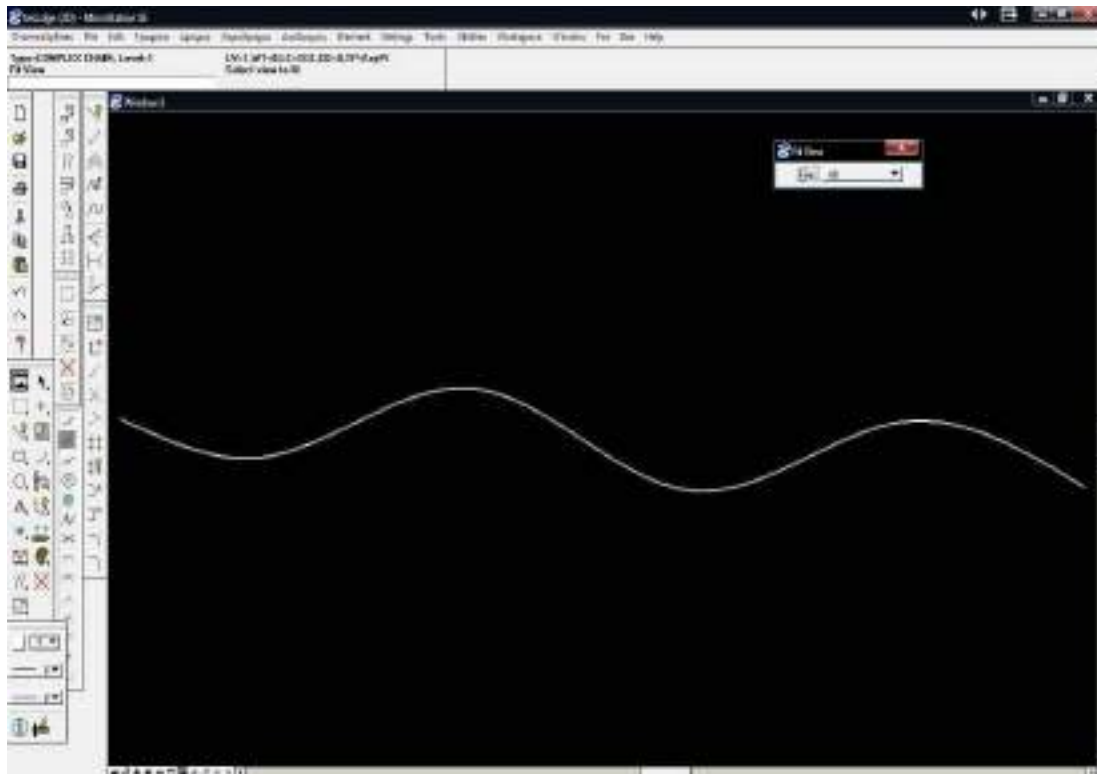
Για να προχωρήσουμε στο σχεδιασμό του δρόμου επιλέγουμε το button «Xarajh Orizontiografias» από το tab “DromosSplines”, το οποίο ανοίγει την macro “aggspl.bas”.



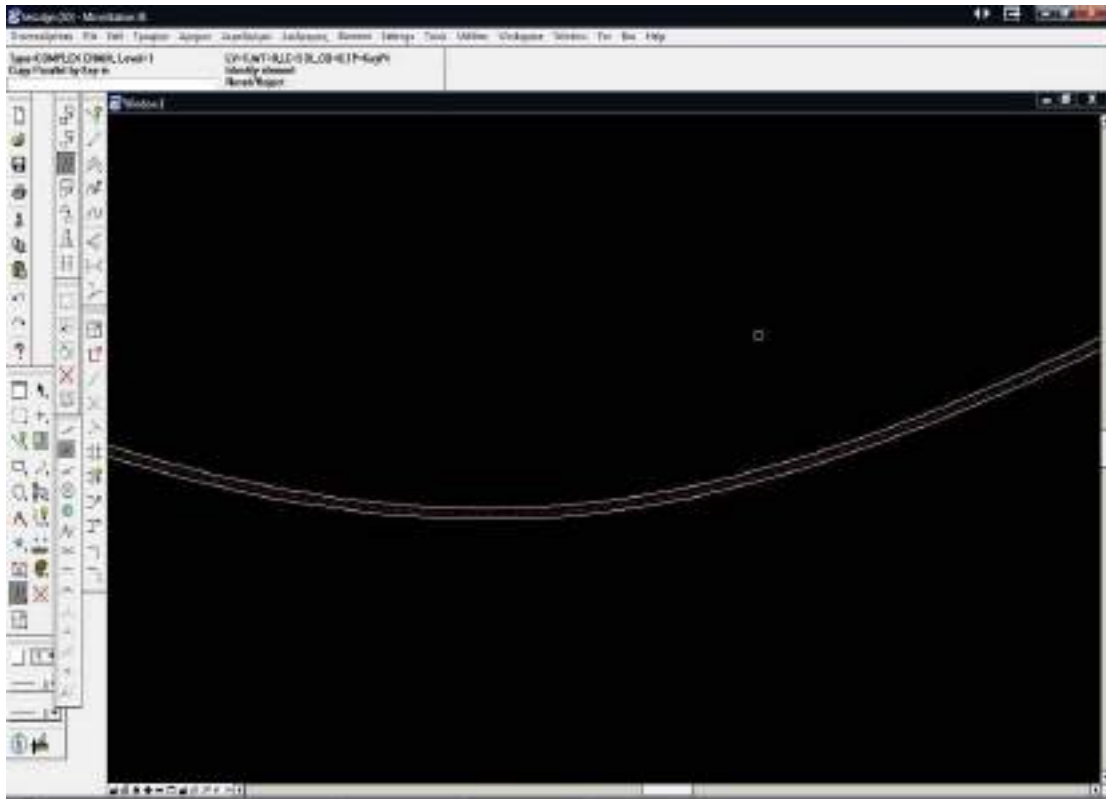
Στο σημείο αυτό το λογισμικό εμφανίζει ένα action window στο οποίο ο χρήστης καλείται να εισάγει το βήμα που επιθυμεί να χρησιμοποιηθεί από τον αλγόριθμο (**βλέπε Splines**) και το πλάτος του δρόμου.



Αφού εισαχθούν οι επιθυμητές τιμές και πατηθεί το πλήκτρο OK, το λογισμικό αυτόματα σχεδιάζει τον δρόμο. Και αποθηκεύει τα σημεία του δρόμου καθώς και τις πρώτες, δεύτερες παραγώγους και την ακτίνα καμπυλότητάς τους στο αρχείο “sp3.dat”.

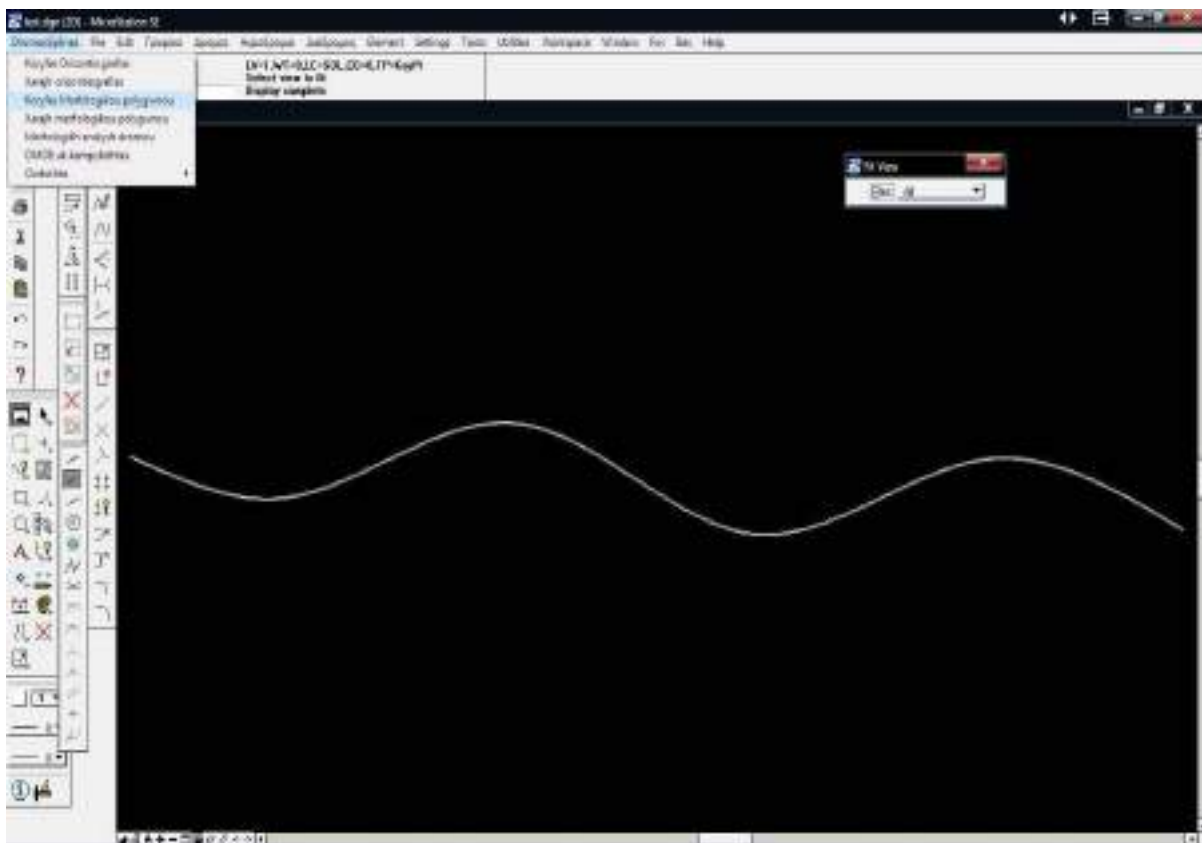


Κάνοντας zoom βλέπουμε μια λεπτομέρεια του σχεδιασμού.

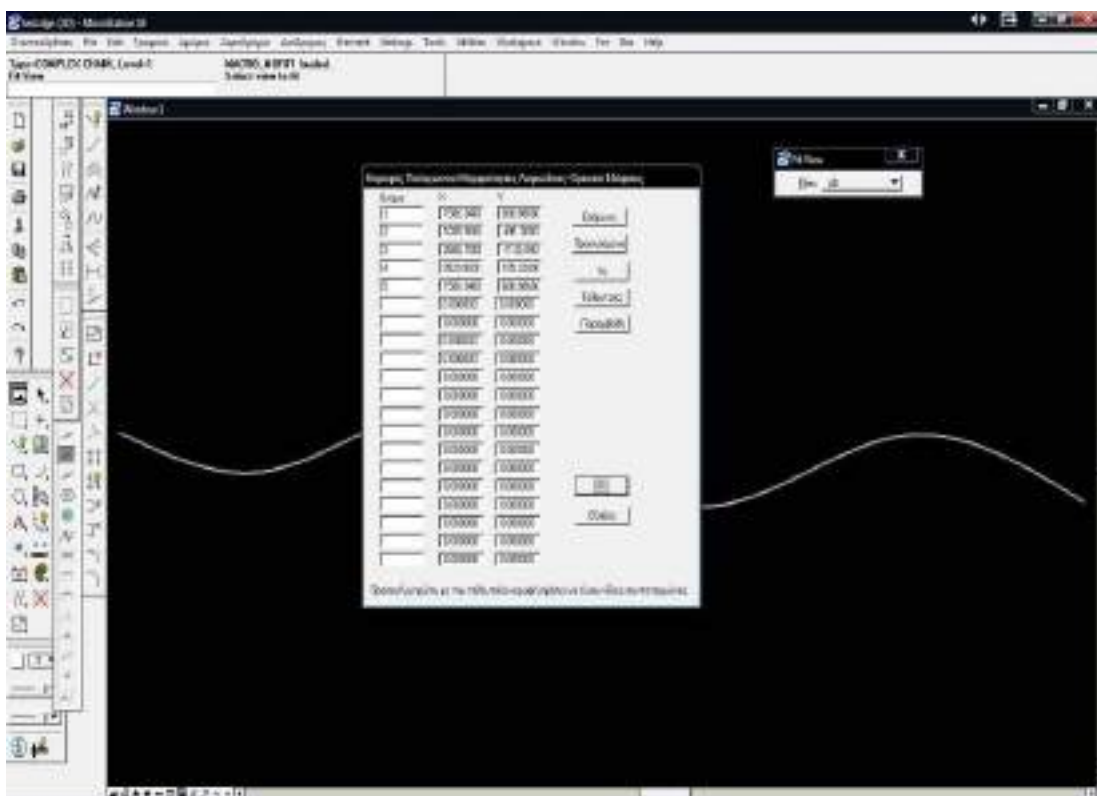


5.2 ΜΟΡΦΟΛΟΓΙΚΗ ΚΑΤΑΤΑΞΗ ΤΟΥ ΔΡΟΜΟΥ

Στην επόμενη φάση ο χρήστης καλείται να προσδιορίσει (εγκλωβίζοντας τες σε πολύγωνο) τις λοφώδεις περιοχές που βρίσκονται στην περιοχή. Η διαδικασία αυτή ξεκινάει πατώντας το button “Koryfes Morfolοgikou polygwnou” στο tab “DromosSplines” το οποίο ανοίγει την macro “morfl.bas”.

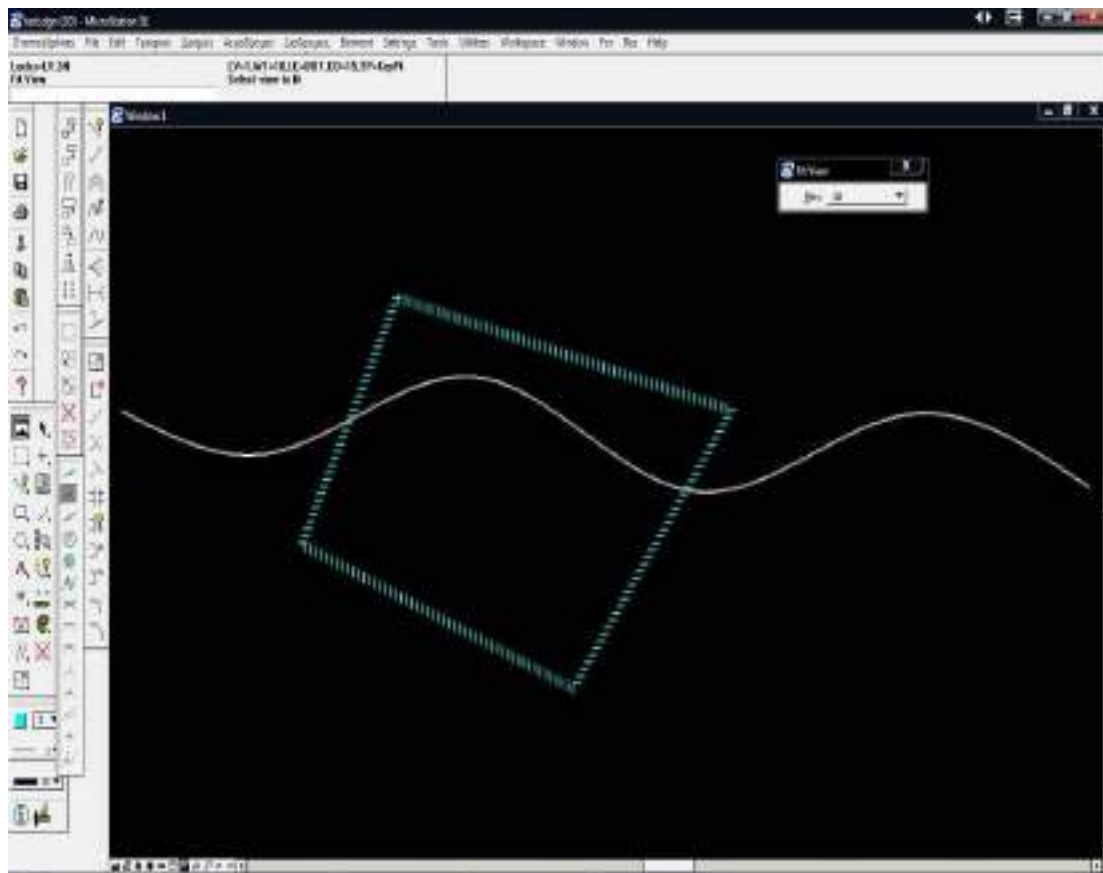


Έτσι εμφανίζεται στην οθόνη το παράθυρο στο οποίο εισάγονται οι συντεταγμένες των κορυφών του μορφολογικού πολυγώνου

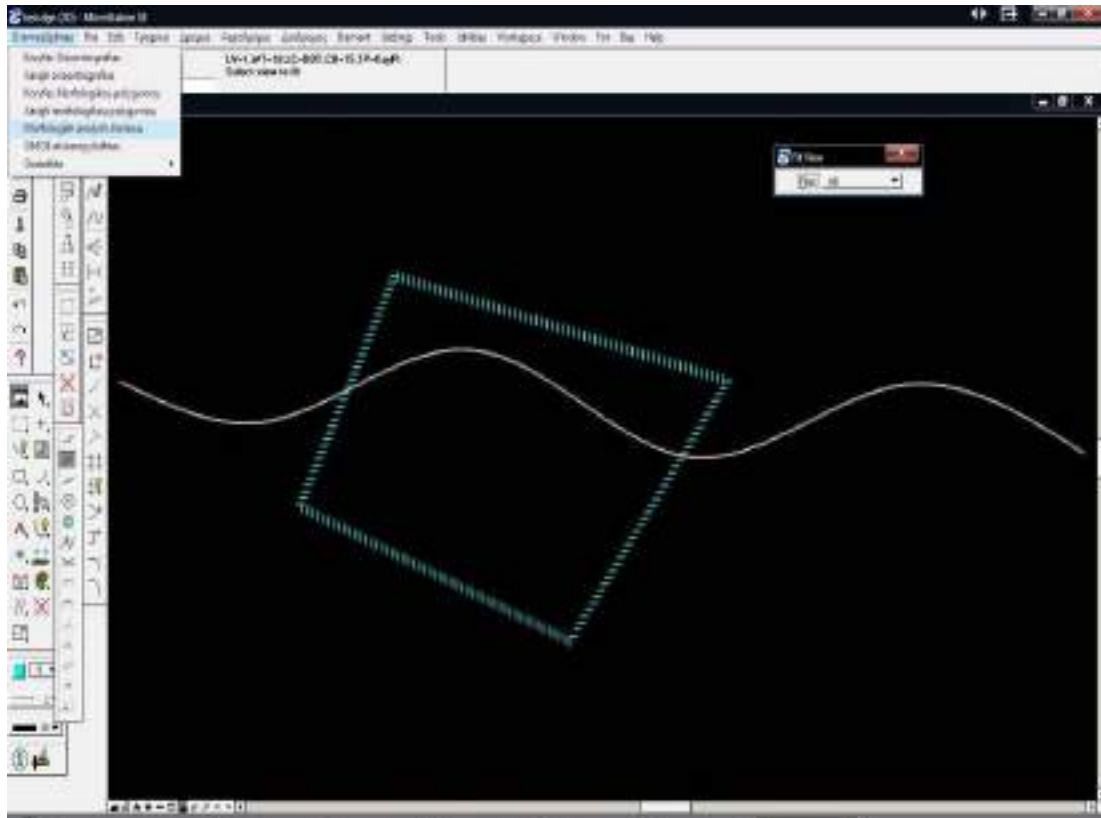


Στο σημείο αυτό πρέπει να σημειωθεί ότι η πρώτη και τελευταία κορυφή του πολυγώνου πρέπει να ταυτίζονται για να κλείσει το πολύγωνο.

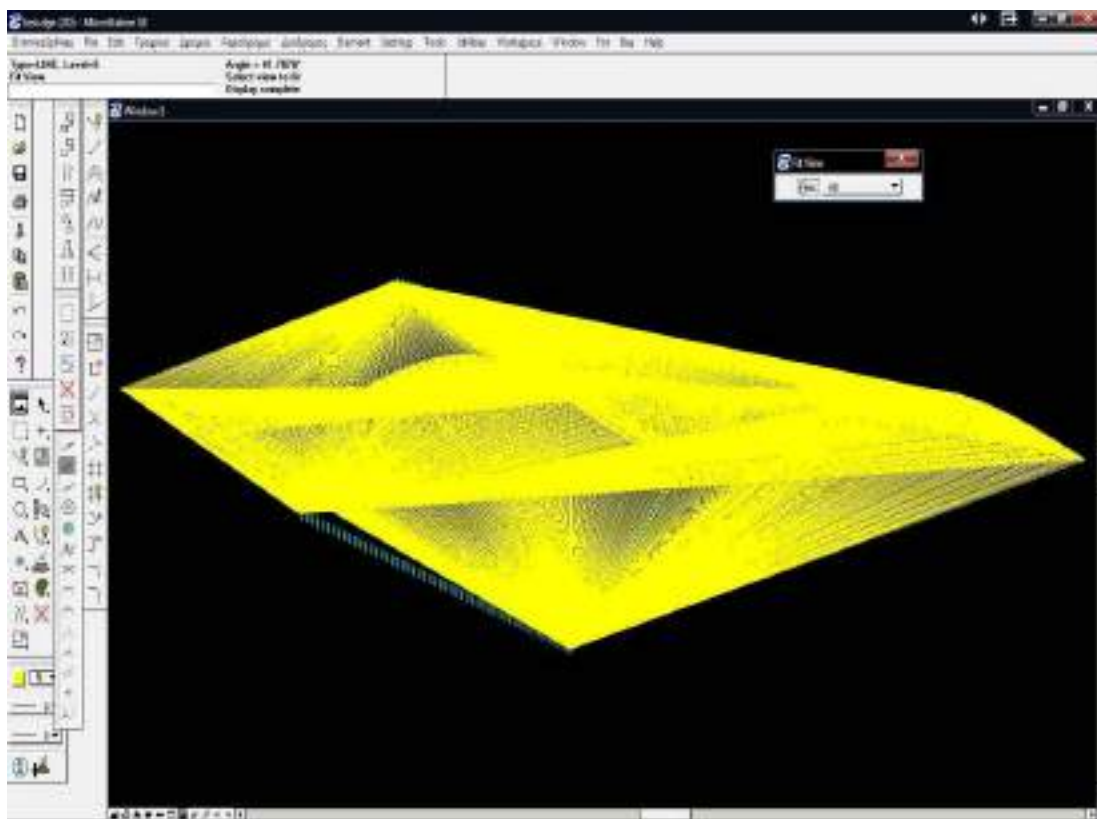
Έτσι αυτόματα ανοίγει η macro “morfgraf.dat” και χαράσσεται το μορφολογικό πολύγωνο που εγκλωβίζει τις λοφώδεις – ορεινές περιοχές του χάρτη.



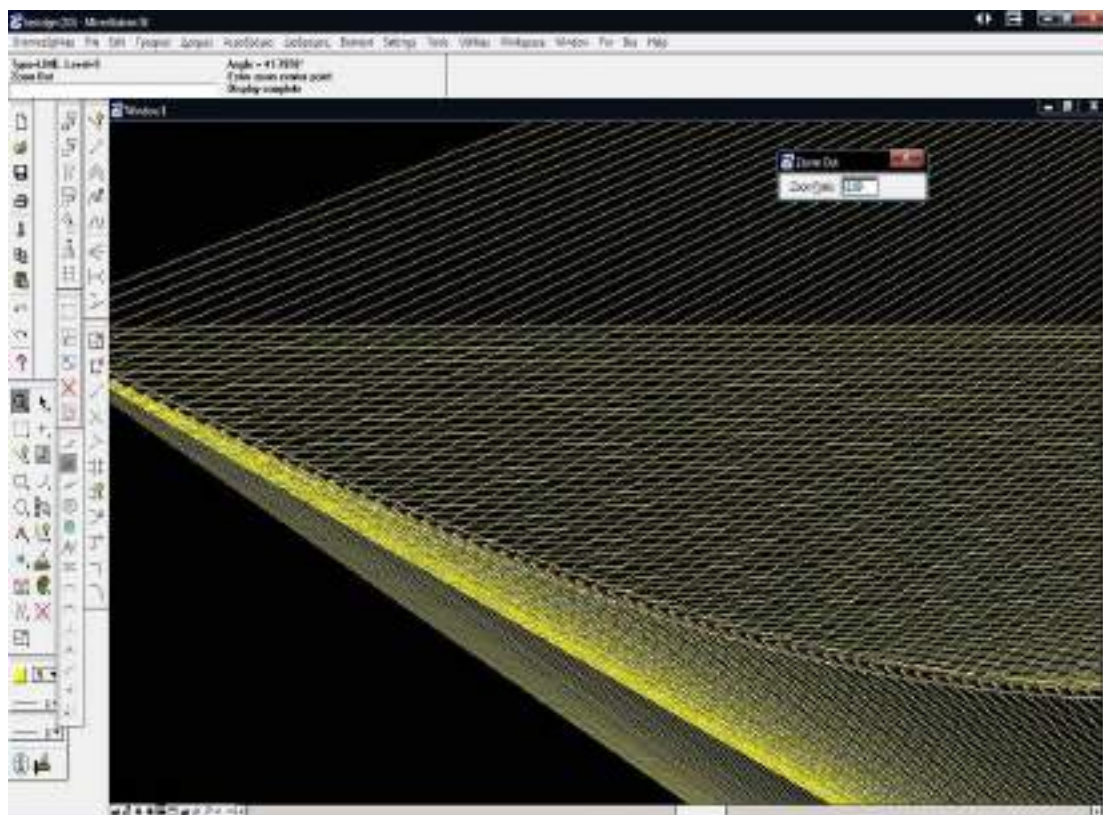
Στη συνέχεια ο χρήστης πρέπει να πατήσει το button “Morfologikh analysh Dromou” του tab “DromosSplines” προκειμένου το λογισμικό να εντοπίσει ποια από τα σημεία του δρόμου βρίσκονται εντός της περιοχής του πολυγώνου.



Έτσι ανοίγεται η macro “tomi.bas” η οποία για κάθε σημείο του δρόμου ενώνει το εκάστοτε σημείο με τις κορυφές του πολυγώνου και μετρώντας τις γωνίες που σχηματίζονται αντιλαμβάνεται εάν το σημείο βρίσκεται εντός ή εκτός. **(βλέπε κεφάλαιο εξισώσεις)**



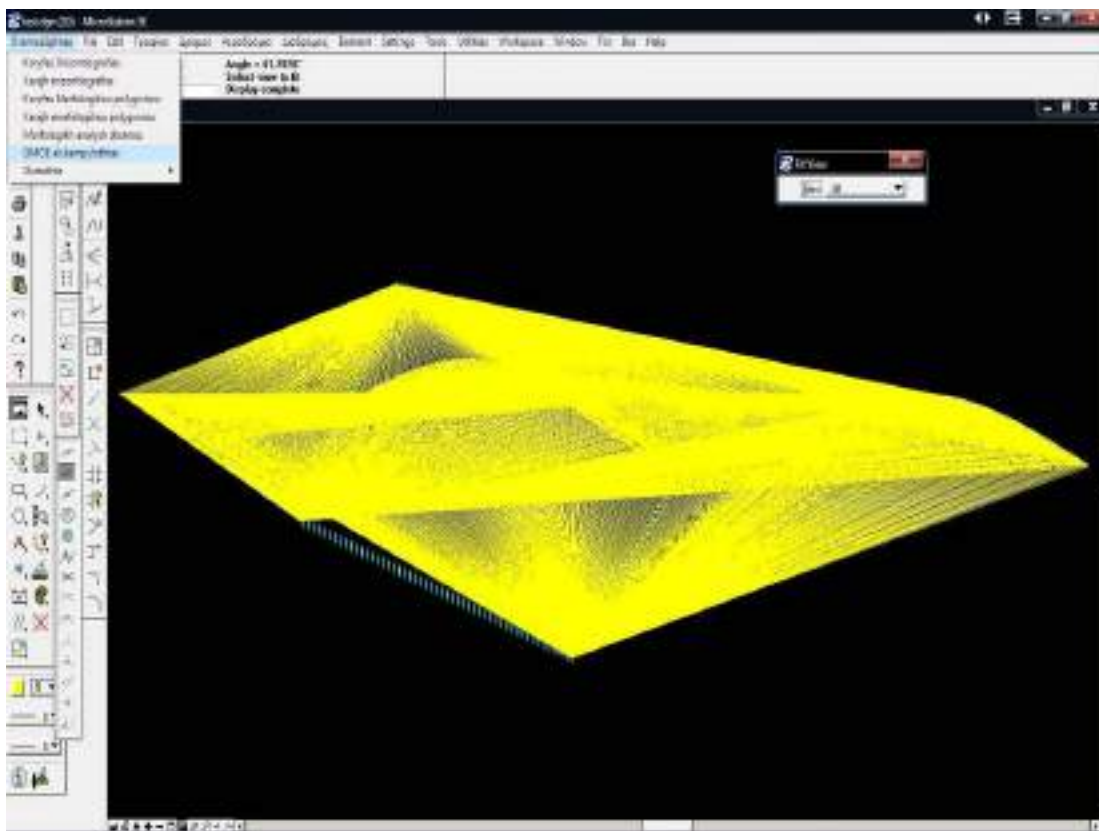
Με zoom φαίνεται μια λεπτομέρεια του σχεδίου προκειμένου να γίνει πιο κατανοητή η εφαρμογή.



Αφού ολοκληρωθεί η διαδικασία το λογισμικό δημιουργεί το αρχείο “tmorf.dat” στο οποίο αποθηκεύεται για κάθε σημείο του δρόμου μια τιμή $tr(i)$ που υποδηλώνει αν είναι 2 το ότι το σημείο βρίσκεται εντός του πολυγώνου και αν είναι 1, εκτός.

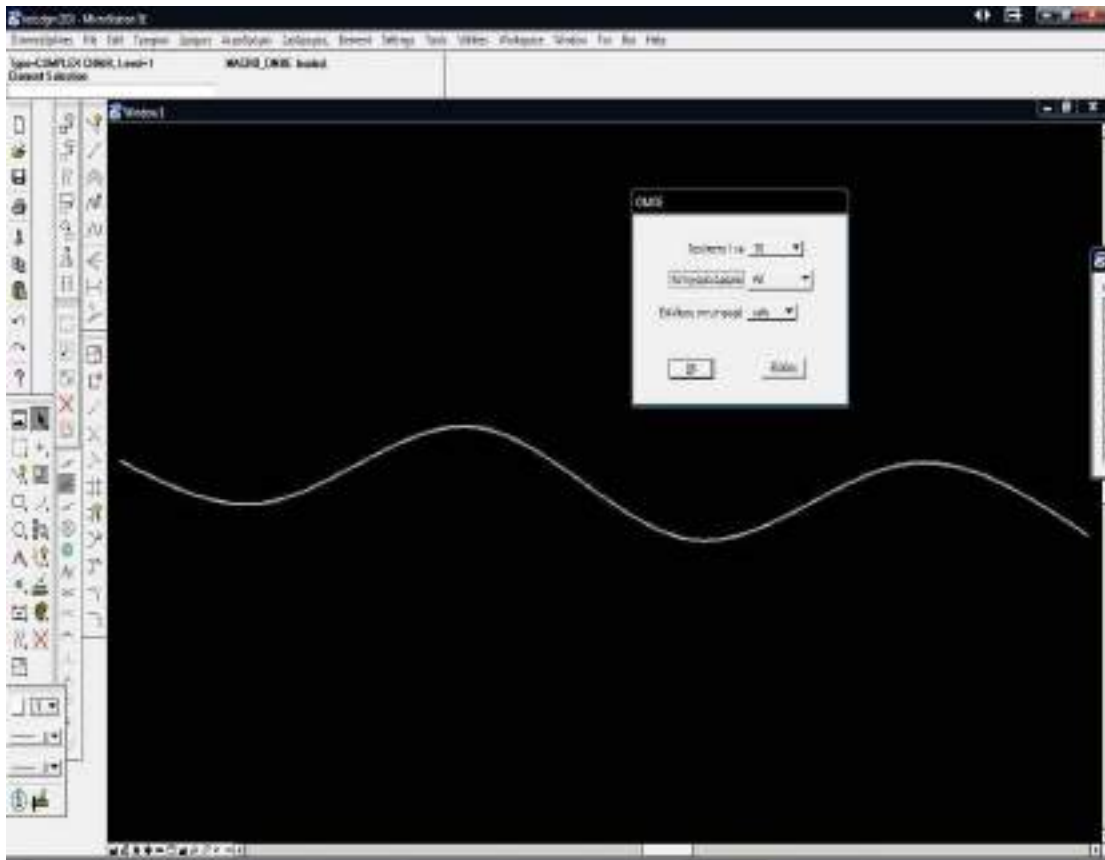
5.3 ΕΛΕΓΧΟΣ ΑΚΤΙΝΑΣ ΚΑΜΠΥΛΟΤΗΤΑΣ ΚΑΤΑ ΟΜΟΕ

Στη συνέχεια ο χρήστης είναι έτοιμος να εκτελέσει τον έλεγχο ακτίνας καμπυλότητας κατά ΟΜΟΕ. Αυτό γίνεται πατώντας το button “ΟΜΟΕ ακ. Καρμυλοθτας” στο tab “DromosSplines”.



Έτσι ενεργοποιείται η macro “omoe.bas” η οποία φορτώνει στη μνήμη όλα τα μέχρι τότε εξαγόμενα dat αρχεία από τα προηγούμενα υπο προγράμματα. Αρχίζοντας ο χρήστης καλείτε μέσω του παραθύρου που εμφανίζεται να επιλέξει την επιθυμητή ταχύτητα V_e , κατηγορία δρόμου (AI,AII,BI,BII κλπ) και αν θέλει να συνυπολογίσει την επίκληση στον έλεγχο του.

(Για καλύτερη ευκρίνεια στις εικόνες από εδώ και στο εξής σβήνονται οι γραμμές που προέκυψαν από την μορφολογική κατάταξη του δρόμου)

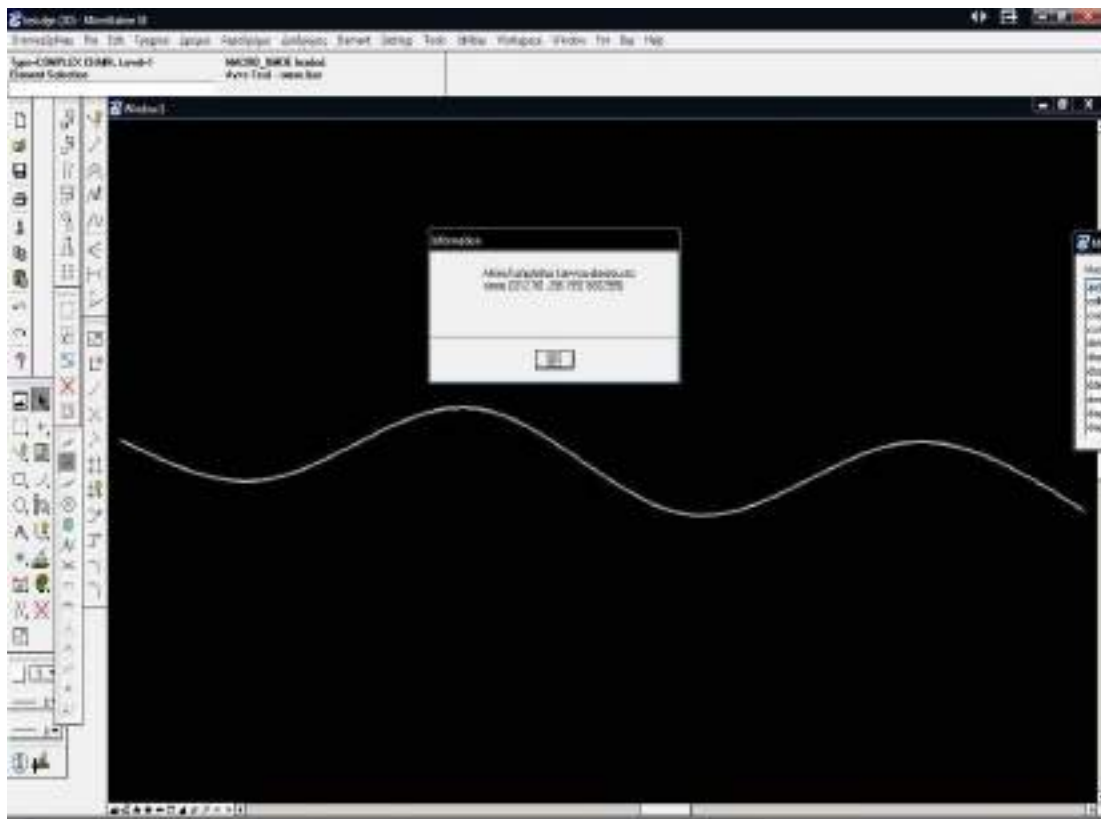


Σε περίπτωση που ο χρήστης έχει επιλέξει συνδυασμό ταχύτητας και κατηγορίας δρόμου που δεν υπάρχει, παρουσιάζεται το παρακάτω μήνυμα.

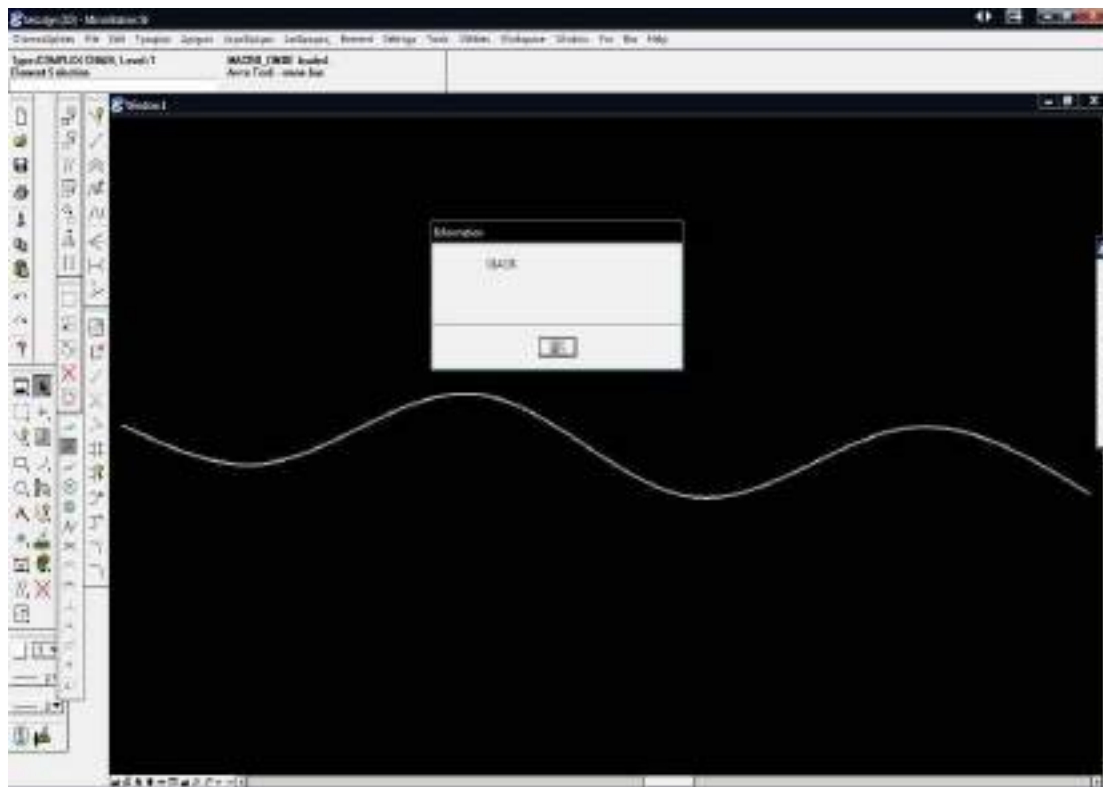


Αν είναι όλα εντάξει το λογισμικό προχωράει κανονικά στον έλεγχο.

Στην περίπτωση που υπάρχει πρόβλημα και η ακτίνα καμπυλότητας είναι κάτω του επιτρεπτού σε κάποιο σημείο, παρουσιάζεται το εξής μήνυμα.



Αν όλα τα σημεία έχουν την απαιτούμενη ακτίνα καμπυλότητας, τότε το λογισμικό τερματίζει κανονικά εμφανίζοντας το παρακάτω μήνυμα.



5.4 ΕΥΡΕΣΗ ΤΗΣ ΟΡΑΤΟΤΗΤΑΣ ΓΙΑ ΚΑΘΕ ΣΗΜΕΙΟ ΤΟΥ ΔΡΟΜΟΥ ΚΑΙ ΚΑΤΟΠΙΝ ΕΛΕΓΧΟΣ

Αρχικά ο χρήστης πρέπει να τρέξει τέσσερα υποπρογράμματα τα οποία βρίσκονται μέσα στο “DromosSplines”/”Oratotita”, και είναι τα :

1.Eyresh Oratofhtas Stashes A-D

Ενεργοποιείται η macro “oratStashA-D.bas” που βρίσκει για κάθε σημείο του δρόμου την ορατότητά του για στάση σε πορεία από αριστερά προς δεξιά και εν συνεχεία δημιουργεί το αρχείο “oratotitaSTA-D.dat” στο οποίο και αποθηκεύει τις τιμές που υπολόγισε.

2.Eyresh Oratotitas Stashes D-A

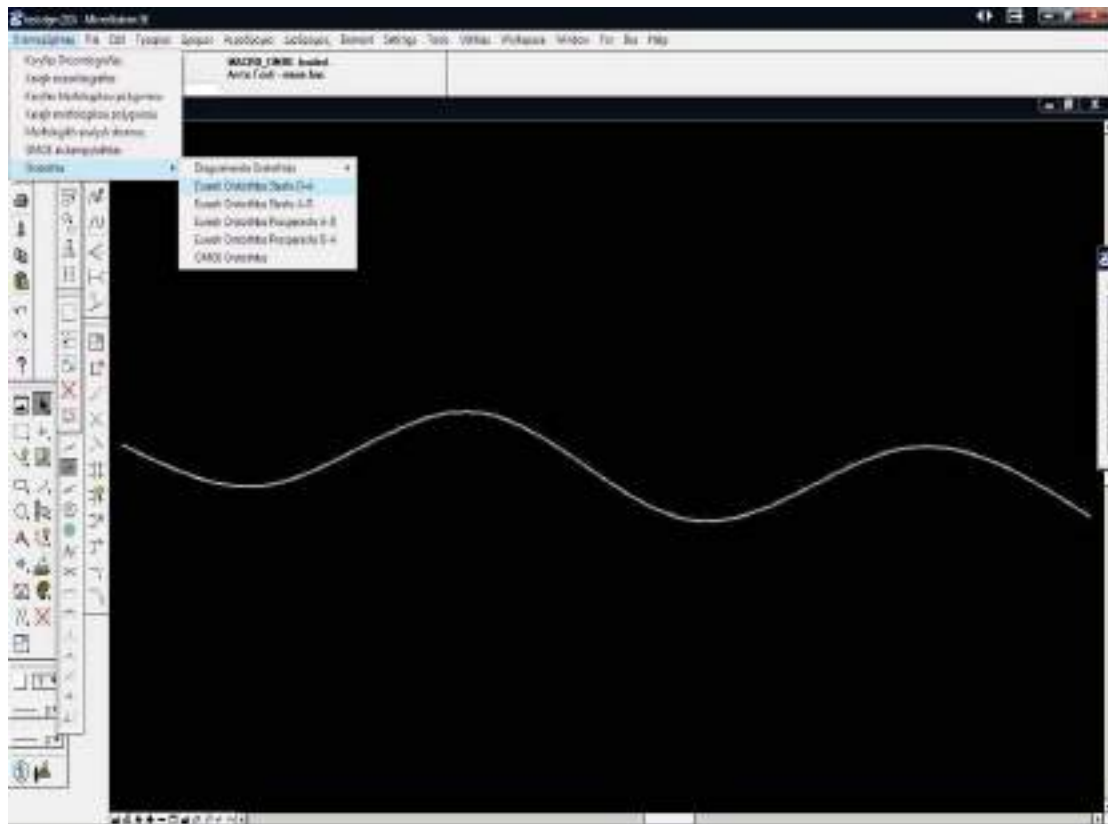
Ενεργοποιείται η macro “oratStashD-A.bas” που βρίσκει για κάθε σημείο του δρόμου την ορατότητά του για στάση σε πορεία από δεξιά προς αριστερά και εν συνεχεία δημιουργεί το αρχείο “oratotitaSTD-A.dat” στο οποίο και αποθηκεύει τις τιμές που υπολόγισε.

3.Eyresh Oratofhtas Prosperashs A-D

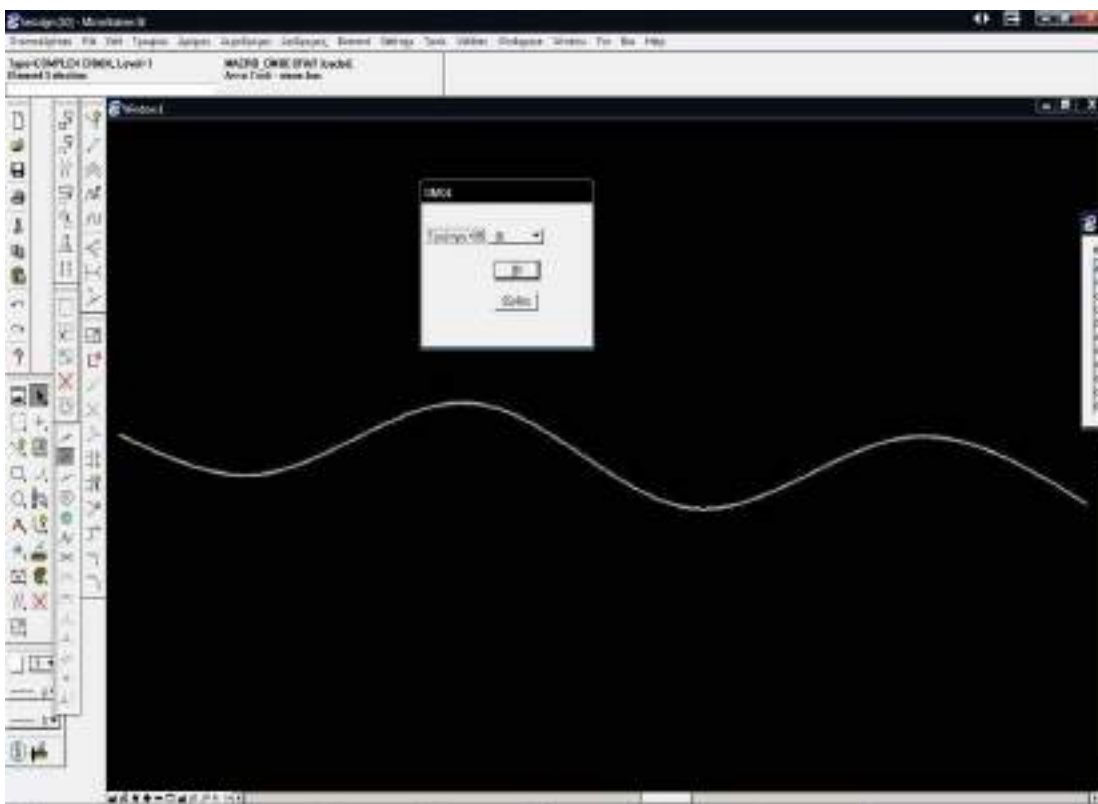
Ενεργοποιείται η macro “oratProsA-D.bas” που βρίσκει για κάθε σημείο του δρόμου την ορατότητά του για στάση σε πορεία από αριστερά προς δεξιά και εν συνεχεία δημιουργεί το αρχείο “oratotitaPrA-D.dat” στο οποίο και αποθηκεύει τις τιμές που υπολόγισε.

4.Eyresh Oratofhtas Prosperashs D-A

Ενεργοποιείται η macro “oratProsD-A.bas” που βρίσκει για κάθε σημείο του δρόμου την ορατότητά του για στάση σε πορεία από δεξιά προς αριστερά και εν συνεχεία δημιουργεί το αρχείο “oratotitaPrD-A.dat” στο οποίο και αποθηκεύει τις τιμές που υπολόγισε.



Έτσι λοιπόν παρουσιάζεται το παράθυρο στο οποίο ο χρήστης επιλέγει την V_{85} που θέλει να χρησιμοποιήσει στον έλεγχό του.



Πατώντας OK το λογισμικό δημιουργεί το αρχείο “apaitorat.dat” στο οποίο αποθηκεύει τις ελάχιστες τιμές ορατότητας για στάση και προσπέραση που θα πρέπει να πληροί ο δρόμος.

5.5 ΧΑΡΑΞΗ ΔΙΑΓΡΑΜΜΑΤΩΝ ΟΡΑΤΟΤΗΤΑΣ

Αφού έχουν ολοκληρωθεί όλα τα παραπάνω, σειρά έχει η χάραξη των διαγραμμάτων ορατότητας. Ο χρήστης πρέπει να πατήσει τα buttons :

1.Prosperash A-D

Σχεδιάζει το διάγραμμα ορατότητας για προσπέραση σε πορεία από αριστερά προς δεξιά.

2.Prosperash D-A

Σχεδιάζει το διάγραμμα ορατότητας για προσπέραση σε πορεία από δεξιά προς αριστερά

3.Stash A-D

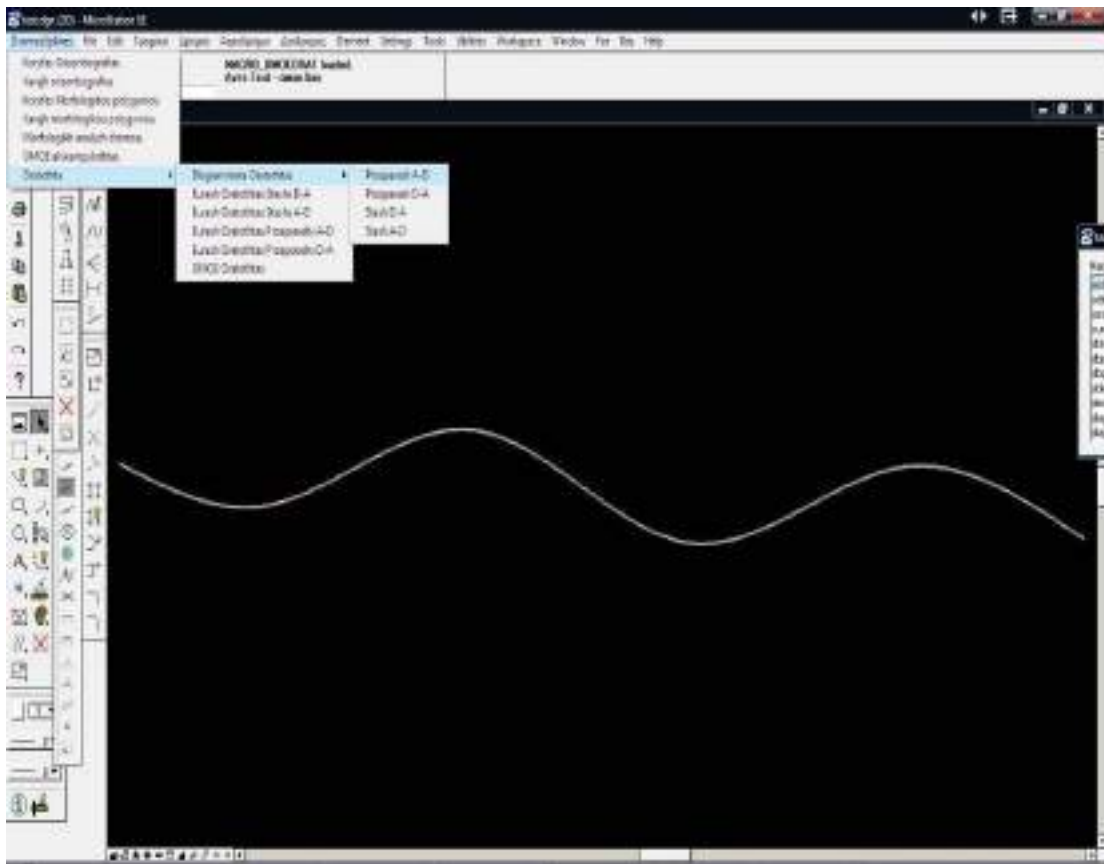
Σχεδιάζει το διάγραμμα ορατότητας για στάση σε πορεία από αριστερά προς δεξιά.

4.Stash D-A

Σχεδιάζει το διάγραμμα ορατότητας για στάση σε πορεία από δεξιά προς αριστερά.

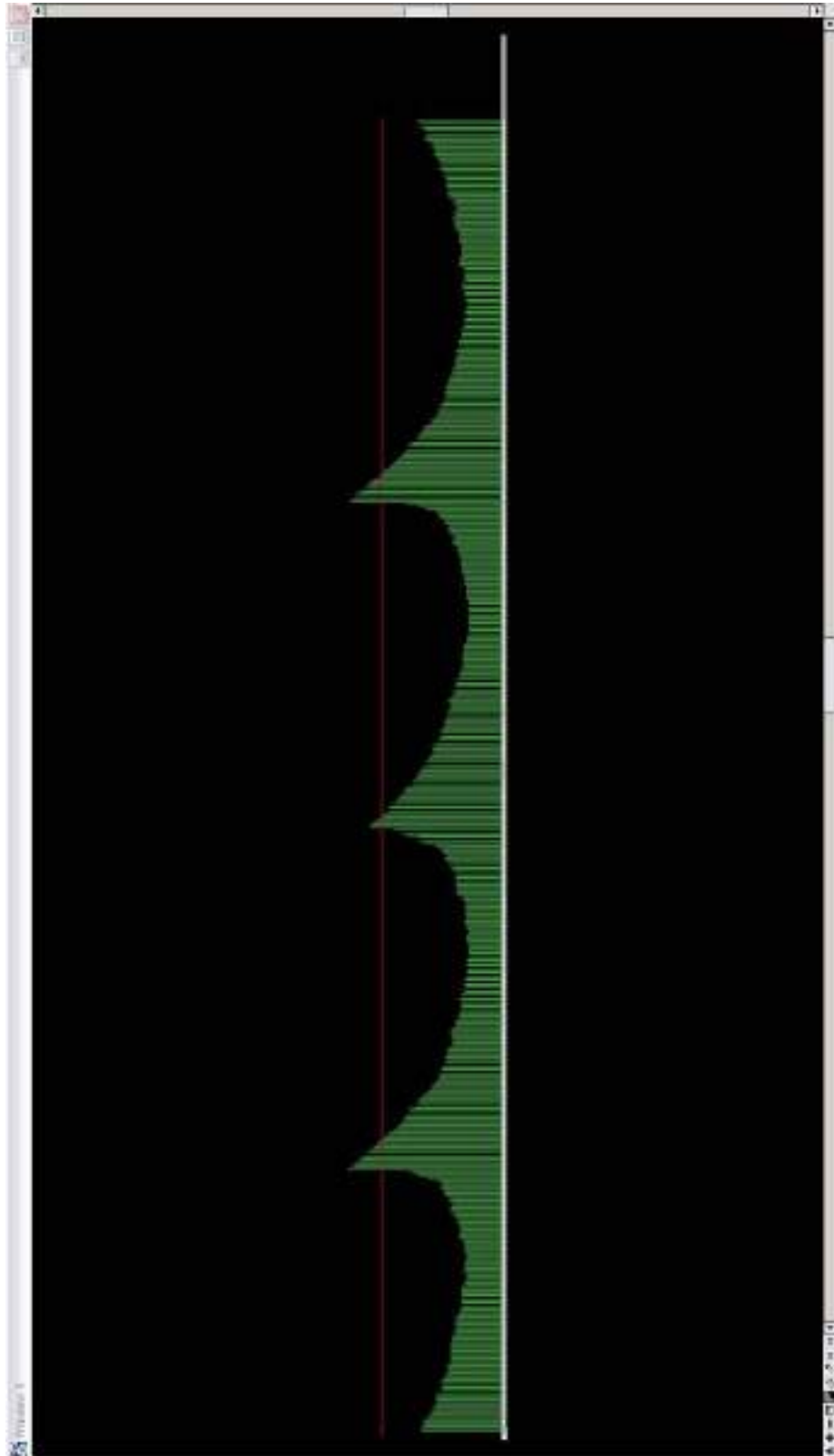
Όλα τα παραπάνω buttons βρίσκονται στο tab

“DromosSplines”/”Oratotita”/”Diagrammata Oratothtas”

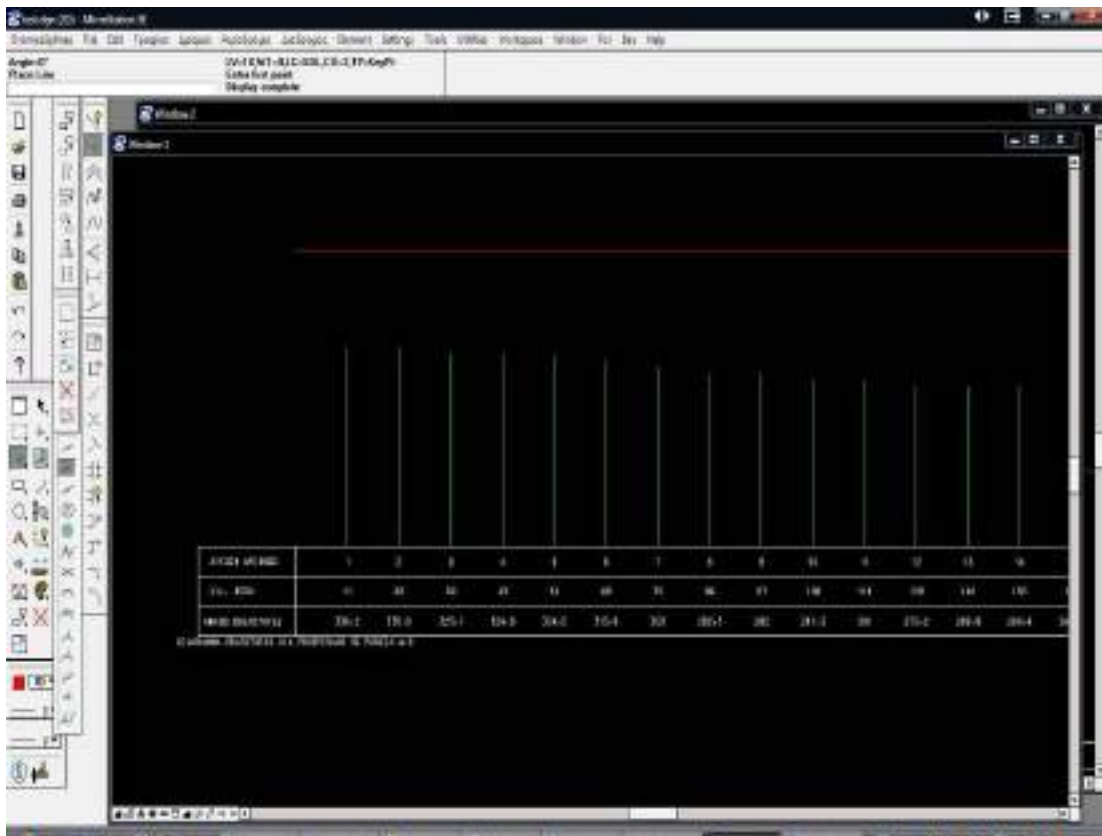


Έτσι λοιπόν θα προκύψουν τα παρακάτω διαγράμματα :

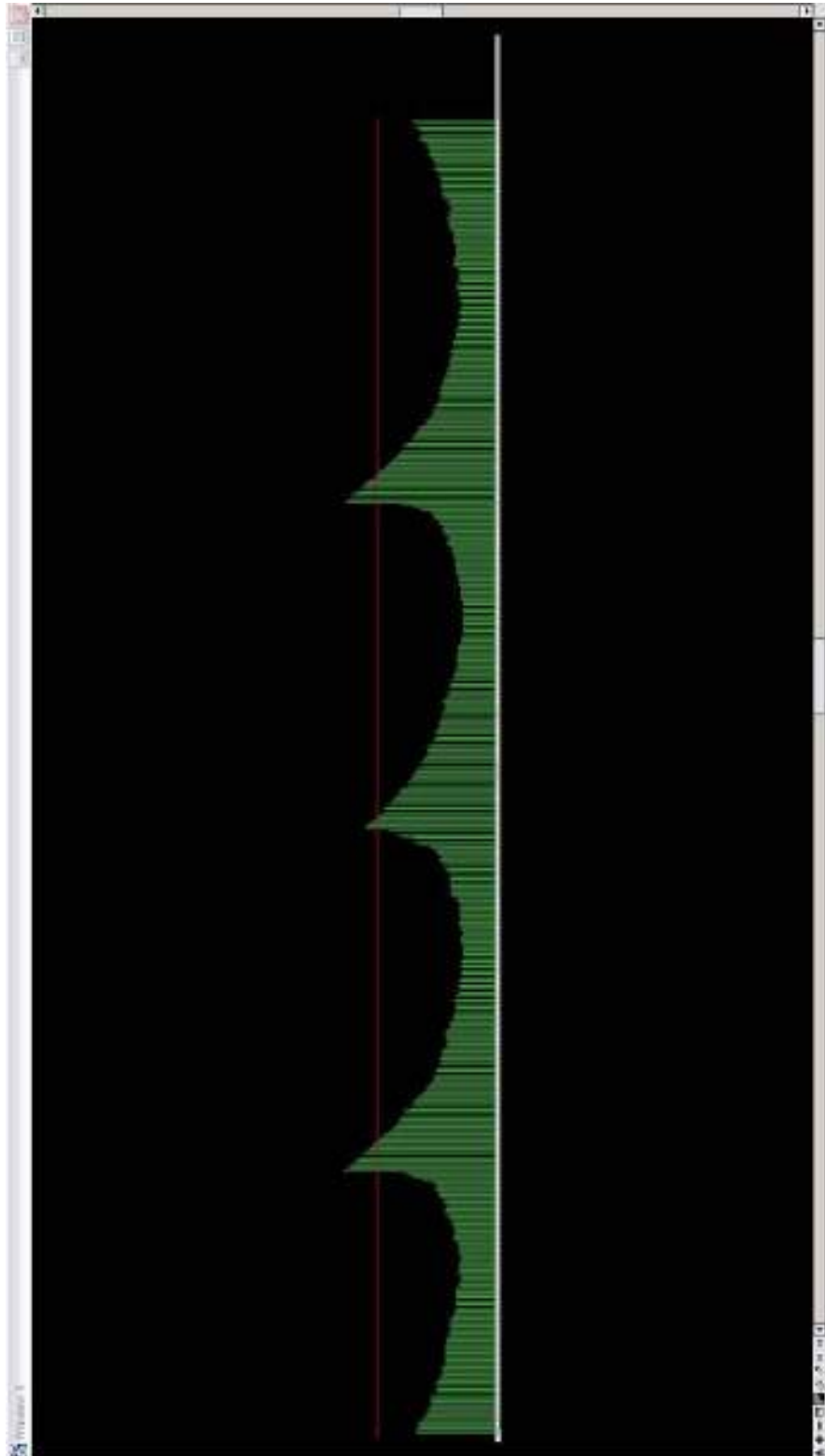
Σχήμα 5.1 Διάγραμμα Ορατότητας Για Προσπέραση Από Αριστερά Προς Δεξιά



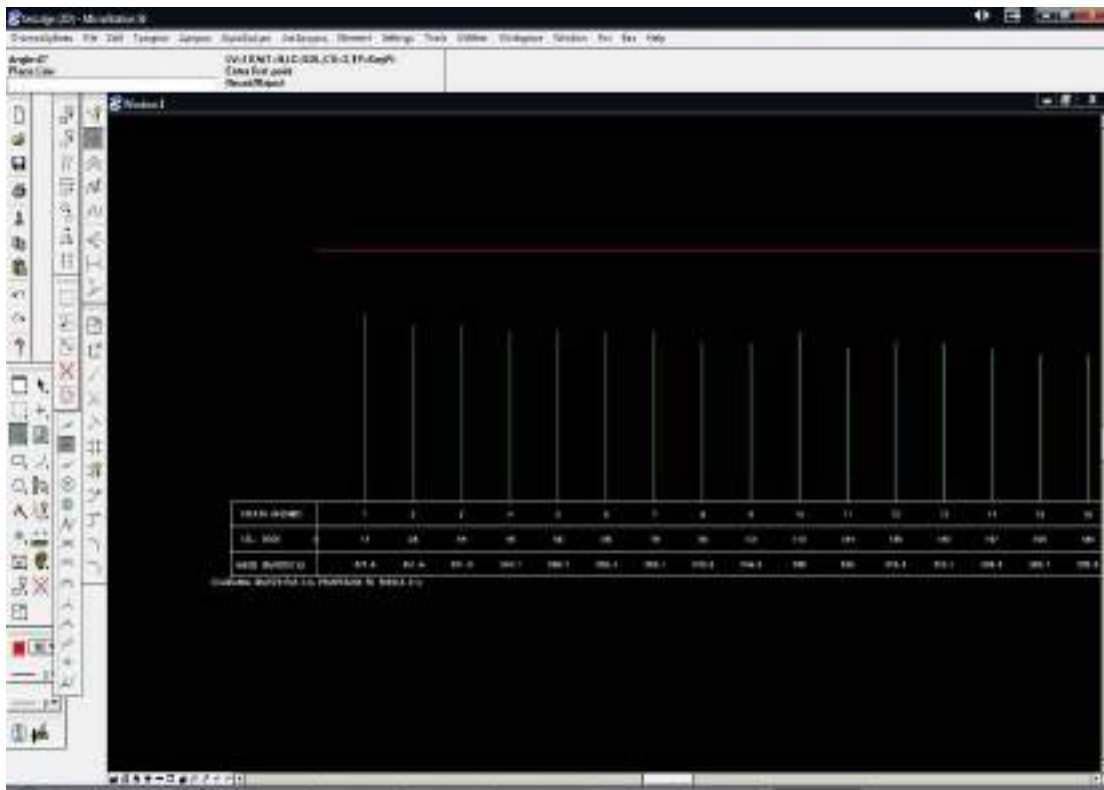
Λεπτομέρεια :



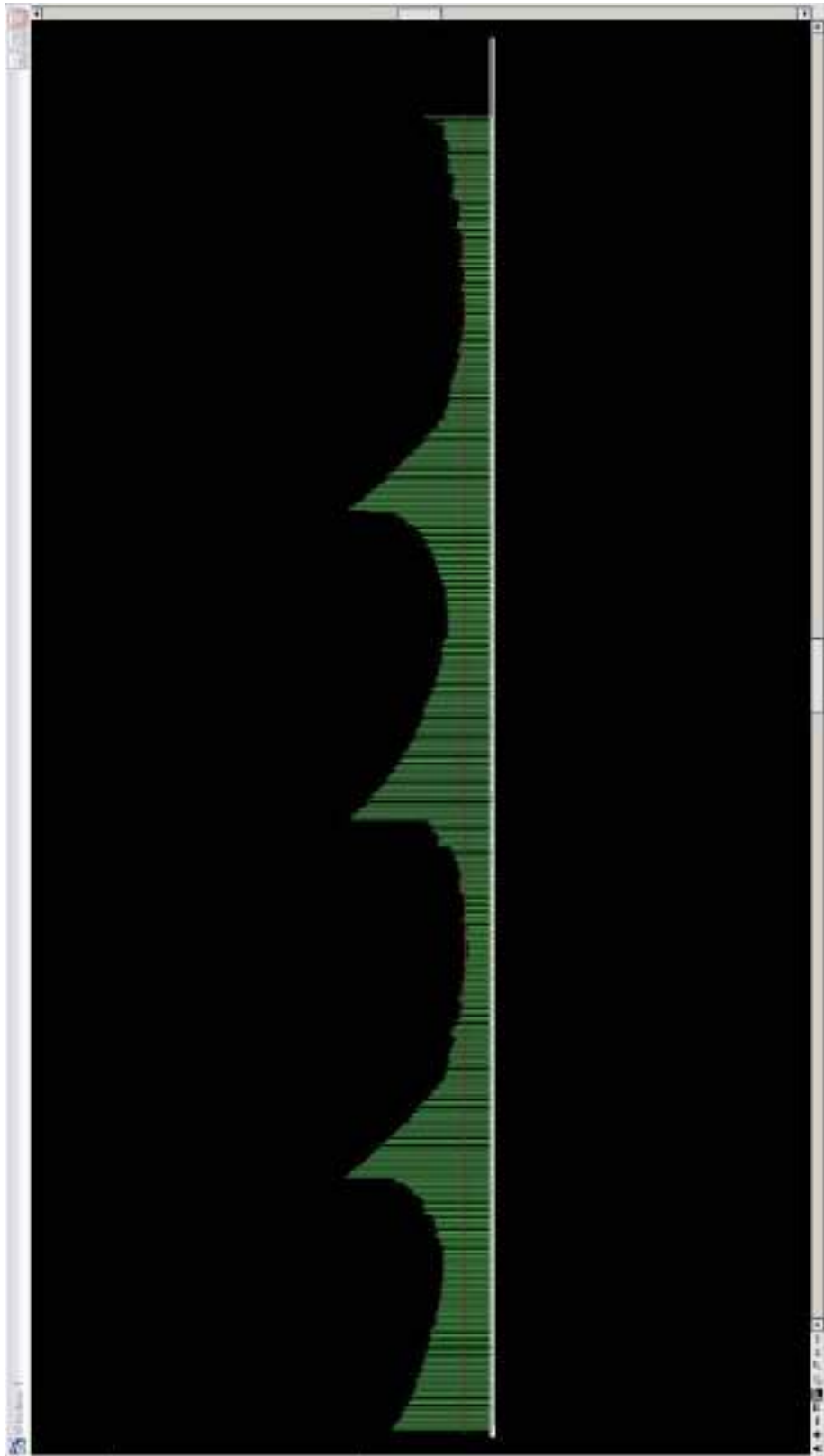
Σχήμα 5.2 Διάγραμμα Ορατότητας Για Προσπέραση Από Δεξιά Προς Αριστερά



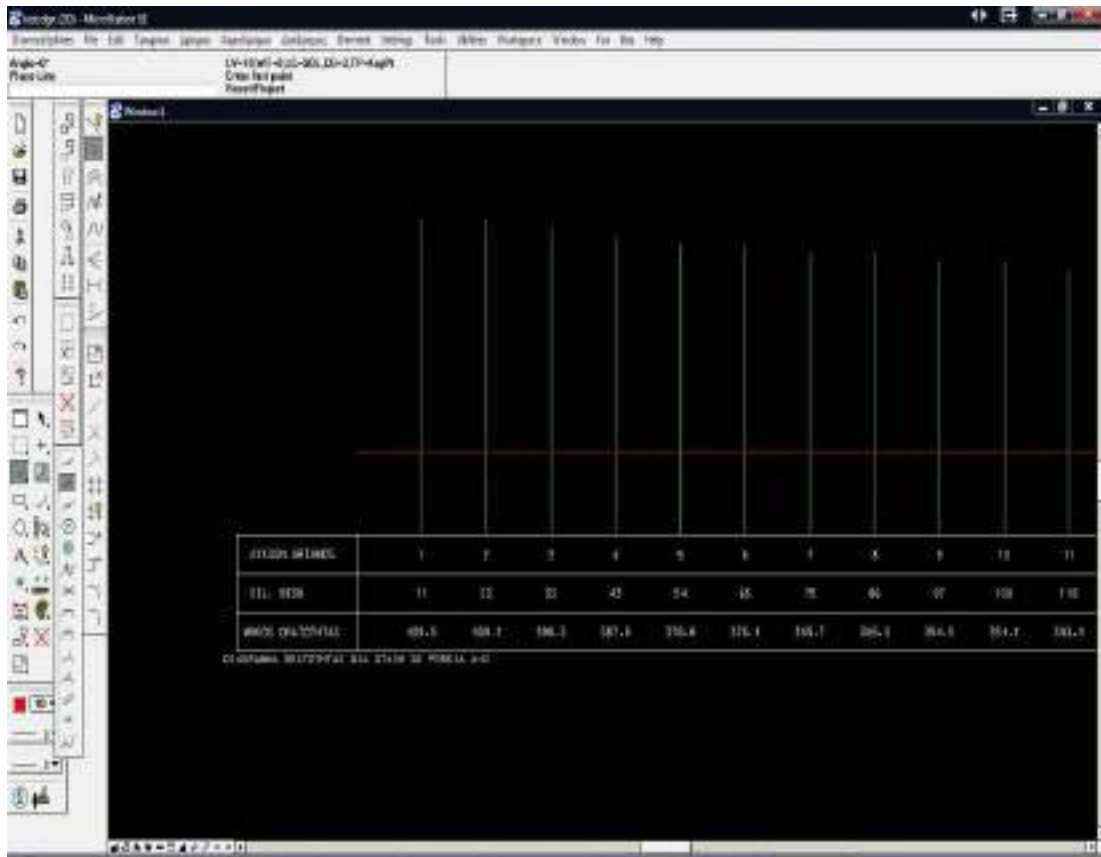
Λεπτομέρεια:



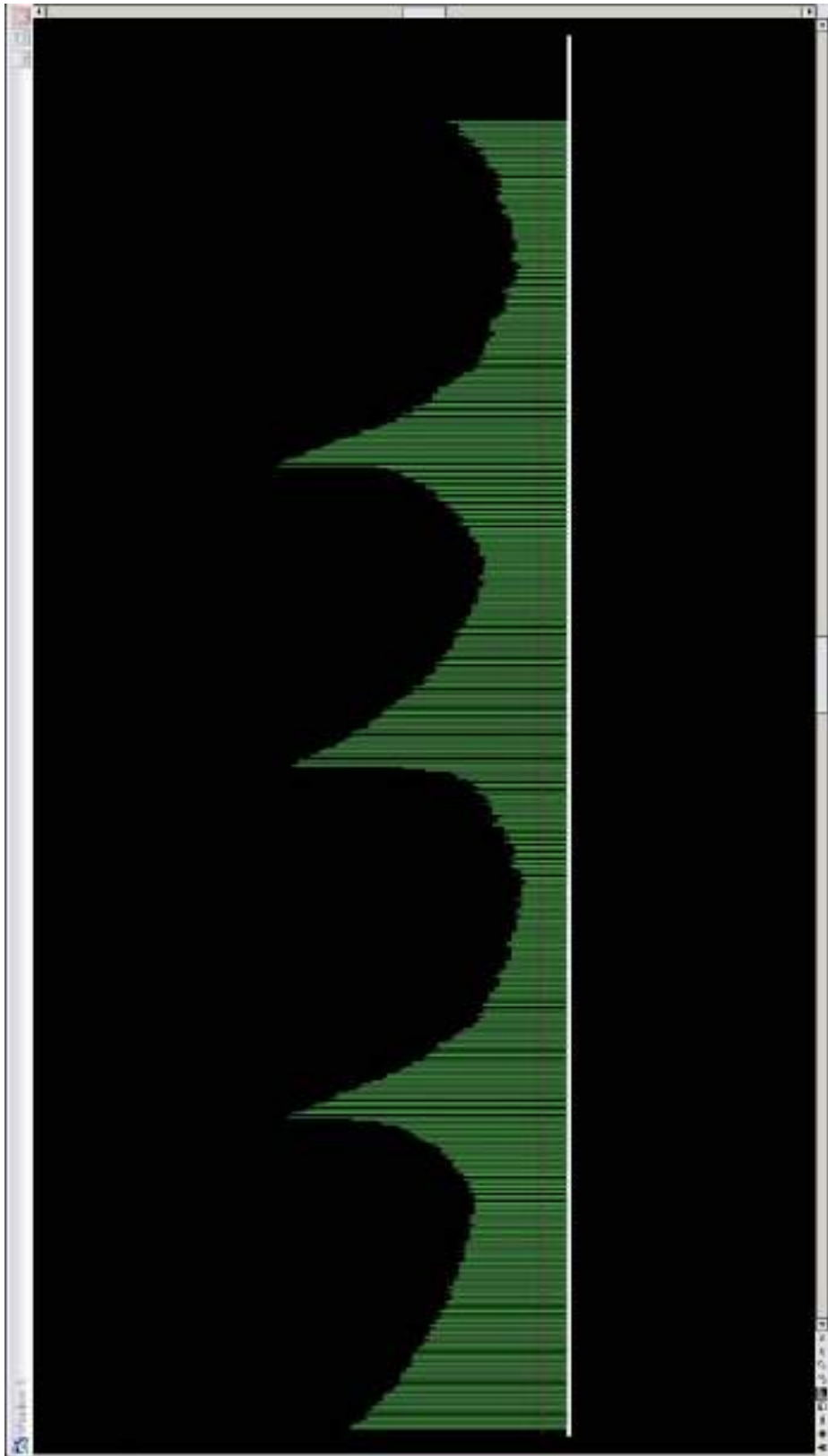
Σχήμα 5.1 Διάγραμμα Ορατότητας Για Στάση Από Αριστερά Προς Δεξιά



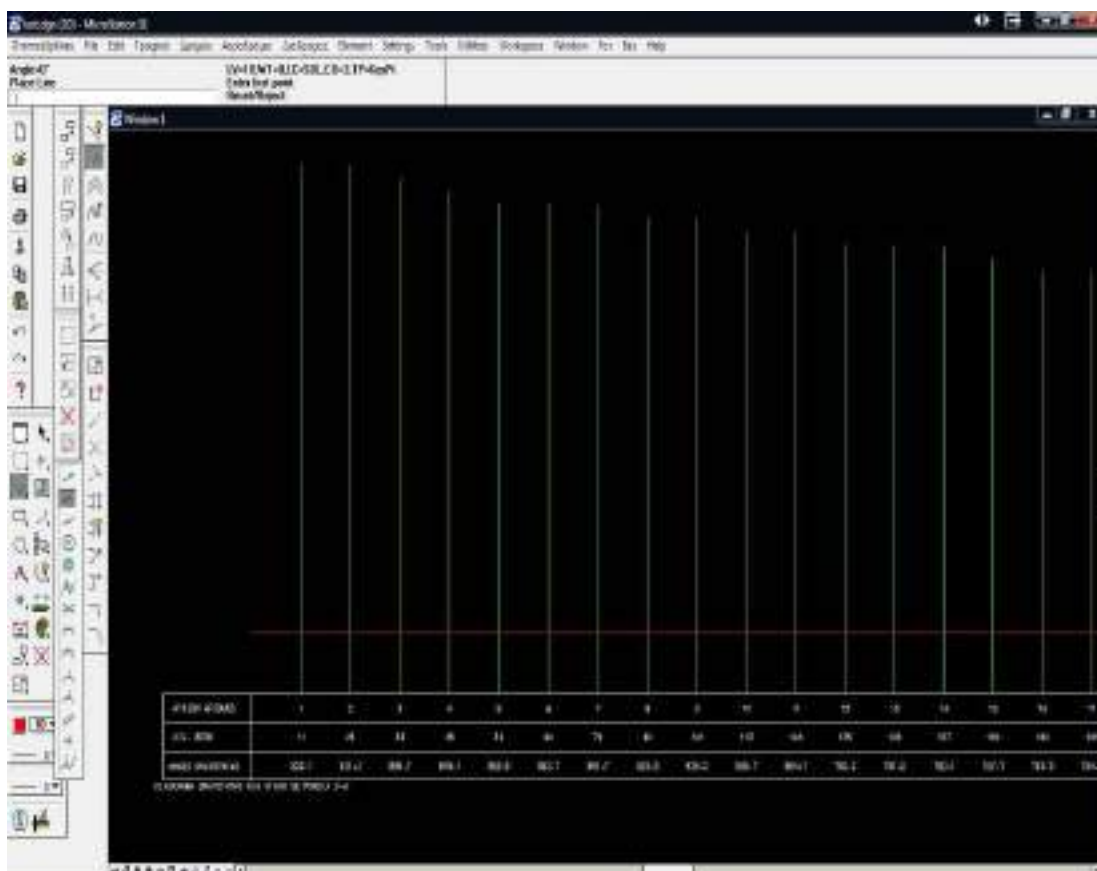
Λεπτομέρεια:



Σχήμα 5.4 Διάγραμμα Ορατότητας Για Στάση Από Δεξιά Προς Αριστερά



Λεπτομέρεια:



Στα διαγράμματα η κάθε πράσινη γραμμή αποτυπώνει την εκάστοτε ορατότητα, ενώ η κόκκινη γραμμή το όριο της ελάχιστης ορατότητας. Συνεπώς για να είναι αποδεκτός ο δρόμος με βάση τα κριτήρια που επιλέγονται, πρέπει οι πράσινες γραμμές να ξεπερνούν πάντα την κόκκινη.

5.6 ΕΛΕΓΧΟΣ ΠΙΣΤΟΤΗΤΑΣ ΧΑΡΑΞΗΣ ΟΔΟΥ

Προκειμένου να επαληθευτεί η πιστότητα του προγράμματος, θα χρησιμοποιηθεί πάνω σε ήδη χαραγμένη οδό με τις κλασσικές μεθόδους. Το συγκεκριμένο παράδειγμα αφορά στην εθνική οδό Κορίνθου – Τριπόλεως.

Στο ακόλουθο σχήμα φαίνεται η ευρύτερη περιοχή εφαρμογής.



Στη συνέχεια παραβάλλεται screenshot από στην λεπτομέρεια στην οποία δοκιμάστηκε το λογισμικό.



Είναι λοιπόν φανερό ότι το λογισμικό συμβαδίζει με τις κλασσικές μεθόδους χάραξης. Η πιστότητά του εξαρτάται από τον αριθμό των κορυφών Οριζοντιογραφίας που επιλέγει να εισάγει ο χρήστης.

Κεφάλαιο 6

Συγκριτικό παράδειγμα χάραξης

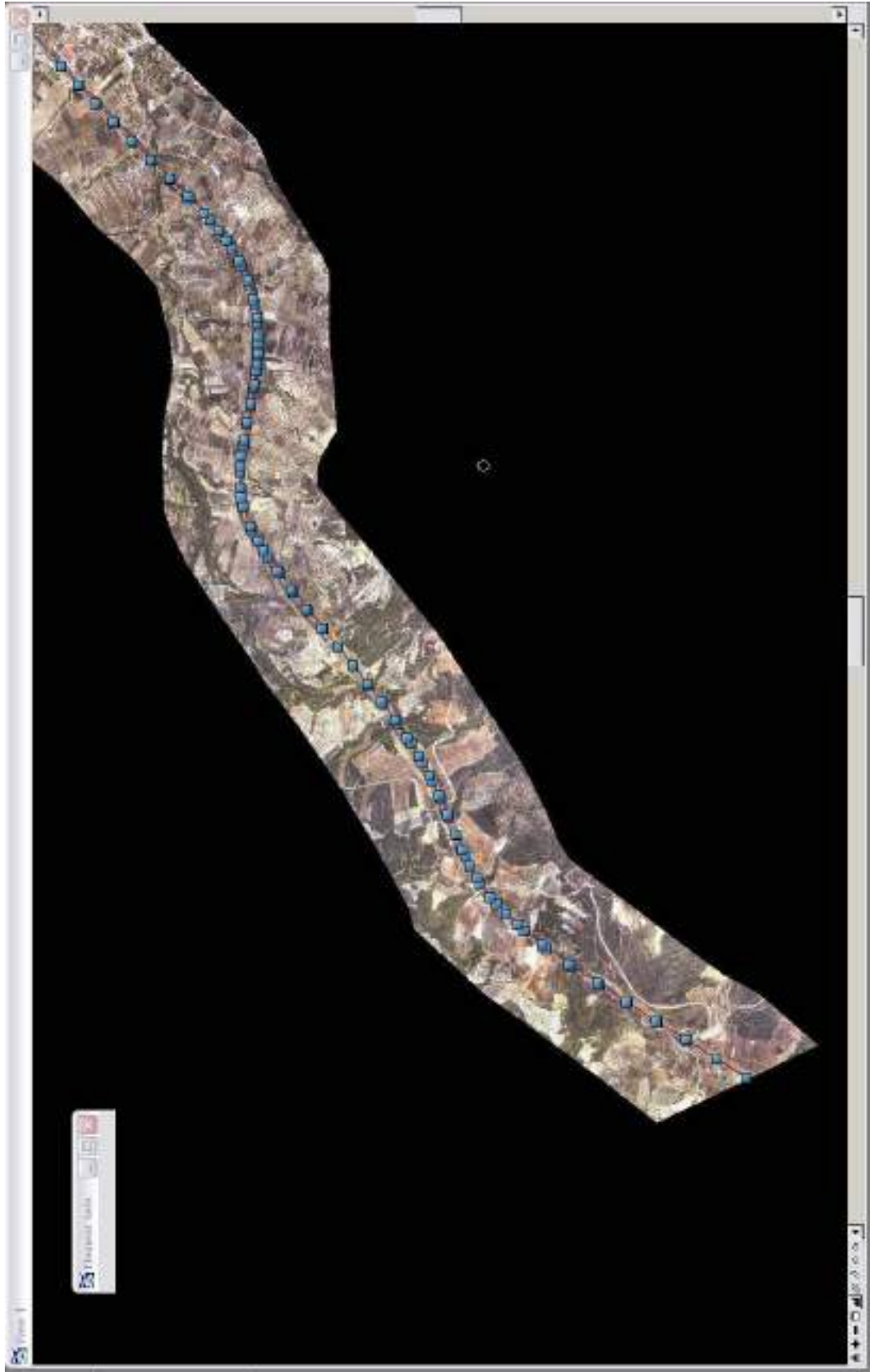
6.1 ΣΥΓΚΡΙΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ ΧΑΡΑΞΗΣ

Για τον υπολογισμό των παραδειγμάτων του κεφαλαίου αυτού χρησιμοποιήθηκε υλικό από την εταιρία ΓΕΩΓΡΑΦΙΚΗ Ε.Π.Ε και αφορά στη Μελέτη οδικού δικτύου «Κορίνθου – Τριπόλεως»

Η μελέτη της οδού είχε γίνει ήδη με συμβατικές μεθόδους και η εφαρμογή θα γίνει εκ νέου με το πακέτο «Dromos&Splines» για οδό ΑΙΠ και ταχύτητα 80 χιλιόμετρα/ώρα.

Το λογισμικό που χρησιμοποιήθηκε για τη μελέτη με τη συμβατική μέθοδο είναι δημιουργημένο από τον κ. Φώτη Μερτζάνη και θα χρησιμοποιηθεί επίσης και για τη σύγκριση αποτελεσμάτων για τον έλεγχο ορατότητας σε στάση.

Αρχικά εισάγονται οι κορυφές της καμπύλης «splines» με σκοπό την κατά το δυνατό ταύτιση της νέας χάραξης με την παλιά.



Στη συνέχεια γίνεται έλεγχος ακτίνας καμπυλότητας:



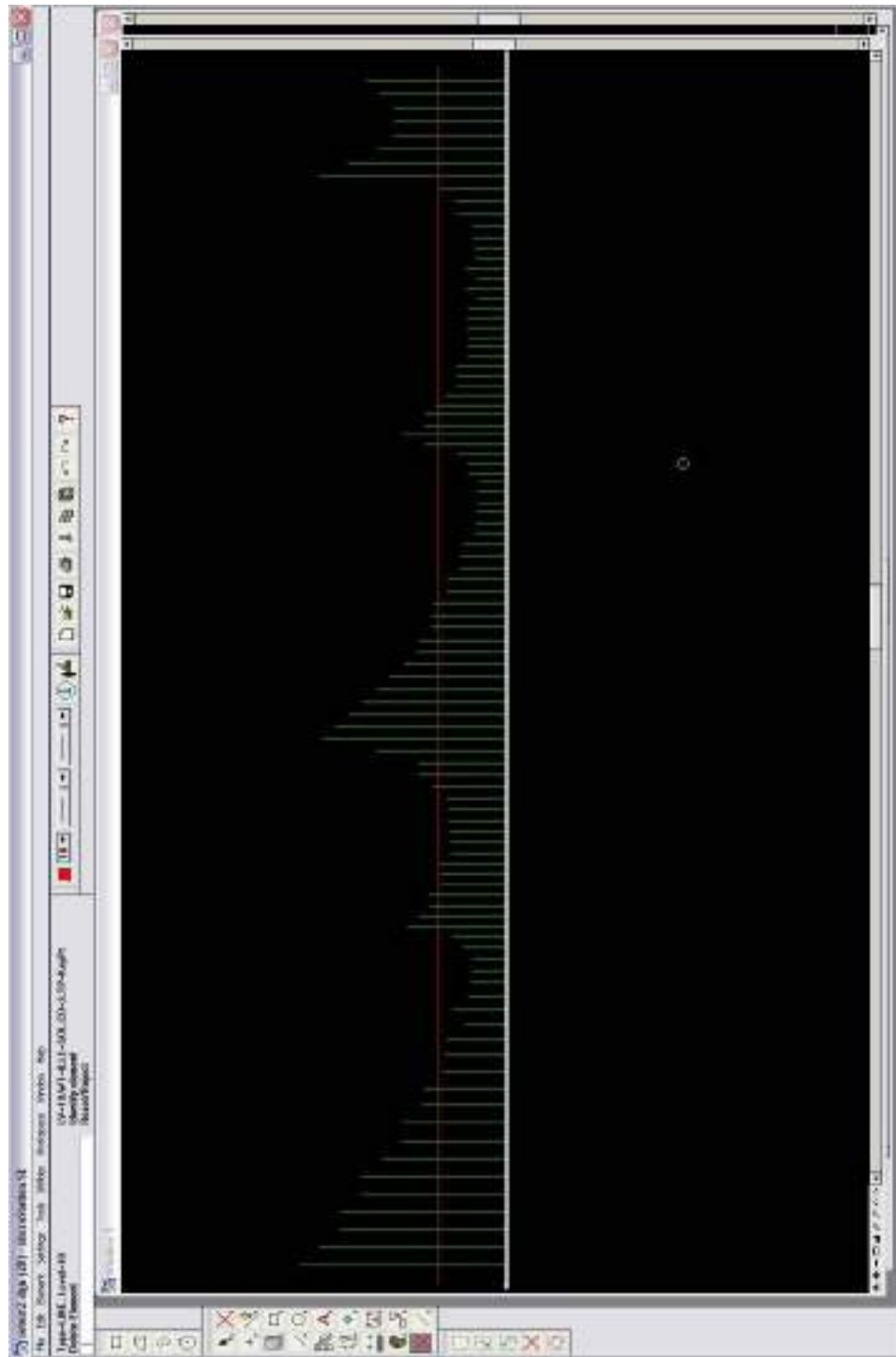


Εισαγωγή χαρακτηριστικών της οδού για τον έλεγχο ορατότητας:

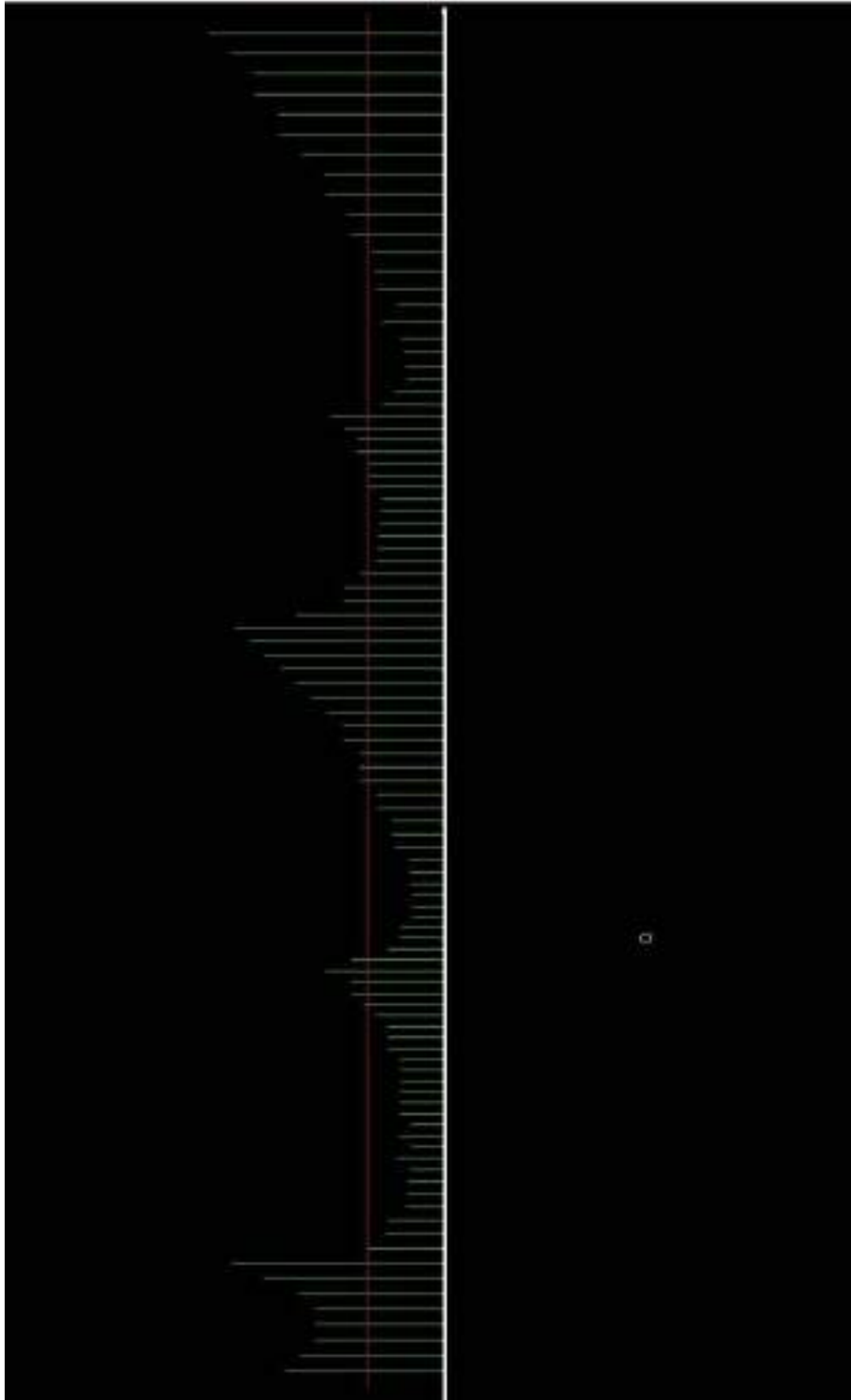


Και τέλος παρατίθενται τα διαγράμματα ορατότητας:

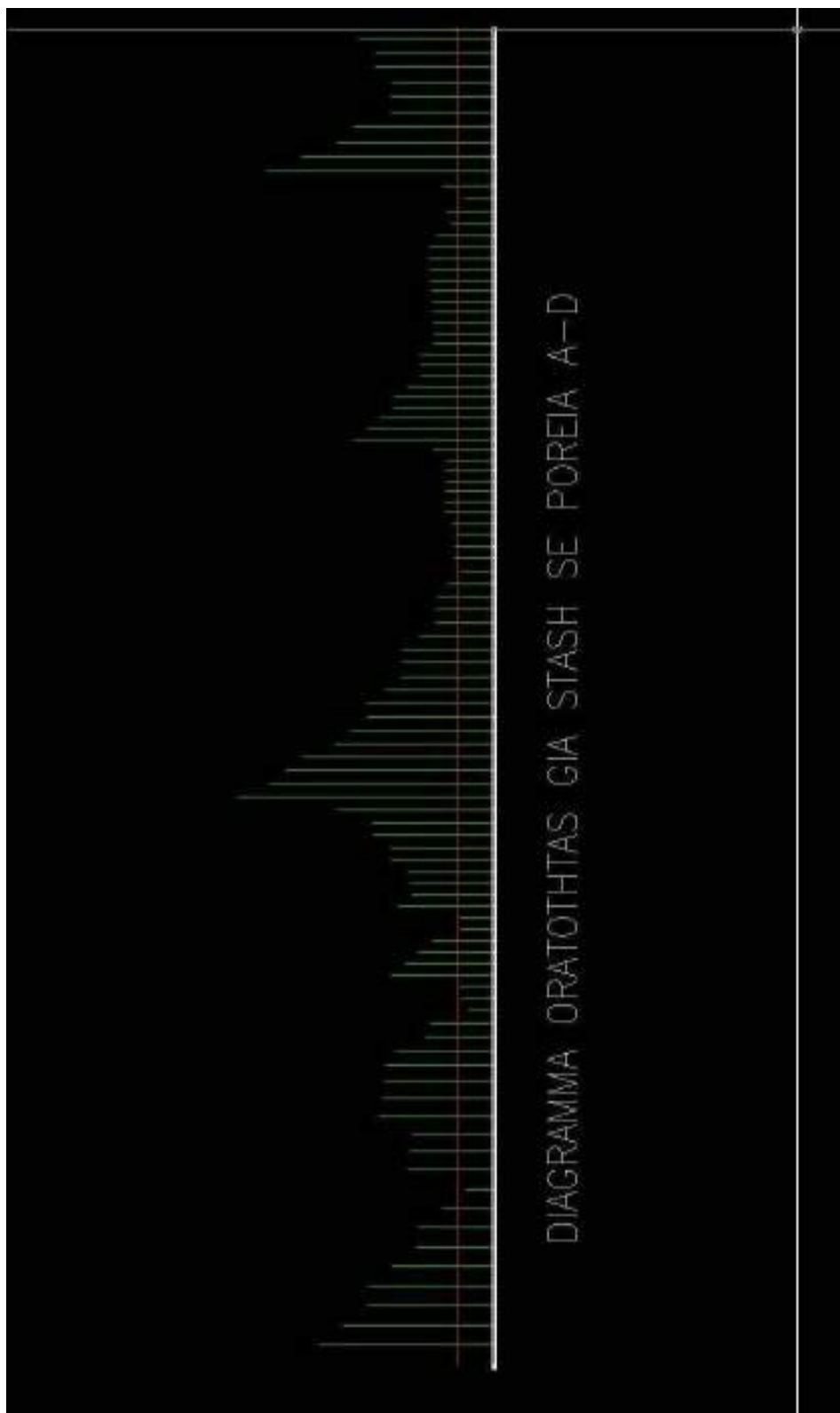
1) Ορατότητα για προσπέραση σε πορεία Α-Δ



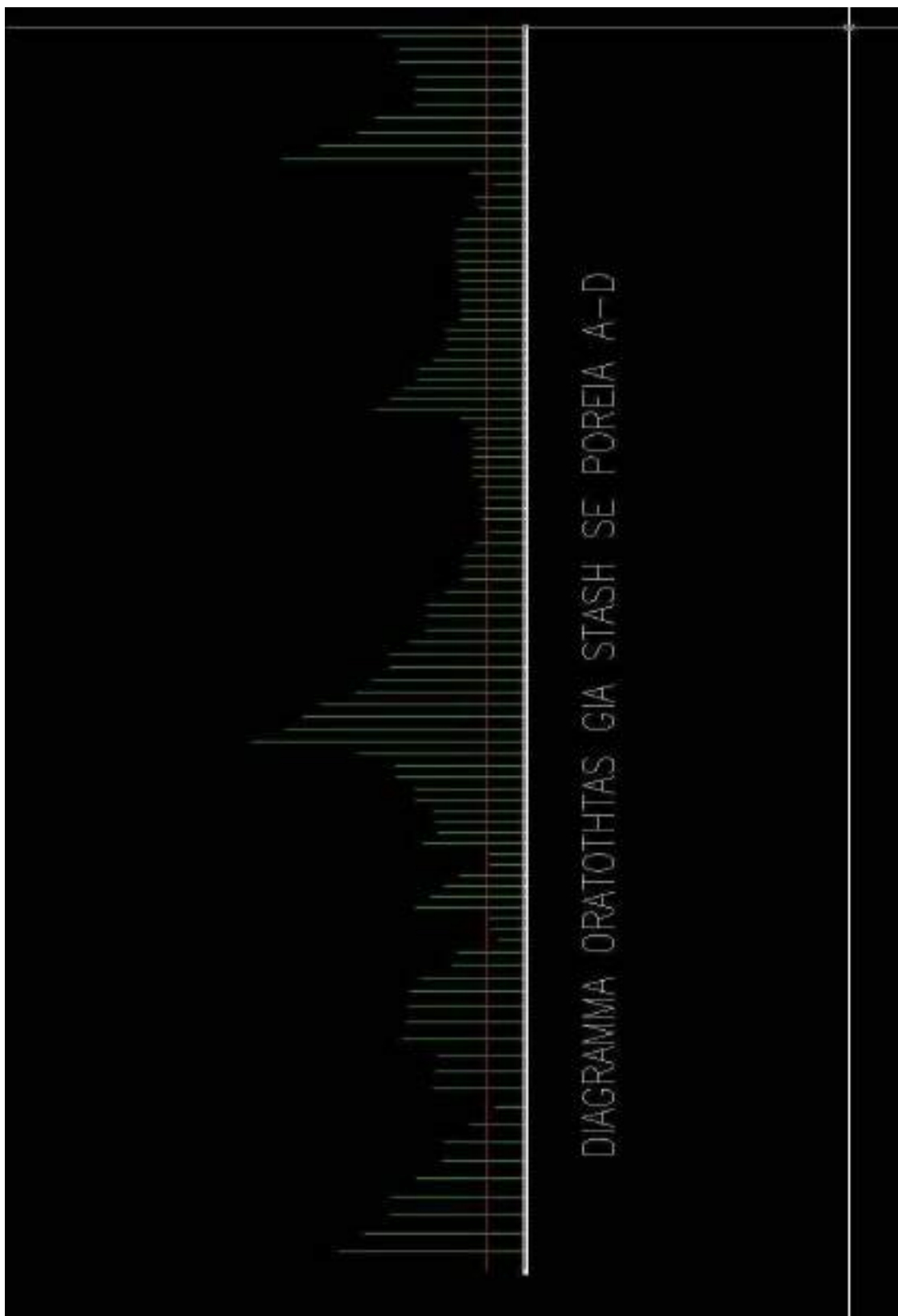
2) Ορατότητα για προσπέραση σε πορεία Δ-Α



3) Ορατότητα για στάση σε πορεία Α-Δ



4) Ορατότητα για στάση σε πορεία Δ-Α



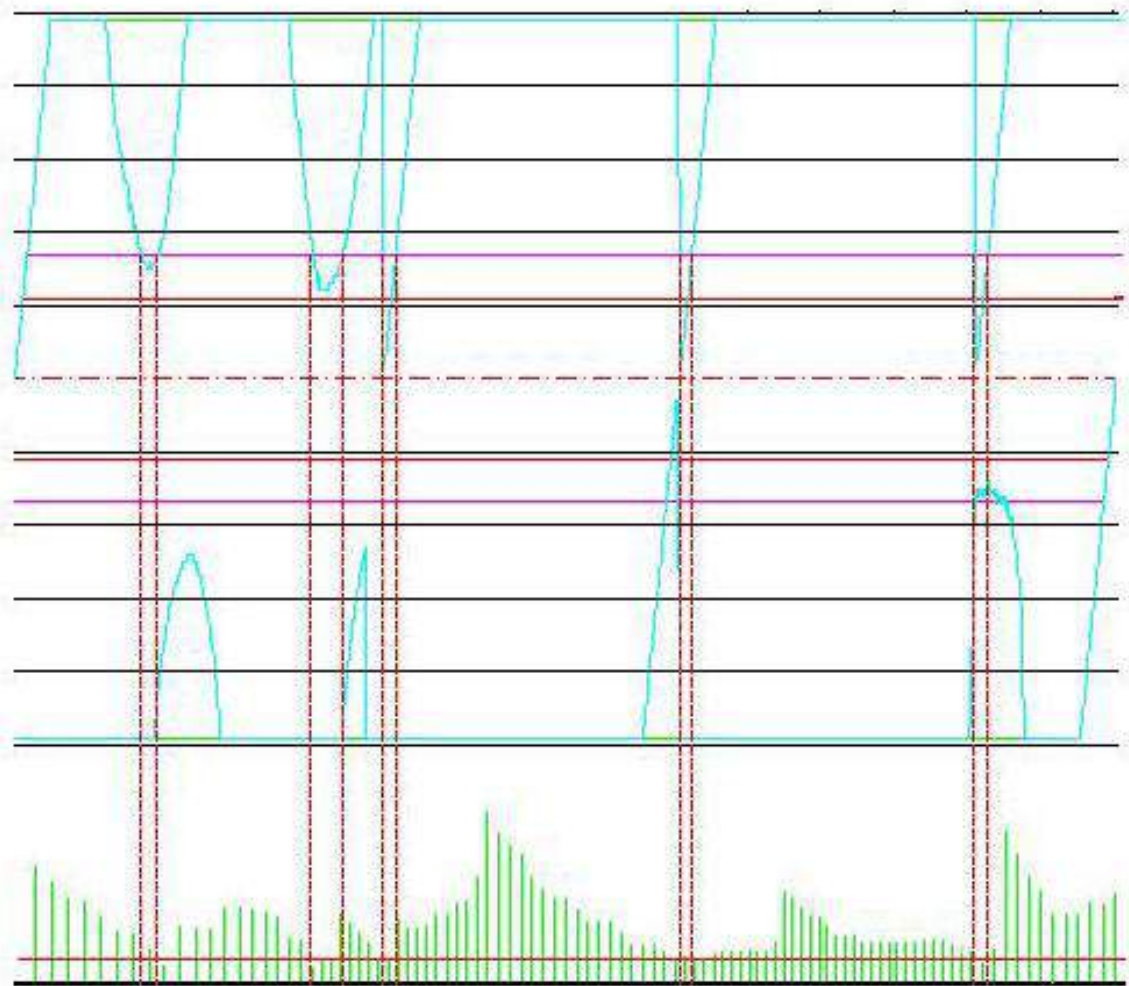


DIAGRAMMA ORATOTHTAS GIA STASH SE POREIA A-D



DIAGRAMMA ORATOTHTAS GIA STASH SE POREIA D—A

Όπως είναι φανερό από τα δύο διαγράμματα η ταύτιση των αποτελεσμάτων είναι αρκετά ικανοποιητική, καθώς φαίνεται πως οι προβληματικές περιοχές που παρουσιάζονται στη χάραξη, ταυτίζονται χιλιομετρικά.

Έλεγχος όμως σε ορατότητα προσπέρασης δεν είναι δυνατόν να γίνει, καθώς δεν βρέθηκε αντίστοιχο λογισμικό που να τον πραγματοποιεί.

Αξιόλογο είναι επίσης να σημειωθεί ότι για περισσότερη ακρίβεια, μετά την μελέτη με την spline χάραξη, τα αποτελέσματα εισήχθησαν στο λογισμικό του κυρίου

Μερτζάνη για λύση καινούρια , ακριβώς πάνω στα δεδομένα της νέας χάραξης. Αυτό επιτεύχθηκε χρησιμοποιώντας τα σημεία-αποτελέσματα που προκύπτουν μετά την επίλυση του Spline αλγόριθμου, ως Κορυφές Οριζοντιογραφίας συμβατικής χάραξης με ακτίνα Κλωθοειδούς καμπύλης μηδέν.

Συμπεράσματα

Στην παρούσα διπλωματική εργασία παρουσιάζεται μια προσπάθεια διερεύνησης της δυνατότητα χρήσης των καμπύλων Splines για τη χάραξη οριζοντιογραφίας οδού και ανάπτυξης σχετικού λογισμικού Η/Υ.

Η μορφή και οι ιδιότητες των καμπύλων Splines παρουσιάζουν μεγάλο ενδιαφέρον γι' αυτό και έχουν αρχίσει να χρησιμοποιούνται όλο και σε περισσότερους τομείς της επιστήμης. Στην οδοποιία συγκεκριμένα, παρότι δεν υπάρχει μέχρι σήμερα εκτεταμένη έρευνα πάνω στις καμπύλες Splines, έχουν αρχίσει να παρουσιάζονται έρευνες σε συνέδρια που αναφέρονται στην χρήση των καμπύλων αυτών για τον σχεδιασμό ή την προσομοίωση μιας οδού.

Είναι γεγονός ότι από την εργασία αυτή αλλά και από την υπάρχουσα βιβλιογραφία για τις εφαρμογές των καμπύλων Splines στην οδοποιία, δεν μπόρεσαν να προκύψουν σαφή συμπεράσματα για το αν οι καμπύλες αυτές υπερτερούν της συμβατικής χάραξης γιατί δεν υπήρξαν πειραματικά δεδομένα, ούτε μακροχρόνια έρευνα που θα έδινε σαφή συμπεράσματα για την ποιότητα της Spline χάραξης. Για να προκύψουν κάποια ενδεικτικά συμπεράσματα, στο πλαίσιο της παρούσης διπλωματικής εργασίας σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή ηλεκτρονικού υπολογιστή, που παρέχει στον μελετητή την δυνατότητα να σχεδιάσει μια οδό με καμπύλες Splines αλλά και να εξετάσει το αποτέλεσμα μέσω ελέγχου των καμπυλοτήτων και των μηκών ορατότητας που προκύπτουν.

Με την χρήση της εφαρμογής αυτής έγινε μια σύγκριση μιας Spline χάραξης με μία συμβατική χάραξη σε ένα πραγματικό παράδειγμα αναγνωριστικής μελέτης οδού, η

οποία έδειξε ότι μια Spline χάραξη μπορεί να παρουσιάσει ποιοτικώς τα ίδια αποτελέσματα με μια συμβατική χάραξη.

Η Spline χάραξη μπορεί να προσαρμοστεί ευκολότερα στο ανάγλυφο της περιοχής γιατί λόγω της συνεχούς και ομαλής μεταβολής της καμπυλότητας, οι μεταβολές των ακτίνων και των κλίσεων κατανέμονται σε μεγαλύτερο τμήμα οδού αλλά με όποιον ρυθμό μεταβολής επιθυμεί ο μελετητής. Αυτό το πλεονέκτημα είναι πολύ σημαντικό όχι τόσο στην χάραξη εθνικών οδών, αλλά στην χάραξη επαρχιακών οδών και ιδιαίτερα στην χώρα μας που συνήθως απαιτήση από τους φορείς ανάθεσης του έργου είναι οι ελάχιστες απαλλοτριώσεις γης και η χρήση του υπάρχοντος οδικού δικτύου.

Το ίδιο γεγονός που αποτελεί πλεονέκτημα της Spline χάραξης όμως ταυτόχρονα είναι και μειονέκτημα γιατί η ευελιξία που παρέχει στον μελετητή απαιτεί μεγάλη προσοχή και εμπειρία στην χρήση τους ώστε το αποτέλεσμα της χάραξης να είναι σωστό και ασφαλές. Η υπάρχουσα τεχνογνωσία και οι υπάρχοντες κανονισμοί δεν επαρκούν ώστε να τεθεί ένα πλαίσιο περιορισμών και οδηγιών για σωστή χρήση των καμπυλών Spline στην οδοποιία.

Συμπερασματικά μπορεί να υποστηριχθεί ότι η Spline χάραξη μπορεί να προσομοιώσει σε ικανοποιητικό βαθμό μία συμβατική χάραξη, αλλά και να παρουσιάσει μία διαφορετική μορφή χάραξης με τον κίνδυνο η μορφή αυτή να μην είναι επιβεβαιωμένα απολύτως ασφαλής, αφού δεν έχουν ακόμα διερευνηθεί πλήρως τα αποτελέσματα των ιδιοτήτων τους στην οδοποιία.

Ένα χαρακτηριστικό της Spline χάραξης που θεωρείται βασικό να ερευνηθεί είναι η κόπωση που μπορεί να επιφέρει στην συμπεριφορά του οδηγού η συνεχής μεταβολή της καμπυλότητας και η συνέπεια της απουσίας τμημάτων ευθυγραμμίων. Επίσης μια σημαντική παράμετρος της Spline χάραξης που πρέπει να ερευνηθεί είναι ο ρυθμός μεταβολής της καμπυλότητας εφόσον είναι παράμετρος που δεν υπάρχει στην συμβατική χάραξη. Για να ερευνηθούν αυτές οι παράμετροι και κυρίως η επίδραση της Spline

χάραξης στην συμπεριφορά του οδηγού είναι βασικό αρχικά να δοκιμαστεί η χάραξη μέσω ειδικών εξομοιωτών και αφού προκύψουν θετικά συμπεράσματα, να κατασκευαστεί πιλοτικά ένα τμήμα οδού μέσα σε κάποιο ερευνητικό κέντρο (π.χ. η πίστα AASHTO) ώστε να υπάρξουν πραγματικά πειραματικά στοιχεία που θα οδηγήσουν σε συγκεκριμένα συμπεράσματα.

Βιβλιογραφία

1. **The Theory of Splines and Their Applications** (J.H. Ahlberg, E.N. Nilson, J.L. Walsh)
2. **Investigation On the use of Calculus of Variations and Spline Functions in Highway Location Analysis** (P.S. Mertzanis)
3. **Computer Aided Geometric Design** (Thomas W. Sedberg)
4. **Microstation 95 SE BASIC Guide** (Bentley Corporations)
5. **Microstation 95 Quick Reference** (Robert T. McElligott)
6. **Οδηγίες Μελετών Οδικών Έργων (OMOE)** Τεύχος 3: Χαράξεις (2001)
7. **Προγράμματα Σχεδίασης σε BASIC με έμφαση στις συναρτήσεις SPLINE** (Muller/Scheller)
8. **Σχεδιασμός Μηκοτομής Διαδρόμου Αεροδρομίου με τη Χρήση Splines :**
Διπλωματική εργασία Άγγελου Γιαννίρη.
9. **Paper: Precise Road Geometry for Integrated Transport Safety Systems,** 5th
Swiss Transport Research Conference, Monte Verita / Ascona, March 2005.
(Gontran H., Gilliéron P.-Y., Skaloud J.)
10. **Modeling a Road Using Spline Interpolation. Reports of Computational Mathematics #145, Department of Mathematics, University of Iowa, USA 2002.** (Atkinson K).
11. **Paper : The Basics of a Three – Dimensional Geometric Design Methodology,**
3rd International Symposium on Highway Geometric Design, June 2005. (Kuhn Wolfgang).
12. **Paper : New Complex Solutions for the Real Time Spatial Road Design,** 3rd
International Symposium on Highway Geometric Design, June 2005.(Tilger Klaus)

13. **Norme funzionali e geometriche per la costruzione delle strade.** Decreto Del Ministero Delle Infrastrutture E Dei Trasporti del 5 novembre 2001.
14. **Χάραξη οδού στο χώρο με χρήση καμπυλών ελεύθερης μορφής «B-Splines»**
Διπλωματική εργασία Αθανασόπουλου Μιχάλη.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΕΤΑΦΟΡΩΝ & ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΥΠΟΔΟΜΗΣ

Επιβλέπων Καθηγητής: Γ. Κανελλαΐδης
Συνεπιβλέπων : Φ.Μερτζάνης
Ονοματεπώνυμο: Θεοφανίδης Αχιλλέας – Μάριος

ΣΧΕΔΙΑΣΜΟΣ ΟΡΙΖΟΝΤΙΟΓΡΑΦΙΑΣ ΟΔΟΥ ΜΕ ΧΡΗΣΗ ΚΑΜΠΥΛΩΝ ΕΛΕΥΘΕΡΗΣ ΜΟΡΦΗΣ ΤΥΠΟΥ “SPLINES”

ΠΑΡΑΡΤΗΜΑ

ΙΟΥΛΙΟΣ 2008

Sub main

Dim n(500) As string

Dim x(500) as double

Dim y(500) as double

Dim ws as string

Dim point As MbePoint

Dim ds as double 'platos dromou

Dim H(500) as double

Dim Q(500) as double

Dim P(500) as double

Dim R(500) as double

Dim S(500) as double

Dim Z as double

Dim XM(500) as double

Dim XL(5000) as double

Dim YL(5000) as double

Dim XX(5000) as double

Dim YDL(5000) as double

Dim YDDL(5000) as double

Dim YDLL(5000) as double

Dim CURV(5000) as double

Dim XLSTEP as double

Dim SYDDL as double

Dim startPoint As MbePoint

Dim point2 As MbePoint

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 1"

MbeSendCommand "LOCK LEVEL TOGGLE "

MbeSendCommand "ACTIVE LEVEL 1"

MbeSendKeyIn "LV=1"

MbeSendKeyIn "LC=0"

MbeSendKeyIn "CO=0"

MbeSendKeyIn "WT=0"

MbeSendCommand "CHOOSE ELEMENT "

startPoint.x = -1000#

startPoint.y = 2000#

startPoint.z = 0#

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

point2.x = point.x + 7000#

point2.y = point.y - 5000#

point2.z = point.z

MbeSendDragPoints point, point2, 1%

'MbeSendCommand "DELETE ELEMENT "

'MbeSendReset

MbeSendCommand "FIT VIEW EXTENDED 1%"

MbeSettings.TextJustification=2

MbeSettings.Font=78

XLSTEP=10

actionButton = mbeOpenModalDialog (1)

if actionbutton=4 then

' MBEMESSAGEBOX "CANCELED"

goto TELOS

end if

if actionbutton=3 then

goto ARXH

end if

ARXH:

open "c:\aggelos\XLSTEP.dat" for output as 1

write #1,XLSTEP, ds

close #1

MbeSendKeyin "co=0"

open ws+ "fm.hor" for input as 1

i=1

Do while not Eof(1)

input #1,n(i),x(i),y(i)

i=i+1

Loop

close #1

T=i-1

- ```
' Send a keyin that can be a command string
' MbeSendKeyin "place curve picon"
```

```

H(1)=x(2)-x(1)
for j = 2 to T-1
 K=j-1
 H(j)=x(j+1)-x(j)
 Q(K)=2
 R(K)=H(j)/(H(K)+H(j))
 P(K)=1-R(K)
 S(K)=6*((y(j+1)-y(j))/H(j)-(y(j)-y(K))/H(K))/(H(K)+H(j))
next j

```

[illegible]

```

XM(T-2)=S(T-2)/Q(T-2)
for j = 2 to T-2
j1=T-1-j
XM(j1)=(S(j1)-R(j1)*XM(j1+1))/Q(j1)

```

next j

for j = 2 to T-1

$XM(T+1-j)=XM(T-j)$

next j

$XM(1)=0$

$XM(T)=0$

$XL(1)=X(1)$

$YL(1)=Y(1)$

$MAX=(x(T)-x(1))/XLSTEP+1$

open "c:\aggelos\sp2.dat" for output as 1

write #1,MAX

close #1

for j2 = 2 to MAX

$XL(j2)=XL(j2-1)+XLSTEP$

for j1 = 1 to T-1

if  $XL(j2) \geq X(j1)$  and  $XL(j2) < X(j1+1)$  then

j=j1+1

end if

next j1

$YL(j2)=XM(j-1)*(X(j)-XL(j2))^3/(6*H(j-1))+XM(j)*(XL(j2)-x(j-1))^3/(6*H(j-1))+(y(j-1)-(XM(j-1)*H(j-1)^2/6))*(x(j)-XL(j2))/H(j-1)+(y(j)-XM(j)*H(j-1)^2/6)*(XL(j2)-x(j-1))/H(j-1)$

$YDL(j2)=-XM(j-1)*(X(j)-XL(j2))^2/(2*H(j-1))+XM(j)*(XL(j2)-x(j-1))^2/(2*H(j-1))+(y(j)-y(j-1))/H(j-1)-(XM(j)-XM(j-1))*H(j-1)/6$

$YDDL(j2)=XM(j-1)*(X(j)-XL(j2))/H(j-1)-XM(j)*(XL(j2)-x(j-1))/H(j-1)$

if  $YDDL(j2) \neq 0$  then

$CURV(j2)=(((1+YDL(j2)^2)^{1.5}/YDDL(j2))^2)^{(1/2)}$

else

**CURV(j2)=10000000**

**end if**

**if CURV(j2)=0 then CURV(j2)=10000000**

**next j2**

**YDL(1)=(y(2)-y(1))/H(1)-H(1)\*XM(2)/6**

**'YDDL(1)=10000000**

**open "c:\aggelos\sp3.dat" for output as 1**

**for j3 = 1 to MAX**

**write #1,XL(j3),YL(j3),YDL(j3),YDDL(j3),CURV(j3)**

**next j3**

**close #1**

**open "c:\aggelos\curv.dat" for output as 1**

**for j4 = 1 to MAX**

**write #1,CURV(j4)**

**next j4**

**close #1**

**MBESENDCOMMAND "PLACE SMARTLINE"**

**for j3 = 1 to MAX**

**MBESENDDATAPOINT XL(j3),YL(j3),0,1%**

**next j3**



open "c:\aggelos\YDLL.dat" for output as 1

SYDDL=0

for i = 1 to MAX

SYDDL=SYDDL+ABS(YDDL(i))

if YDDL(i)=0 then

YDLL(i)=10000000

Goto EPOM

end if

YDLL(i)=(1+YDL(i)^2)^1.5/YDDL(i)

EPOM:

write #1, ABS(YDLL(i))

next i

close #1

'EDW KANW PAREMBASH

MbeSendCommand "CHOOSE ELEMENT"

' Coordinates are in master units

startPoint.x = XL(3)

startPoint.y = YL(3)

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = XL(3)

point.y = YL(3)

point.z = 0#

MbeSendDataPoint point, 1%

MbeSendCommand "FIT VIEW EXTENDED 1%"

MbeSendCommand "ACTIVE COLOR 3"

MbeSendCommand "CHANGE COLOR "

MbeSendCommand "ACTIVE STYLE 4"

MbeSendCommand "CHANGE STYLE "

mbesendreset

MbeSendCommand "CHOOSE ELEMENT"

' Coordinates are in master units

startPoint.x = 0

startPoint.y = 0

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = 0

point.y = 0

point.z = 0#

MbeSendDataPoint point, 1%

mbesendreset

TELOS:

open ws+"max.dat" for output as 10

write #10, max

close #10

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 1"**

**MbeSendCommand "FIT VIEW EXTENDED 1%"**

**End Sub**

## Horafi.bas

---

'-----

' Horizontal Alignment - hor.bas - 15/11/2002

'-----

Const CrLf = Chr\$(13) + Chr\$(10)

Sub Main

dim ws as string           ' Working\_Directory

dim v1(500) as string      ' Name

dim v2(500) as double      ' X

dim v3(500) as double      ' Y

dim v7(500) as double      ' Superelevation

dim v8(500) as integer      ' Index

dim n(20) as string        ' Name

dim x(20) as double        ' X

dim y(20) as double        ' Y

dim tv as integer          ' Total\_HPIs

dim tt as integer          ' Total\_Tables

dim tb as integer          ' Active\_Table

dim ni as string           ' Roll\_Down

dim ns as string           ' Active

dim nz as string           ' Zero\_String

dim zr as double           ' Zero\_Double

dim ic as integer          ' Check

ws= "c:\aggelos\"

pnt2:

```
if fileexists (ws+"fm.hor") then
filecopy ws+"fm.hor" , ws+"fmhor.bak"
else
MbeMessageBox "Νέο Αρχείο Κορυφών Οριζοντιογραφίας fm.hor"
open ws+"fm.hor" for output as 1
close #1
end if
```

```
open ws+"fm.hor" for input as 1
i=0
do while not eof(1)
i=i+1
input #1,v1(i),v2(i),v3(i)
loop
tt=int(i/20)+1
close #1
tv=i
tb=1
```

pnt1:

```
for i = 1 to tv-1
for j = i+1 to tv
if i <> j then
if v1(i)=v1(j) then
MbeMessageBox " Ιδια Ονομασία Κορυφών "+v1(i)+"/" +v1(j)
end if
end if
next j
next i
```

```

for i = 1 to 20
n(i)=""
x(i)=0
y(i)=0
next i
for i = 1 to 20
if (tb-1)*20+i > tv then goto pnt3
n(i)=v1((tb-1)*20+i)
x(i)=v2((tb-1)*20+i)
y(i)=v3((tb-1)*20+i)

next i
pnt3:
ActionButton=MbeOpenModalDialog(13)
for i = 1 to 20
v1((tb-1)*20+i)=n(i)
v2((tb-1)*20+i)=x(i)
v3((tb-1)*20+i)=y(i)
next i

'OK
if ActionButton=9001 then
open ws+"fm.hor" for output as 1
for i = 1 to 500

if v1(i) ="0" then
v2(i)=0
v3(i)=0

end if

```

if abs(v2(i))+abs(v3(i)) > 0.00001 then

if i =1 then

write #1,v1(i),v2(i),v3(i)

end if

if i <>1 then

if v1(i) <> v1(i-1) or v2(i) <> v2(i-1) or v3(i) <> v3(i-1) then

write #1,v1(i),v2(i),v3(i)

end if

end if

end if

next i

close #1

open ws+"fm.hor" for input as 1

i=0

do while not eof(1)

i=i+1

input #1,v1(i),v2(i),v3(i)

loop

close #1

if i=0 then kill ws+"fm.hor"

tv=i

goto pnt1

end if

**'NEXT**

**if ActionButton=9002 then**

**tb=tb+1**

**goto pnt1**

**end if**

**'PREVIOUS**

**if ActionButton=9003 then**

**if tb = 1 then**

**MbeMessageBox "Δεν Υπάρχει Προηγούμενη"**

**goto pnt1**

**end if**

**tb=tb-1**

**goto pnt1**

**end if**

**'FIRST**

**if ActionButton=9004 then**

**tb=1**

**goto pnt1**

**end if**

**'LAST**

**if ActionButton=9005 then**

**tb=tt**

**goto pnt1**

**end if**

**'INSERT**

**if ActionButton=9006 then**

**ic=0**



```

open "\fm\work" for output as 2
for i = 1 to tv
if ni=v1(i) then
write #2,nz,zr,zr,zr,zr,zr,zr
ic=1
tb=int(i/20)+1
end if
if abs(v2(i))+abs(v3(i))> 0.00001 then
write #2,v1(i),v2(i),v3(i)
end if
next i
close #2
filecopy "\fm\work" , ws+"fm.hor"
if ic=0 then MbeMessageBox "Δεν Βρέθηκε Η "+ni
open ws+"fm.hor" for input as 1
i=0
do while not eof(1)
i=i+1
input #1,v1(i),v2(i),v3(i)
loop
tt=int(i/20)+1
close #1
tv=i
goto pnt1
end if

```

'ACTIVE

```

if ActionButton=9007 then
for i = 1 to tv
if ns=v1(i) then
tb=int(i/20)+1

```

```
goto pnt1
end if
next i
MbeMessageBox "Δεν Βρέθηκε Η "+ns
goto pnt2
end if
```

'CANCEL

```
if ActionButton=9008 then
if fileexists (ws+"fm.hor") then
open ws+"fm.hor" for input as 1
i=0
do while not eof(1)
i=i+1
input #1,v1(i),v2(i),v3(i)
loop
close #1
if i=0 then kill ws+"fm.hor"
end if
mbewriteprompt "ΑΥΤΕ ΓΕΙΛΑ - hor.bas"
end
end if
```

'INFORMATION

```
if ActionButton=9009 then
tb=tb
ActionButton=MbeOpenModalDialog(9)
goto pnt1
end if
```

End Sub

**sub main**

**dim XL(5000) as double**

**dim YL(5000) as double**

**dim ds as double**

**dim a1(5000) as double**

**dim a2(5000) as double**

**dim a3(5000) as double**

**dim a4(5000) as double**

**Dim YDL(5000) as double**

**Dim YDDL(5000) as double**

**dim CURV(5000) as double**

**dim ws as string**

**dim max as double**

**dim x1 as double**

**dim x2 as double**

**dim x3 as double**

**dim x4 as double**

**dim x5 as double**

**dim x6 as double**

**dim y1 as double**

**dim y2 as double**

**dim y3 as double**

**dim y4 as double**

**dim y5 as double**

**dim y6 as double**

**dim xtp(5000) as double**

**dim ytp(5000) as double**

**dim xtk(5000) as double**

dim ytk(5000) as double  
dim DIST as double  
dim x(5000) as double  
dim y(5000) as double  
dim YB(2000) as double  
dim YO(2000) as double  
dim XB(2000) as double  
dim XO(2000) as double  
dim a(2000) as double  
dim b(2000) as double  
dim c(2000) as double  
dim d(2000) as double  
dim e(2000) as double  
dim f(2000) as double  
'dim g(5000) as double  
dim YK(5000) as double  
dim XK(5000) as double  
dim YKO(2000) as double  
dim XKO(2000) as double  
dim YBO(2000) as double  
dim XBO(2000) as double

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 6"

MbeSendCommand "LOCK LEVEL TOGGLE "

MbeSendCommand "ACTIVE LEVEL 6"

MbeSendKeyIn "LV=6"

MbeSendKeyIn "LC=0"

**MbeSendKeyIn "CO=0"**

**MbeSendKeyIn "WT=0"**

**open ws+"XLSTEP.dat" for input as 2**

**input #2, XLSTEP, ds**

**close #2**

**open ws+"max.dat" for input as 5**

**input #5, max**

**close #5**

**i=1**

**MBESENDCOMMAND "PLACE SMARTLINE"**

**open ws+"sp3.dat" for input as 1**

**do while not eof(1)**

**input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)**

**i=i+1**

**loop**

**close #1**

**open ws+"blemap.dat" for output as 10**

**open ws+"blemak2.dat" for output as 15**

**open ws+"oriop.dat" for output as 7**

**for i=1 to max-1**

$a1(i) = (YL(i+1) - YL(i)) / (XL(i+1) - XL(i))$

$a(i) = (((XL(i+1) - XL(i))^2 + (YL(i+1) - YL(i))^2))^{1/2}$

$c(i) = YL(i+1) - YL(i)$

$b(i) = XL(i+1) - XL(i)$

$XB(i) = XL(i) - (a1(i) * ds) * (b(i) / a(i)) + ds * (c(i) / a(i))$

$YB(i) = YL(i) + (a1(i) * ds) * (c(i) / a(i)) + ds * (b(i) / a(i))$

$XBO(i) = XL(i) - (a1(i) * (ds/2)) * (b(i) / a(i)) + (ds/2) * (c(i) / a(i))$

$YBO(i) = YL(i) + (a1(i) * (ds/2)) * (c(i) / a(i)) + (ds/2) * (b(i) / a(i))$

write #7, XB(i), YB(i)

write #10, XBO(i), YBO(i)

write #15, XBO(i), YBO(i)

MBESENDDATAPOINT XB(i), YB(i), 0, 1%

next i

close #10

close #7

mbesendreset

open ws+"blemak.dat" for output as 9

open ws+"blemap2.dat" for output as 16

open ws+"oriok.dat" for output as 8

for i=1 to max-1

$XK(i) = XL(i) + (a1(i) * ds) * (b(i)/a(i)) + ds * (c(i)/a(i))$

$YK(i) = YL(i) + (a1(i) * ds) * (c(i)/a(i)) - ds * (b(i)/a(i))$

$XKO(i) = XL(i) + (a1(i) * (ds/2)) * (b(i)/a(i)) + (ds/2) * (c(i)/a(i))$

$YKO(i) = YL(i) + (a1(i) * (ds/2)) * (c(i)/a(i)) - (ds/2) * (b(i)/a(i))$

write #8,XK(i),YK(i)

write #9,XKO(i),YKO(i)

write #16,XKO(i),YKO(i)

MBESENDATAPOINT XK(i),YK(i),0,1%

next i

close #8

close #9

mbesendreset

end sub

---

## Morf1.bas

---

### Sub Main

```
dim ws as string ' Working_Directory
dim v1(500) as string ' Name
dim v2(500) as double ' X
dim v3(500) as double ' Y
dim v7(500) as double ' Superelevation
dim v8(500) as integer ' Index
dim n(20) as string ' Name
dim x(20) as double ' X
dim y(20) as double ' Y
dim k as double
dim tv as integer ' Total_HPIs
dim tt as integer ' Total_Tables
dim tb as integer ' Active_Table
dim ni as string ' Roll_Down
dim ns as string ' Active
dim nz as string ' Zero_String
dim zr as double ' Zero_Double
dim ic as integer ' Check
```

```
ws= "c:\aggelos\"
```

```
pnt2:
```

```
if fileexists (ws+"morf.dat") then
filecopy ws+"morf.dat" , ws+"morf.bak"
else
```



**MbeMessageBox "Νέο Αρχείο Κορυφών Οριζοντιογραφίας morf.dat"**

open ws+"morf.dat" for output as 1

close #1

end if

open ws+"morf.dat" for input as 1

i=0

do while not eof(1)

i=i+1

input #1,v1(i),v2(i),v3(i)

loop

tt=int(i/20)+1

close #1

tv=i

tb=1

pnt1:

for i = 1 to tv-1

for j = i+1 to tv

if i <> j then

if v1(i)=v1(j) then

**MbeMessageBox " Ιδια Ονομασία Κορυφών "+v1(i)+"/" +v1(j)**

end if

end if

next j

next i

for i = 1 to 20

n(i)=""

x(i)=0

**y(i)=0**

**next i**

**for i = 1 to 20**

**if (tb-1)\*20+i > tv then goto pnt3**

**n(i)=v1((tb-1)\*20+i)**

**x(i)=v2((tb-1)\*20+i)**

**y(i)=v3((tb-1)\*20+i)**

**next i**

**pnt3:**

**ActionButton=MbeOpenModalDialog(3)**

**for i = 1 to 20**

**v1((tb-1)\*20+i)=n(i)**

**v2((tb-1)\*20+i)=x(i)**

**v3((tb-1)\*20+i)=y(i)**

**next i**

**'OK**

**if ActionButton=9001 then**

**open ws+"morf.dat" for output as 1**

**for i = 1 to 500**

**if v1(i) ="0" then**

**v2(i)=0**

**v3(i)=0**

**end if**

```
if abs(v2(i))+abs(v3(i)) > 0.00001 then
```

```
if i =1 then
```

```
write #1,v1(i),v2(i),v3(i)
```

```
end if
```

```
if i <>1 then
```

```
if v1(i) <> v1(i-1) or v2(i) <> v2(i-1) or v3(i) <> v3(i-1) then
```

```
write #1,v1(i),v2(i),v3(i)
```

```
end if
```

```
end if
```

```
end if
```

```
next i
```

```
close #1
```

```
open ws+"morf.dat" for input as 1
```

```
i=0
```

```
do while not eof(1)
```

```
i=i+1
```

```
input #1,v1(i),v2(i),v3(i)
```

```
loop
```

```
close #1
```

```
if i=0 then kill ws+"morf.dat"
```

```
tv=i
```

```
goto pnt1
```

```
end if
```

```
'NEXT
```

if ActionButton=9002 then

tb=tb+1

goto pnt1

end if

'PREVIOUS

if ActionButton=9003 then

if tb = 1 then

MbeMessageBox "Δεν Υπάρχει Προηγούμενη"

goto pnt1

end if

tb=tb-1

goto pnt1

end if

'FIRST

if ActionButton=9004 then

tb=1

goto pnt1

end if

'LAST

if ActionButton=9005 then

tb=tt

goto pnt1

end if

'INSERT

if ActionButton=9006 then

ic=0

open "\fm\work" for output as 2

```

for i = 1 to tv
if ni=v1(i) then
write #2,nz,zr,zr,zr,zr,zr,zr
ic=1
tb=int(i/20)+1
end if
if abs(v2(i))+abs(v3(i))> 0.00001 then
write #2,v1(i),v2(i),v3(i)
end if
next i
close #2
filecopy "\\fm\work" , ws+"morf.dat"
if ic=0 then MbeMessageBox "Δεν Βρέθηκε Η "+ni
open ws+"morf.dat" for input as 1
i=0
do while not eof(1)
i=i+1
input #1,v1(i),v2(i),v3(i)
loop
tt=int(i/20)+1
close #1
tv=i
goto pnt1
end if

```

'ACTIVE

```

if ActionButton=9007 then
for i = 1 to tv
if ns=v1(i) then
tb=int(i/20)+1
goto pnt1

```

```
end if
next i
MbeMessageBox "Δεν Βρέθηκε Η "+ns
goto pnt2
end if
```

```
'CANCEL
```

```
if ActionButton=9008 then
if fileexists (ws+"morf.dat") then
open ws+"morf.dat" for input as 1
i=0
do while not eof(1)
i=i+1
input #1,v1(i),v2(i),v3(i)
loop
close #1
if i=0 then kill ws+"morf.dat"
end if
mbewriteprompt "Αντε Γειά - morf.bas"
end
end if
```

```
'INFORMATION
```

```
if ActionButton=9009 then
tb=tb
ActionButton=MbeOpenModalDialog(9)
goto pnt1
end i
```

```
End Sub
```

## Morfgraf.bas

---

```
sub main

dim x(5000) as double
dim y(5000) as double
dim k as double
dim ws as string
dim n(5000) as double
dim z(5000) as double
Dim startPoint As MbePoint
Dim point As MbePoint, point2 As MbePoint

MbeSendKeyIn "LV=3"
MbeSendKeyIn "LC=1"
MbeSendKeyIn "CO=15"
MbeSendKeyIn "WT=10"

ws= "c:\aggelos\"

open ws+ "morf.dat" for input as 1
mbesendcommand "place smartline"
i=1
Do while not Eof(1)
input #1,n(i),x(i),y(i)
MBESENDDATAPOINT x(i),y(i),0,1%
i=i+1
Loop
close #1
```

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 1"**

**MbeSendCommand "FIT VIEW EXTENDED 1%"**

**end sub**

---



**sub main**

**dim x(5000) as double**

**dim y(5000) as double**

**dim XL(5000) as double**

**dim YL(5000) as double**

**dim YDL(5000) as double**

**dim YDDL(5000) as double**

**dim a2(5000) as double**

**dim b2(5000) as double**

**dim xp(5000) as double**

**dim yp(5000) as double**

**dim CURV(5000) as double**

**dim n(5000) as double**

**dim ws as string**

**dim ANGLE as double**

**dim angstring as string**

**dim tp(5000) as double**

**Dim startPoint As MbePoint**

**Dim point As MbePoint, point2 As MbePoint**

**dim sum(5000) as double**

**MbeSendCommand "ACTIVE LEVEL 5"**

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 5"**

**MbeSendKeyIn "LV=5"**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=4"**

**MbeSendKeyIn "WT=0"**

**ws="c:\aggelos\"**

**open ws+"sp3.dat" for input as 1**

**i=1**

**Do while not Eof(1)**

**input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)**

**open ws+ "morf.dat" for input as 2**

**j=1**

**Do while not Eof(2)**

**input #2,n(j),x(j),y(j)**

**mbesendcommand "place smartline"**

**mbesenddatapoint XL(i),YL(i),0,1%**

**mbesenddatapoint x(j),y(j),0,1%**

**'mbesendreset**

**j=j+1**

**Loop**

**close #2**

**i=i+1**

**Loop**

**close #1**

**max=j-2**

**open ws+"sp3.dat" for input as 5**

**i=1**

**Do while not Eof(5)**

**input #5,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)**

**for j=1 to max**

MbeSendCommand "MEASURE ANGLE "

mbesenddatapoint (x(j)+XL(i))/2,(y(j)+YL(i))/2,0,1%

mbesenddatapoint (x(j+1)+XL(i))/2,(y(j+1)+YL(i))/2,0,1%

mbesenddatapoint (x(j+1)+XL(i))/2,(y(j+1)+YL(i))/2,0,1%

angle = MbeState.measureResult1

sum(i)=sum(i)+angle

if sum(i)>358.5 then

tp(i)=1

else

tp(i)=2

end if

next j

print sum(i)

i=i+1

Loop

maxi=i-1

close #5

open ws+"tmorf.dat" for output as 6

for i=1 to maxi

write #6,XL(i),YL(i),YDL(i),YDDL(i),CURV(i),tp(i)

next i

close #6

end sub

## omoe.bas

---

'-----

'OMOE - omoe.bas - 8/12/2002

'-----

Const CrLf = Chr\$(13) + Chr\$(10)

sub main

'GENERAL

dim ws as string ' Working Subdirector

dim ve as double ' Speed

dim rc as string ' Road Code

dim tp(1000) as double ' Terrain Morphology

dim XL(1000) as double

dim YL(1000) as double

dim YDL(1000) as double

dim YDDL(1000) as double

dim CURV(1000) as double

dim qi as string

dim rm(1000) as double

dim max as double

ws= "c:\aggelos\"

i=1

open ws+"tmorf.dat" for input as 1

do while not eof (1)

input #1, XL(i),YL(i),YDL(i),YDDL(i),CURV(i),tp(i)

'print CURV(i),tp(i)

i=i+1

loop

close #1

open ws+"max.dat" for input as 3

input #3, max

close #3

open ws+"aktKapm.dat" for output as 2

menu:

ActionButton = MbeOpenModalDialog(1)

'CANCEL

if ActionButton = 4 then

MbeWritePrompt "Αντε Γειά - omoe.bas"

end

end if

'WRITE FM.GEN

if actionbutton = 3 then

if ve=50 then cm=1

if ve=60 then cm=2

if ve=70 then cm=3

if ve=80 then cm=4

if ve=90 then cm=5

if ve=100 then cm=6

if ve=110 then cm=7

if ve=120 then cm=8

if rc="AI" then cn=1

if rc="AII" then cn=2

if rc="AIII" then cn=3

if rc="AIV" then cn=4

if rc="BII" then cn=5

if rc="BIII" then cn=6

if rc="BIV" then cn=7

if qi="with" then qa=1

if qi="without" then qa=2

if cm=1 then ve=50

if cm=2 then ve=60

if cm=3 then ve=70

if cm=4 then ve=80

if cm=5 then ve=90

if cm=6 then ve=100

if cm=7 then ve=110

if cm=8 then ve=120

if cn=1 then rc="AI"

if cn=2 then rc="AII"

if cn=3 then rc="AIII"

if cn=4 then rc="AIV"

if cn=5 then rc="BII"

if cn=6 then rc="BIII"

if cn=7 then rc="BIV"

if qa=1 then qi="with"

if qa=2 then qi="without"

for i=2 to max

if (cn=1 or cn=2 or cn=3 or cn=4) then

if (cm=8 and tp(i)=2 and qa=1) then rm(i)=650

if (cm=7 and tp(i)=2 and qa=1) then rm(i)=530

if (cm=6 and tp(i)=2 and qa=1) then rm(i)=420

if (cm=5 and tp(i)=2 and qa=1) then rm(i)=330

if (cm=4 and tp(i)=2 and qa=1) then rm(i)=250

if (cm=3 and tp(i)=2 and qa=1) then rm(i)=180

if (cm=2 and tp(i)=2 and qa=1) then rm(i)=125

if (cm=1 and tp(i)=2 and qa=1) then rm(i)=80

end if

if (cn=1 or cn=2 or cn=3 or cn=4) then

if (cm=8 and tp(i)=2 and qa=2) then rm(i)=2470

if (cm=7 and tp(i)=2 and qa=2) then rm(i)=2020

if (cm=6 and tp(i)=2 and qa=2) then rm(i)=1620

if (cm=5 and tp(i)=2 and qa=2) then rm(i)=1260

if (cm=4 and tp(i)=2 and qa=2) then rm(i)=960

if (cm=3 and tp(i)=2 and qa=2) then rm(i)=700

if (cm=2 and tp(i)=2 and qa=2) then rm(i)=490

if (cm=1 and tp(i)=2 and qa=2) then rm(i)=325

end if

```
if (cn=1 or cn=2 or cn=3 or cn=4) then
 if (cm=8 and tp(i)=1 and qa=1) then rm(i)=740
 if (cm=7 and tp(i)=1 and qa=1) then rm(i)=600
 if (cm=6 and tp(i)=1 and qa=1) then rm(i)=480
 if (cm=5 and tp(i)=1 and qa=1) then rm(i)=370
 if (cm=4 and tp(i)=1 and qa=1) then rm(i)=280
 if (cm=3 and tp(i)=1 and qa=1) then rm(i)=200
 if (cm=2 and tp(i)=1 and qa=1) then rm(i)=140
 if (cm=1 and tp(i)=1 and qa=1) then rm(i)=95
end if
```

```
if (cn=1 or cn=2 or cn=3 or cn=4) then
 if (cm=8 and tp(i)=1 and qa=2) then rm(i)=2470
 if (cm=7 and tp(i)=1 and qa=2) then rm(i)=2020
 if (cm=6 and tp(i)=1 and qa=2) then rm(i)=1620
 if (cm=5 and tp(i)=1 and qa=2) then rm(i)=1260
 if (cm=4 and tp(i)=1 and qa=2) then rm(i)=960
 if (cm=3 and tp(i)=1 and qa=2) then rm(i)=700
 if (cm=2 and tp(i)=1 and qa=2) then rm(i)=490
 if (cm=1 and tp(i)=1 and qa=2) then rm(i)=325
end if
```

```
if (cn=5 or cn=6 or cn=7) then
 if (cm=5 and qa=1) then rm(i)=70
 if (cm=4 and qa=1) then rm(i)=110
 if (cm=3 and qa=1) then rm(i)=160
 if (cm=2 and qa=1) then rm(i)=220
 if (cm=1 and qa=1) then rm(i)=300
end if
```



```
if (cn=5 or cn=6 or cn=7) then
 if (cm=5 and qa=2) then rm(i)=70
 if (cm=4 and qa=2) then rm(i)=110
 if (cm=3 and qa=2) then rm(i)=160
 if (cm=2 and qa=2) then rm(i)=220
 if (cm=1 and qa=2) then rm(i)=300
end if
```

```
if (cn=5 or cn=6 or cn=7) then
 if cm>5 then
 mbemessagebox "AKYROS SYNDYASMOS TAXYTHTAS KAI
KATHGORIAS DROMOU"
 goto menu
 end if
end if
```

```
if rm(i)>curv(i) then
 mbemessagebox "Aktina Kampylotitas katw tou elaxistou sto simeio ("
+ cstr(XL(i))+ "," +cstr(YL(i))+")"
 end
end if
```

```
write #2, cm,cn,qa,tp(i),curv(i),rm(i)
next i
close #2
end if
mbemessagebox "OLA OK"
end sub
```

## omoeorat11.bas

---

'-----

'OMOE - omoe.bas - 8/12/2002

'-----

Const CrLf = Chr\$(13) + Chr\$(10)

sub main

'GENERAL

dim ws as string           ' Working Subdirectory

dim ve as double           ' Speed

menu:

ws= "c:\aggelos\"

if ve=50 then cm=1

if ve=60 then cm=2

if ve=70 then cm=3

if ve=80 then cm=4

if ve=90 then cm=5

if ve=100 then cm=6

if ve=110 then cm=7

if ve=120 then cm=8

if ve=130 then cm=9

ActionButton = MbeOpenModalDialog(1)

'CANCEL

```
if ActionButton = 4 then
MbeWritePrompt "ΑΥΤΕ ΓΕΙΛΑ - omoe.bas"
end
end if
```

```
'WRITE FM.GEN
```

```
if actionbutton = 3 then
```

```
'CHECKING SPEED & ROAD_CATEGORY
```

```
if cm=1 then ve=50
if cm=2 then ve=60
if cm=3 then ve=70
if cm=4 then ve=80
if cm=5 then ve=90
if cm=6 then ve=100
if cm=7 then ve=110
if cm=8 then ve=120
if cm=9 then ve=130
```

```
end if
```

```
if ve=50 then Su=475
if ve=60 then Su=475
if ve=70 then Su=500
if ve=80 then Su=525
if ve=90 then Su=575
if ve=100 then Su=625
if ve=110 then Su=675
if ve=120 then Su=675
```

if ve=130 then Su=675

if ve=50 then Ss=58

if ve=60 then Ss=80

if ve=70 then Ss=108

if ve=80 then Ss=132

if ve=90 then Ss=117

if ve=100 then Ss=210

if ve=110 then Ss=250

if ve=120 then Ss=303

if ve=130 then Ss=340

print ve,su,ss

open ws+"apaitorat.dat" for output as 1

write #1, ve,Su,Ss

close #1

End Sub

---

sub main

dim ds as double

dim a1(1000) as double

dim a3(1000) as double

dim ws as string

dim max as double

dim xtp(1000) as double

dim ytp(1000) as double

dim xtk(1000) as double

dim ytk(1000) as double

dim DIST as double

dim x(1000) as double

dim y(1000) as double

dim Orat(1000) as double

dim XB(1000) as double

dim XK(1000) as double

dim YB(1000) as double

dim YK(1000) as double

dim YKO(1000) as double

dim XKO(1000) as double

dim YBO(1000) as double

dim XBO(1000) as double

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 7"

MbeSendCommand "LOCK LEVEL TOGGLE "

**MbeSendCommand "ACTIVE LEVEL 7"**

**MbeSendKeyIn "LV=7"**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**open ws+"oratotitaPrA-D.dat" for output as 8**

**open ws+"max.dat" for input as 3**

**input #3, max**

**close #3**

**m=1**

**open ws+"oriop.dat" for input as 4**

**do while not eof (4)**

**input #4,XB(m),YB(m)**

**m=m+1**

**loop**

**close #4**

**m=1**

**open ws+"oriok.dat" for input as 5**

**do while not eof (5)**

**input #5,XK(m),YK(m)**

**m=m+1**

**loop**

**close #5**

**i=1**

open ws+"blemak.dat" for input as 7

do while not eof (7)

input #7,XKO(i),YKO(i)

i=i+1

loop

close #7

j=1

open ws+"blemap.dat" for input as 6

do while not eof (6)

input #6,XBO(j),YBO(j)

j=j+1

loop

close #6

'YPOLOGISMOS TOMWN GIA PROSPERASH

n=50

for i=1 to max-1

for j=i+1 to max-1

x(1)=XKO(i)

y(1)=YKO(i)

x(n)=XBO(j)

y(n)=YBO(j)

for k=1 to n

'diairesh ths eu8eias blemmatos odhgou se n tmhmata

'apo katw lwrida kineitai to oxhma kai koitaei panw

$x(k+1)=x(k)+(XBO(j)-XKO(i))/n$

$y(k+1)=y(k)+(YBO(j)-YKO(i))/n$

for m=i to max-1

if  $x(k+1)>XB(m)$  and  $XB(m+1)>x(k+1)$  then goto DOULEIA

'if  $x(k+1)<XB(m)$  then goto HERE

next m

DOULEIA :

$y1=YB(m)$

$y2=YB(m+1)$

$x1=XB(m)$

$x2=XB(m+1)$

'tomh panw oriou me katakoryfh

$y3=1$

$y4=2$

$x3=x(k+1)$

$x4=x(k+1)$

if  $x1=x2$  then  $x1=x1+0.001$

$a1(m)=(y1-y2)/(x1-x2)$



$x_{tp}(m) = x_3$

$y_{tp}(m) = y_1 + a_1(m) \cdot (x_3 - x_1)$

'katw orio

$y_5 = YK(m)$

$y_6 = YK(m+1)$

$x_5 = XK(m)$

$x_6 = Xk(m+1)$

if  $x_5 = x_6$  then  $x_5 = x_5 + 0.001$

$a_3(m) = (y_5 - y_6) / (x_5 - x_6)$

$x_{tk}(m) = x_3$

$y_{tk}(m) = y_5 + a_3(m) \cdot (x_3 - x_5)$

if  $y_{tp}(m) < y(k+1)$  or  $y(k+1) < y_{tk}(m)$  then

mbesendreset

Dim startPoint As MbePoint

Dim point As MbePoint, point2 As MbePoint

' Start a command

MbeSendCommand "MEASURE DISTANCE ICON "

' Coordinates are in master units

startPoint.x = XKO(i)

startPoint.y = YKO(i)

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = (XBO(j-1)+XBO(j))/2**

**point.y = (YBO(j-1)+YBO(j))/2**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**DIST = MbeState.measureResult1**

**mbesendreset**

**Orat(i)=DIST/10000**

**print i,orat(i)**

**write #8, i,Orat(i)**

**mbesendreset**

**goto EDW**

**end if**

**HERE :**

**next k**

**next j**

**EDW :**

**next i**

**close #8**

**mbesendreset**

**end sub**

sub main

dim ds as double

dim a1(1000) as double

dim a3(1000) as double

dim ws as string

dim max as double

dim xtp(1000) as double

dim ytp(1000) as double

dim xtk(1000) as double

dim ytk(1000) as double

dim DIST as double

dim x(1000) as double

dim y(1000) as double

dim Orat(1000) as double

dim XB(1000) as double

dim XK(1000) as double

dim YB(1000) as double

dim YK(1000) as double

dim YKO(1000) as double

dim XKO(1000) as double

dim YBO(1000) as double

dim XBO(1000) as double

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 7"

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 7"**

**MbeSendKeyIn "LV=7"**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**open ws+"oratotitaPrd-a.dat" for output as 8**

**open ws+"max.dat" for input as 3**

**input #3, max**

**close #3**

**m=1**

**open ws+"oriop.dat" for input as 4**

**do while not eof (4)**

**input #4,XB(m),YB(m)**

**m=m+1**

**loop**

**close #4**

**m=1**

**open ws+"oriok.dat" for input as 5**

**do while not eof (5)**

**input #5,XK(m),YK(m)**

**m=m+1**

**loop**

**close #5**

```
i=1
open ws+"blemak.dat" for input as 7
do while not eof (7)
input #7,XKO(i),YKO(i)
i=i+1
loop
close #7
```

```
j=1
open ws+"blemap.dat" for input as 6
do while not eof (6)
input #6,XBO(j),YBO(j)
j=j+1
loop
close #6
```

**'YPOLOGISMOS TOMWN GIA PROSPERASH**

```
n=50
for i=max-1 to 2 step-1
for j=i-1 to 1 step-1
```

```
x(1)=XBO(i)
y(1)=YBO(i)
x(n)=XKO(j)
y(n)=YKO(j)
```

```
for k=1 to n-1
```

'diaresh ths eu8eias blemmatos odhgou se n tmhmata

'apo katw lwrída kineítai to oxhma kai koítai panw

$x(k+1)=x(k)-(XBO(i)-XKO(j))/n$

$y(k+1)=y(k)-(YBO(i)-YKO(j))/n$

for m=i to 2 step-1

if  $(x(k+1)>XB(m)$  and  $XB(m+1)>x(k+1))$  then goto DOULEIA

next m

DOULEIA :

'print  $xB(m),x(k),XB(m+1)$

'stop

'panw orio

'stop

$y1=YB(m)$

$y2=YB(m+1)$

$x1=XB(m)$

$x2=XB(m+1)$

'tomh panw oriou me katakoryfh

$y3=1$

$y4=2$

$x3=x(k+1)$

$x4=x(k+1)$

if  $x1=x2$  then  $x1=x1+0.001$

$a1(m)=(y1-y2)/(x1-x2)$

'print a1(m)

xtp(m)=x3

ytp(m)=y1+a1(m)\*(x3-x1)

'katw orio

y5=YK(m)

y6=YK(m+1)

x5=XK(m)

x6=Xk(m+1)

if x5=x6 then

x5=x5+0.001

end if

$a3(m)=(y5-y6)/(x5-x6)$

xtk(m)=x3

ytk(m)=y5+a3(m)\*(x3-x5)

if ytp(m)<y(k+1) or y(k+1)<ytk(m) then

mbesendreset

Dim startPoint As MbePoint

Dim point As MbePoint, point2 As MbePoint

' Start a command

MbeSendCommand "MEASURE DISTANCE ICON "

' Coordinates are in master units

startPoint.x = XBO(i)

startPoint.y = YBO(i)

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = (XKO(j+1)+XKO(j))/2

point.y = (YKO(j+1)+YKO(j))/2

point.z = startPoint.z

MbeSendDataPoint point, 1%

DIST = MbeState.measureResult1

mbesendreset

Orat(i)=DIST/10000

print i,orat(i)

write #8, i,Orat(i)

mbesendreset

goto EDW

end if



HERE :

next k

next j

EDW :

next i

close #8

mbesendreset

end sub

---

sub main

dim ds as double

dim a1(1000) as double

dim a3(1000) as double

dim ws as string

dim max as double

dim xtp(1000) as double

dim ytp(1000) as double

dim xtk(1000) as double

dim ytk(1000) as double

dim DIST as double

dim x(1000) as double

dim y(1000) as double

dim Orat(1000) as double

dim XB(1000) as double

dim XK(1000) as double

dim YB(1000) as double

dim YK(1000) as double

dim YKO(1000) as double

dim XKO(1000) as double

dim YBO(1000) as double

dim XBO(1000) as double

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 7"

MbeSendCommand "LOCK LEVEL TOGGLE "

**MbeSendCommand "ACTIVE LEVEL 7"**

**MbeSendKeyIn "LV=7"**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**open ws+"oratotitaSTA-d.dat" for output as 8**

**open ws+"max.dat" for input as 3**

**input #3, max**

**close #3**

**m=1**

**open ws+"oriop.dat" for input as 4**

**do while not eof (4)**

**input #4,XB(m),YB(m)**

**m=m+1**

**loop**

**close #4**

**m=1**

**open ws+"oriok.dat" for input as 5**

**do while not eof (5)**

**input #5,XK(m),YK(m)**

**m=m+1**

**loop**

**close #5**

**i=1**

open ws+"blemak.dat" for input as 7

do while not eof (7)

input #7,XKO(i),YKO(i)

i=i+1

loop

close #7

j=1

open ws+"blemap2.dat" for input as 6

do while not eof (6)

input #6,XBO(j),YBO(j)

j=j+1

loop

close #6

'YPOLOGISMOS TOMWN GIA PROSPERASH

n=50

for i=1 to max-2

for j=i+1 to max-1

x(1)=XKO(i)

y(1)=YKO(i)

x(n)=XBO(j)

y(n)=YBO(j)

for k=1 to n-1

'diaresh ths eu8eias blemmatos odhgou se n tmhmata

'apo katw lwrda kineitai to oxhma kai koitaei panw

$x(k+1)=x(k)+(XBO(j)-XKO(i))/n$

$y(k+1)=y(k)+(YBO(j)-YKO(i))/n$

for m=i to max-1

if  $x(k+1)>XB(m)$  and  $XB(m+1)>x(k+1)$  then goto DOULEIA

next m

DOULEIA :

$y1=YB(m)$

$y2=YB(m+1)$

$x1=XB(m)$

$x2=XB(m+1)$

'tomh panw oriou me katakoryfh

$y3=1$

$y4=2$

$x3=x(k+1)$

$x4=x(k+1)$

if  $x1=x2$  then  $x1=x1+0.001$

$a1(m)=(y1-y2)/(x1-x2)$

'print a1(m)

$xtp(m)=x3$

$ytp(m)=y1+a1(m)*(x3-x1)$

'katw orio

**y5=YK(m)**

**y6=YK(m+1)**

**x5=XK(m)**

**x6=Xk(m+1)**

**if x5=x6 then x5=x5+0.001**

**a3(m)=(y5-y6)/(x5-x6)**

**xtk(m)=x3**

**ytk(m)=y5+a3(m)\*(x3-x5)**

**if ytp(m)<y(k+1) or y(k+1)<ytk(m) then**

**mbesendreset**

**Dim startPoint As MbePoint**

**Dim point As MbePoint, point2 As MbePoint**

**' Start a command**

**MbeSendCommand "MEASURE DISTANCE ICON "**

**' Coordinates are in master units**

**startPoint.x = XKO(i)**

**startPoint.y = YKO(i)**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

```
point.x = (XBO(j-1)+XBO(j))/2
point.y = (YBO(j-1)+YBO(j))/2
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
DIST = MbeState.measureResult1
mbesendreset
Orat(i)=DIST/10000
print i,orat(i)
write #8, i,Orat(i)
```

```
mbesendreset
goto EDW
end if
```

HERE :

```
next k
next j
```

```
EDW :
next i
close #8
mbesendreset
end sub
```

---

sub main

dim ds as double

dim a1(1000) as double

dim a3(1000) as double

dim ws as string

dim max as double

dim xtp(1000) as double

dim ytp(1000) as double

dim xtk(1000) as double

dim ytk(1000) as double

dim DIST as double

dim x(1000) as double

dim y(1000) as double

dim Orat(1000) as double

dim XB(1000) as double

dim XK(1000) as double

dim YB(1000) as double

dim YK(1000) as double

dim YKO(1000) as double

dim XKO(1000) as double

dim YBO(1000) as double

dim XBO(1000) as double

ws= "c:\aggelos\"

MbeSendCommand "ACTIVE LEVEL 7"



**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 7"**

**MbeSendKeyIn "LV=7"**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**open ws+"oratotitaStrd-a.dat" for output as 8**

**open ws+"max.dat" for input as 3**

**input #3, max**

**close #3**

**m=1**

**open ws+"oriop.dat" for input as 4**

**do while not eof (4)**

**input #4,XB(m),YB(m)**

**m=m+1**

**loop**

**close #4**

**m=1**

**open ws+"oriok.dat" for input as 5**

**do while not eof (5)**

**input #5,XK(m),YK(m)**

**m=m+1**

**loop**

**close #5**

i=1

open ws+"blemak2.dat" for input as 7

do while not eof (7)

input #7,XKO(i),YKO(i)

i=i+1

loop

close #7

j=1

open ws+"blemap.dat" for input as 6

do while not eof (6)

input #6,XBO(j),YBO(j)

j=j+1

loop

close #6

'YPOLOGISMOS TOMWN GIA PROSPERASH

n=50

for i=max-1 to 2 step-1

for j=i-1 to 1 step-1

x(1)=XBO(i)

y(1)=YBO(i)

x(n)=XKO(j)

y(n)=YKO(j)

for k=1 to n-1

'diairesh ths eu8eias blemmatos odhgou se n tmhmata

'apo katw lwrída kineítai to oxhma kai koítai panw

$x(k+1)=x(k)-(XBO(i)-XKO(j))/n$

$y(k+1)=y(k)-(YBO(i)-YKO(j))/n$

for m=i to 1 step-1

if  $(x(k+1)>XB(m)$  and  $XB(m+1)>x(k+1))$  then goto DOULEIA

next m

DOULEIA :

$y1=YB(m)$

$y2=YB(m+1)$

$x1=XB(m)$

$x2=XB(m+1)$

'tomh panw oriou me katakoryfh

$y3=1$

$y4=2$

$x3=x(k+1)$

$x4=x(k+1)$

if  $x1=x2$  then  $x1=x1+0.001$

$a1(m)=(y1-y2)/(x1-x2)$

'print a1(m)

$xtp(m)=x3$

$y_{tp}(m) = y_1 + a_1(m) \cdot (x_3 - x_1)$

'katw orio

$y_5 = YK(m)$

$y_6 = YK(m+1)$

$x_5 = XK(m)$

$x_6 = Xk(m+1)$

if  $x_6 = x_5$  then  $a_3(m) = (y_5 - y_6) / (x_5 - x_6 + 0.0001)$

if  $x_6 \neq x_5$  then  $a_3(m) = (y_5 - y_6) / (x_5 - x_6)$

$x_{tk}(m) = x_3$

$y_{tk}(m) = y_5 + a_3(m) \cdot (x_3 - x_5)$

if  $y_{tp}(m) < y(k+1)$  or  $y(k+1) < y_{tk}(m)$  then

mbesendreset

Dim startPoint As MbePoint

Dim point As MbePoint, point2 As MbePoint

' Start a command

MbeSendCommand "MEASURE DISTANCE ICON "

' Coordinates are in master units

startPoint.x = XBO(i)

startPoint.y = YBO(i)

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

```
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = (XKO(j+1)+XKO(j))/2
point.y = (YKO(j+1)+YKO(j))/2
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
DIST = MbeState.measureResult1
mbesendreset
Orat(i)=DIST/5000
print i,orat(i)
write #8, i,Orat(i)
mbesendreset
```

```
goto EDW
end if
```

```
HERE :
next k
next j
```

```
EDW :
next i
close #8
mbesendreset
end sub
```

---

**sub main**

**dim XL(1000) as double**

**dim YL(1000) as double**

**dim ds as double**

**dim YDL(1000) as double**

**dim YDDL(1000) as double**

**dim CURV(1000) as double**

**dim ws as string**

**dim max as double**

**dim X8(1000) as double**

**dim DIST as double**

**dim XLSTEP as double**

**dim ORAT(1000) as double**

**dim mfin as double**

**dim XU(1000) as double**

**dim SA(1000) as double**

**ws= "c:\aggelos\"**

**MbeSendCommand "ACTIVE LEVEL 10"**

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 10"**

**i=1**

```
open ws+"sp3.dat" for input as 1
do while not eof(1)
input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)
i=i+1
loop
close #1
```

```
open ws+"XLSTEP.dat" for input as 2
input #2, XLSTEP, ds
close #2
```

```
open ws+"max.dat" for input as 5
input #5, max
close #5
```

```
open ws+"oratotitaPrA-D.dat" for input as 6
do while not eof (6)
input #6, i,ORAT(i)
m=Orat(i)*10
print m
SA(i)=m/10
```

```
i=i+1
loop
mfin=i
close #6
```

```
X8(1)=0
```

for i=1 to max

Dim startPoint As MbePoint

Dim point As MbePoint, point2 As MbePoint

' Start a command

MbeSendCommand "MEASURE DISTANCE ICON "

' Coordinates are in master units

startPoint.x = XL(i)

startPoint.y = YL(i)

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = XL(i+1)

point.y = YL(i+1)

point.z = startPoint.z

MbeSendDataPoint point, 1%

DIST = MbeState.measureResult1

if i>1 then

X8(i)=X8(i-1)+DIST/10000

end if

'print i,X8(i)



k=x8(i)

XU(i)=k

next i

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE SMARTLINE "

' Coordinates are in master units

startPoint.x = 0.000000#

startPoint.y = 0.000000#

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y-5

point.z = startPoint.z

MbeSendDataPoint point, 1%

```
point.x = 0
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

' Start a command

```
MbeSendCommand "PLACE SMARTLINE "
```

' Coordinates are in master units

```
startPoint.x = 0.000000#
startPoint.y = -10
startPoint.z = 0.000000#
```

' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(max-2)
point.y = startPoint.y
```

```
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(max-2)
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

' Start a command

```
MbeSendCommand "PLACE SMARTLINE "
```

' Coordinates are in master units

```
startPoint.x = 0.000000#
startPoint.y = 0.000000#
startPoint.z = 0.000000#
```

- ' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y-15

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = 0

point.y = startPoint.y-15

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = 0

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

mbesendreset

- ' Start a command

MbeSendCommand "WINDOW AREA EXTENDED 1"

- ' Coordinates are in master units

startPoint.x = 0.140800#

startPoint.y = 1.697900#

startPoint.z = 0.000000#

- ' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = startPoint.x - 2.780100#

point.y = startPoint.y - 19.975000#

point.z = startPoint.z

MbeSendDataPoint point, 1%

- ' Send a reset to the current command

MbeSendReset

MbeSendCommand "PLACE LINE CONSTRAINED "

- ' Send a tentative point

point.x = startPoint.x + 0.134200#

point.y = startPoint.y - 1.601000#

point.z = startPoint.z

MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.140800#

point.y = startPoint.y - 1.697900#

point.z = startPoint.z

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.091600#**

**point.y = startPoint.y - 6.567200#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 6.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.204500#**

**point.y = startPoint.y - 11.561700#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.232800#**

**point.y = startPoint.y - 11.787500#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 11.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.176300#**

**point.y = startPoint.y - 16.443300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

point.x = startPoint.x - 0.140800#  
point.y = startPoint.y - 16.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

MbeSendReset

MbeSendCommand "ZOOM OUT EXTENDED 1"

MbeSendCommand "PLACE LINE CONSTRAINED "

MbeSendCommand "IGEN CONSTRAIN LENGTH UNLOCK "

MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "

MbeSendReset

point.x = startPoint.x - 19.964100#  
point.y = startPoint.y - 1.911300#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.140800#  
point.y = startPoint.y - 1.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 20.020500#  
point.y = startPoint.y - 15.907100#  
point.z = startPoint.z

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.738300#**

**point.y = startPoint.y - 16.415000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

- ' **Send a message string to an application**

- ' **Content is defined by the application**

**MbeSendAppMessage "TEXTEDIT", "FirstLine AYKSON ARI8MOS"**

**MbeSendCommand "ACTIVE ANGLE 0.0000°"**

**MbeSendCommand "ACTIVE ANGLE "**

**point.x = startPoint.x - 10.818200#**

**point.y = startPoint.y - 4.112200#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine XIL. 8ESH"**



**point.x = startPoint.x - 13.358800#**

**point.y = startPoint.y - 9.134900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine MHKOS ORATOTHTAS"**

**point.x = startPoint.x - 10.931100#**

**point.y = startPoint.y - 14.270500#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a keyin that can be a command string

**MbeSendKeyin Chr\$(27)**

## **MBESENDRESET**

- ' Start a command

**MbeSendCommand "PLACE SMARTLINE "**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

- ' Send a tentative point

- ' Coordinates are in master units

**MbeSendTentPoint -0.311400#, -0.091100#, 0.000000#, 1%**

- ' Coordinates are in master units

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

- ' **Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.790000#**

**point.y = startPoint.y - 0.261900#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' **Send a reset to the current command**

**MbeSendReset**

**MbeSendCommand "IGEN CONSTRAIN ANGLE LOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

**point.x = startPoint.x - 0.396800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 5.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**point.x = startPoint.x - 0.055100#**

**point.y = startPoint.y - 10.083000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 10.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**

**point.y = startPoint.y - 9.826800#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.875400#**

**point.y = startPoint.y - 9.997600#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**

point.y = startPoint.y - 10.083000#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.131700#  
point.y = startPoint.y - 9.997600#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.046300#  
point.y = startPoint.y - 9.826800#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.000000#  
point.y = startPoint.y - 7.500000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

MbeSendReset

point.x = startPoint.x - 0.311400#  
point.y = startPoint.y - 15.036200#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x  
point.y = startPoint.y - 15.000000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

```
point.x = startPoint.x - 19.448200#
point.y = startPoint.y - 14.950800#
point.z = startPoint.z
MbeSendTentPoint point, 1%
```

```
point.x = startPoint.x - 20.000000#
point.y = startPoint.y - 15.000000#
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

**mbesendreset**

```
MbeSendKeyIn "LC=0"
MbeSendKeyIn "CO=2"
MbeSendKeyIn "WT=0"
```

**mbesendreset**

**for i=1 to max-2**

**' Start a command**

```
MbeSendCommand "PLACE LINE CONSTRAINED "
```

**' Coordinates are in master units**

```
startPoint.x = X8(i)
```

```
startPoint.y = 0
```

```
startPoint.z = 0.000000#
```

**' Send a data point to the current command**

```
point.x = startPoint.x
```

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(i)**

**point.y = ORAT(i)/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**next i**

**for i=2 to mfin-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(XU(i))**

**MbeSendDataPoint X8(i),-7.5,0,1%**

**mbesendreset**

next i

for i=2 to mfin-1

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE DIALOGTEXT ICON "

MbeSendCommand "ACTIVE ANGLE "

MbeSendCommand "ACTIVE ANGLE 0°"

MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(i-1)

MbeSendDataPoint X8(i),-2.5,0,1%

mbesendreset

next i

for i=2 to mfin-1

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE DIALOGTEXT ICON "

MbeSendCommand "ACTIVE ANGLE "

MbeSendCommand "ACTIVE ANGLE 0°"

MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(SA(i))

MbeSendDataPoint X8(i),-12.5,0,1%

mbesendreset



next i

MbeSendCommand "PLACE DIALOGTEXT ICON "

MbeSendCommand "ACTIVE ANGLE "

MbeSendCommand "ACTIVE ANGLE 0°"

MbeSendAppMessage "TEXTEDIT","FirstLine "+"DIAGRAMMA ORATOTHTAS  
GIA PROSPERASH SE POREIA A-D"

MbeSendDataPoint 0,-16,0,1%

open ws+"apaitorat.dat" for input as 10

input #10, ve,Su,Ss

close #10

MbeSendKeyIn "CO=3"

'Start a command

MbeSendCommand "PLACE LINE CONSTRAINED "

' Coordinates are in master units

startPoint.x = 0

startPoint.y = Su/10

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

**MbeSendDataPoint point, 2%**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(mfin)**

**point.y = Su/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**mbesendreset**

**End Sub**

---

## **diagrammaprosD-A.bas**

---

**sub main**

**dim XL(1000) as double**

**dim YL(1000) as double**

**dim ds as double**

**dim YDL(1000) as double**

**dim YDDL(1000) as double**

**dim CURV(1000) as double**

**dim ws as string**

**dim max as double**

**dim X8(1000) as double**

**dim DIST as double**

**dim XLSTEP as double**

**dim ORAT(1000) as double**

**dim mfin as double**

**dim XU(1000) as double**

**dim SA(1000) as double**

**dim a(1000) as double**

**ws= "c:\aggelos\"**

**MbeSendCommand "ACTIVE LEVEL 10"**

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 10"**

i=1

open ws+"sp3.dat" for input as 1

do while not eof(1)

input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)

i=i+1

loop

close #1

open ws+"XLSTEP.dat" for input as 2

input #2, XLSTEP, ds

close #2

open ws+"max.dat" for input as 5

input #5, max

close #5

X8(max)=0

for i=max to 1 step-1

Dim startPoint As MbePoint

Dim point As MbePoint, point2 As MbePoint

' Start a command

MbeSendCommand "MEASURE DISTANCE ICON "

' Coordinates are in master units

startPoint.x = XL(i)

startPoint.y = YL(i)

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = XL(i-1)

point.y = YL(i-1)

point.z = startPoint.z

MbeSendDataPoint point, 1%

DIST = MbeState.measureResult1

if i=max then

goto HERE

end if

X8(i)=X8(i+1)+DIST/10000

'print i,X8(i)

k=x8(i)

XU(i)=k

HERE :

next i

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE SMARTLINE "

- ' Coordinates are in master units

startPoint.x = 0.000000#

startPoint.y = 0.000000#

startPoint.z = 0.000000#

- ' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(1)

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(1)

point.y = startPoint.y-5

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = 0

point.y = startPoint.y-5

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = 0

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**' Start a command**

**MbeSendCommand "PLACE SMARTLINE "**

**' Coordinates are in master units**

**startPoint.x = 0.000000#**

**startPoint.y = -10**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(1)**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(1)**

**point.y = startPoint.y-5**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

```
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

' Start a command

```
MbeSendCommand "PLACE SMARTLINE "
```

' Coordinates are in master units

```
startPoint.x = 0.000000#
startPoint.y = 0.000000#
startPoint.z = 0.000000#
```

' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(1)
point.y = startPoint.y
```



```
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(1)
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
mbeendreset
```

- ' Start a command

```
MbeSendCommand "WINDOW AREA EXTENDED 1"
```

- ' Coordinates are in master units

```
startPoint.x = 0.140800#
startPoint.y = 1.697900#
startPoint.z = 0.000000#
```

- ' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
```

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 2.780100#**

**point.y = startPoint.y - 19.975000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a reset to the current command

**MbeSendReset**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

- ' Send a tentative point

**point.x = startPoint.x + 0.134200#**

**point.y = startPoint.y - 1.601000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 1.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.091600#**

**point.y = startPoint.y - 6.567200#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 6.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.204500#**

**point.y = startPoint.y - 11.561700#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.232800#**

**point.y = startPoint.y - 11.787500#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 11.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.176300#**

**point.y = startPoint.y - 16.443300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "ZOOM OUT EXTENDED 1"**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN LENGTH UNLOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**MbeSendReset**

**point.x = startPoint.x - 19.964100#**

**point.y = startPoint.y - 1.911300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 1.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 20.020500#**

**point.y = startPoint.y - 15.907100#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.738300#**

**point.y = startPoint.y - 16.415000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

- ' Send a message string to an application
- ' Content is defined by the application

**MbeSendAppMessage "TEXTEDIT", "FirstLine AYKSON ARI8MOS"**

**MbeSendCommand "ACTIVE ANGLE 0.0000°"**

**MbeSendCommand "ACTIVE ANGLE "**

**point.x = startPoint.x - 10.818200#**

**point.y = startPoint.y - 4.112200#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine XIL. 8ESH"**

**point.x = startPoint.x - 13.358800#**

**point.y = startPoint.y - 9.134900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine MHKOS ORATOTHTAS"**

**point.x = startPoint.x - 10.931100#**

**point.y = startPoint.y - 14.270500#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a keyin that can be a command string

**MbeSendKeyin Chr\$(27)**

## **MBESENDRESET**

- ' Start a command

**MbeSendCommand "PLACE SMARTLINE "**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

- ' Send a tentative point

- ' Coordinates are in master units

**MbeSendTentPoint -0.311400#, -0.091100#, 0.000000#, 1%**

- ' Coordinates are in master units

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

- ' Send a data point to the current command

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.790000#**

**point.y = startPoint.y - 0.261900#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**' Send a reset to the current command**

**MbeSendReset**

**MbeSendCommand "IGEN CONSTRAIN ANGLE LOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

**point.x = startPoint.x - 0.396800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 5.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**  
**point.y = startPoint.y - 5.044400#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**  
**point.y = startPoint.y - 5.044400#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**  
**point.y = startPoint.y - 5.044400#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**  
**point.y = startPoint.y - 4.959000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**  
**point.y = startPoint.y - 4.959000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**  
**point.y = startPoint.y - 4.959000#**  
**point.z = startPoint.z**  
**MbeSendDataPoint point, 1%**



**MbeSendReset**

**point.x = startPoint.x - 0.055100#**  
**point.y = startPoint.y - 10.083000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**  
**point.y = startPoint.y - 10.000000#**  
**point.z = startPoint.z**  
**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**  
**point.y = startPoint.y - 9.826800#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.875400#**  
**point.y = startPoint.y - 9.997600#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**  
**point.y = startPoint.y - 10.083000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**  
**point.y = startPoint.y - 9.997600#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

point.x = startPoint.x - 20.046300#  
point.y = startPoint.y - 9.826800#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.000000#  
point.y = startPoint.y - 7.500000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

MbeSendReset

point.x = startPoint.x - 0.311400#  
point.y = startPoint.y - 15.036200#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x  
point.y = startPoint.y - 15.000000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 19.448200#  
point.y = startPoint.y - 14.950800#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.000000#  
point.y = startPoint.y - 15.000000#  
point.z = startPoint.z

**MbeSendDataPoint point, 1%**

**mbesendreset**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**mbesendreset**

**open ws+"apaitorat.dat" for input as 10**

**input #10, ve,Su,Ss**

**close #10**

**i=1**

**open ws+"oratotitaPrd-a.dat" for input as 6**

**do while not eof (6)**

**input #6, a(i),ORAT(i)**

**'print orat(i)**

**m=Orat(i)\*10**

**SA(i)=m/10**

**i=i+1**

**loop**

**mfin=i**

**close #6**

**MbeSendKeyIn "CO=2"**

**for i=max-1 to (max-mfin+1) step-1**

**' Start a command**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**' Coordinates are in master units**

**startPoint.x = X8(i)**

**startPoint.y = 0**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(i)**

**point.y = SA(max-i)/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**next i**

**for i=max to (max-mfin+1) step-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(XU(i))**

**MbeSendDataPoint X8(i),-7.5,0,1%**

**mbesendreset**

**next i**

**for i=max-1 to (max-mfin+1) step-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(max-i)**

**MbeSendDataPoint X8(i),-2.5,0,1%**

**mbesendreset**

**next i**

**for i=max-1 to (max-mfin+1) step-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(SA(max-i))**

**MbeSendDataPoint X8(i),-12.5,0,1%**

**mbesendreset**

**next i**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine "+"DIAGRAMMA ORATOTHTAS  
GIA PROSPERASH SE POREIA D-A"**

**MbeSendDataPoint 0,-16,0,1%**

**MbeSendKeyIn "CO=3"**

**'Start a command**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**' Coordinates are in master units**

**startPoint.x = 0**

**startPoint.y = Su/10**

**startPoint.z = 0.000000#**

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 2%

MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "

point.x = X8(max-mfin+1)

point.y = Su/10

point.z = startPoint.z

MbeSendDataPoint point, 2%

mbesendreset

End Sub

---

**sub main**

**dim XL(1000) as double**

**dim YL(1000) as double**

**dim ds as double**

**dim YDL(1000) as double**

**dim YDDL(1000) as double**

**dim CURV(1000) as double**

**dim ws as string**

**dim max as double**

**dim X8(1000) as double**

**dim DIST as double**

**dim XLSTEP as double**

**dim ORAT(1000) as double**

**dim mfin as double**

**dim XU(1000) as double**

**dim SA(1000) as double**

**ws= "c:\aggelos\"**

**MbeSendCommand "ACTIVE LEVEL 10"**

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 10"**



```

i=1
open ws+"sp3.dat" for input as 1
do while not eof(1)
input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)
i=i+1
loop
close #1

open ws+"XLSTEP.dat" for input as 2
input #2, XLSTEP, ds
close #2

open ws+"max.dat" for input as 5
input #5, max
close #5

open ws+"oratotitaSTA-d.dat" for input as 6
do while not eof (6)
input #6, i,ORAT(i)
m=Orat(i)*10
SA(i)=m/10

i=i+1
loop
mfin=i
close #6

X8(1)=0
for i=1 to max

Dim startPoint As MbePoint

```

**Dim point As MbePoint, point2 As MbePoint**

**' Start a command**

**MbeSendCommand "MEASURE DISTANCE ICON "**

**' Coordinates are in master units**

**startPoint.x = XL(i)**

**startPoint.y = YL(i)**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = XL(i+1)**

**point.y = YL(i+1)**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**DIST = MbeState.measureResult1**

**if i>1 then**

**X8(i)=X8(i-1)+DIST/10000**

**end if**

**k=x8(i)**

**XU(i)=k**

next i

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE SMARTLINE "

' Coordinates are in master units

startPoint.x = 0.000000#

startPoint.y = 0.000000#

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = X8(max-2)

point.y = startPoint.y-5

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = 0

point.y = startPoint.y-5

point.z = startPoint.z

MbeSendDataPoint point, 1%

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**' Start a command**

**MbeSendCommand "PLACE SMARTLINE "**

**' Coordinates are in master units**

**startPoint.x = 0.000000#**

**startPoint.y = -10**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(max-2)**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(max-2)**

```
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

' Start a command

```
MbeSendCommand "PLACE SMARTLINE "
```

' Coordinates are in master units

```
startPoint.x = 0.000000#
startPoint.y = 0.000000#
startPoint.z = 0.000000#
```

' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
```

```
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(max-2)
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(max-2)
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
mbesendreset
```

- ' Start a command

```
MbeSendCommand "WINDOW AREA EXTENDED 1"
```

- ' Coordinates are in master units

```
startPoint.x = 0.140800#
```

```
startPoint.y = 1.697900#
```

**startPoint.z = 0.000000#**

- ' Send a data point to the current command

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 2.780100#**

**point.y = startPoint.y - 19.975000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a reset to the current command

**MbeSendReset**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

- ' Send a tentative point

**point.x = startPoint.x + 0.134200#**

**point.y = startPoint.y - 1.601000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 1.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.091600#**

**point.y = startPoint.y - 6.567200#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 6.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.204500#**

**point.y = startPoint.y - 11.561700#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.232800#**

**point.y = startPoint.y - 11.787500#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 11.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 0.176300#**

**point.y = startPoint.y - 16.443300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 0.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**



**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "ZOOM OUT EXTENDED 1"**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN LENGTH UNLOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**MbeSendReset**

**point.x = startPoint.x - 19.964100#**

**point.y = startPoint.y - 1.911300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 1.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 20.020500#**

**point.y = startPoint.y - 15.907100#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.738300#**

**point.y = startPoint.y - 16.415000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

- ' Send a message string to an application

- ' Content is defined by the application

**MbeSendAppMessage "TEXTEDIT", "FirstLine AYKSON ARI8MOS"**

**MbeSendCommand "ACTIVE ANGLE 0.0000°"**

**MbeSendCommand "ACTIVE ANGLE "**

**point.x = startPoint.x - 10.818200#**

**point.y = startPoint.y - 4.112200#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine XIL. 8ESH"**

**point.x = startPoint.x - 13.358800#**

**point.y = startPoint.y - 9.134900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendMessage "TEXTEDIT", "FirstLine MHKOS ORATOTHTAS"**

**point.x = startPoint.x - 10.931100#**

**point.y = startPoint.y - 14.270500#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a keyin that can be a command string

**MbeSendKeyin Chr\$(27)**

## **MBESENDRESET**

- ' Start a command

**MbeSendCommand "PLACE SMARTLINE "**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

- ' Send a tentative point
- ' Coordinates are in master units

**MbeSendTentPoint -0.311400#, -0.091100#, 0.000000#, 1%**

- ' Coordinates are in master units

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

- ' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

point.x = startPoint.x - 19.790000#

point.y = startPoint.y - 0.261900#

point.z = startPoint.z

MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.000000#

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 1%

- ' Send a reset to the current command

MbeSendReset

MbeSendCommand "IGEN CONSTRAIN ANGLE LOCK "

MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"

point.x = startPoint.x - 0.396800#

point.y = startPoint.y - 4.959000#

point.z = startPoint.z

MbeSendTentPoint point, 1%

point.x = startPoint.x

point.y = startPoint.y - 5.000000#

point.z = startPoint.z

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

point.x = startPoint.x - 19.960800#  
point.y = startPoint.y - 4.959000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

MbeSendReset

point.x = startPoint.x - 0.055100#  
point.y = startPoint.y - 10.083000#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x  
point.y = startPoint.y - 10.000000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 19.362800#  
point.y = startPoint.y - 9.826800#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 19.875400#  
point.y = startPoint.y - 9.997600#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 20.131700#  
point.y = startPoint.y - 10.083000#  
point.z = startPoint.z

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**

**point.y = startPoint.y - 9.997600#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**

**point.y = startPoint.y - 9.826800#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y - 7.500000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**point.x = startPoint.x - 0.311400#**

**point.y = startPoint.y - 15.036200#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 15.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.448200#**

**point.y = startPoint.y - 14.950800#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y - 15.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**mbesendreset**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**mbesendreset**

**for i=1 to max-2**

**' Start a command**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**' Coordinates are in master units**

**startPoint.x = X8(i)**

**startPoint.y = 0**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**



**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(i)**

**point.y = ORAT(i)/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**next i**

**for i=2 to mfin-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON " MbeSendCommand  
"ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(XU(i))**

**MbeSendDataPoint X8(i),-7.5,0,1%**

**mbesendreset**

**next i**

**for i=2 to mfin-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(i-1)**

**MbeSendDataPoint X8(i),-2.5,0,1%**

**mbesendreset**

**next i**

**for i=2 to mfin-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(SA(i))**

**MbeSendDataPoint X8(i),-12.5,0,1%**

**mbesendreset**

**next i**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine "+"DIAGRAMMA ORATOTHTAS  
GIA STASH SE POREIA A-D"**

**MbeSendDataPoint 0,-16,0,1%**

open ws+"apaitorat.dat" for input as 10

input #10, ve,Su,Ss

close #10

MbeSendKeyIn "CO=3"

'Start a command

MbeSendCommand "PLACE LINE CONSTRAINED "

' Coordinates are in master units

startPoint.x = 0

startPoint.y = Ss/10

startPoint.z = 0.000000#

' Send a data point to the current command

point.x = startPoint.x

point.y = startPoint.y

point.z = startPoint.z

MbeSendDataPoint point, 2%

MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "

point.x = X8(mfin)

point.y = Ss/10

point.z = startPoint.z

MbeSendDataPoint point, 2%

mbesendreset

End Sub

## **diagrammastashD-A.bas**

---

**sub main**

**dim XL(1000) as double**

**dim YL(1000) as double**

**dim ds as double**

**dim YDL(1000) as double**

**dim YDDL(1000) as double**

**dim CURV(1000) as double**

**dim ws as string**

**dim max as double**

**dim X8(1000) as double**

**dim DIST as double**

**dim XLSTEP as double**

**dim ORAT(1000) as double**

**dim mfin as double**

**dim XU(1000) as double**

**dim SA(1000) as double**

**dim a(1000) as double**

**ws= "c:\aggelos\"**

**MbeSendCommand "ACTIVE LEVEL 10"**

**MbeSendCommand "LOCK LEVEL TOGGLE "**

**MbeSendCommand "ACTIVE LEVEL 10"**

```
i=1
open ws+"sp3.dat" for input as 1
do while not eof(1)
input #1,XL(i),YL(i),YDL(i),YDDL(i),CURV(i)
i=i+1
loop
close #1
```

```
open ws+"XLSTEP.dat" for input as 2
input #2, XLSTEP, ds
close #2
```

```
open ws+"max.dat" for input as 5
input #5, max
close #5
```

```
X8(max)=0
```

```
for i=max to 1 step-1
```

```
Dim startPoint As MbePoint
```

```
Dim point As MbePoint, point2 As MbePoint
```

```
' Start a command
```

```
MbeSendCommand "MEASURE DISTANCE ICON "
```

```
' Coordinates are in master units
```

```
startPoint.x = XL(i)
```

```
startPoint.y = YL(i)
```

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = XL(i-1)**

**point.y = YL(i-1)**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**DIST = MbeState.measureResult1**

**if i=max then**

**goto HERE**

**end if**

**X8(i)=X8(i+1)+DIST/10000**

**k=x8(i)**

**XU(i)=k**

**HERE :**

**next i**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE SMARTLINE "**

**' Coordinates are in master units**

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(1)**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(1)**

**point.y = startPoint.y-5**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

**point.y = startPoint.y-5**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

```
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

' Start a command

```
MbeSendCommand "PLACE SMARTLINE "
```

' Coordinates are in master units

```
startPoint.x = 0.000000#
startPoint.y = -10
startPoint.z = 0.000000#
```

' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(1)
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = X8(1)
point.y = startPoint.y-5
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-5
point.z = startPoint.z
```



**MbeSendDataPoint point, 1%**

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = 0**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**' Start a command**

**MbeSendCommand "PLACE SMARTLINE "**

**' Coordinates are in master units**

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = X8(1)**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

```
point.x = X8(1)
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y-15
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
point.x = 0
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

```
mbesendreset
```

- ' Start a command

```
MbeSendCommand "WINDOW AREA EXTENDED 1"
```

- ' Coordinates are in master units

```
startPoint.x = 0.140800#
startPoint.y = 1.697900#
startPoint.z = 0.000000#
```

- ' Send a data point to the current command

```
point.x = startPoint.x
point.y = startPoint.y
point.z = startPoint.z
MbeSendDataPoint point, 1%
```

point.x = startPoint.x - 2.780100#  
point.y = startPoint.y - 19.975000#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

- ' Send a reset to the current command  
MbeSendReset

MbeSendCommand "PLACE LINE CONSTRAINED "

- ' Send a tentative point  
point.x = startPoint.x + 0.134200#  
point.y = startPoint.y - 1.601000#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.140800#  
point.y = startPoint.y - 1.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 0.091600#  
point.y = startPoint.y - 6.567200#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.140800#  
point.y = startPoint.y - 6.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 0.204500#  
point.y = startPoint.y - 11.561700#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.232800#  
point.y = startPoint.y - 11.787500#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.140800#  
point.y = startPoint.y - 11.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

point.x = startPoint.x - 0.176300#  
point.y = startPoint.y - 16.443300#  
point.z = startPoint.z  
MbeSendTentPoint point, 1%

point.x = startPoint.x - 0.140800#  
point.y = startPoint.y - 16.697900#  
point.z = startPoint.z  
MbeSendDataPoint point, 1%

MbeSendReset

MbeSendCommand "ZOOM OUT EXTENDED 1"

MbeSendCommand "PLACE LINE CONSTRAINED "

**MbeSendCommand "IGEN CONSTRAIN LENGTH UNLOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**MbeSendReset**

**point.x = startPoint.x - 19.964100#**

**point.y = startPoint.y - 1.911300#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 1.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 20.020500#**

**point.y = startPoint.y - 15.907100#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.738300#**

**point.y = startPoint.y - 16.415000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.140800#**

**point.y = startPoint.y - 16.697900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

- ' Send a message string to an application
- ' Content is defined by the application

**MbeSendAppMessage "TEXTEDIT", "FirstLine AYKSON ARI8MOS"**

**MbeSendCommand "ACTIVE ANGLE 0.0000°"**

**MbeSendCommand "ACTIVE ANGLE "**

**point.x = startPoint.x - 10.818200#**

**point.y = startPoint.y - 4.112200#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine XIL. 8ESH"**

**point.x = startPoint.x - 13.358800#**

**point.y = startPoint.y - 9.134900#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendAppMessage "TEXTEDIT", "FirstLine MHKOS ORATOTHTAS"**

**point.x = startPoint.x - 10.931100#**

**point.y = startPoint.y - 14.270500#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

- ' Send a keyin that can be a command string

**MbeSendKeyin Chr\$(27)**

## **MBESENDRESET**

- ' Start a command

**MbeSendCommand "PLACE SMARTLINE "**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

- ' Send a tentative point
- ' Coordinates are in master units

**MbeSendTentPoint -0.311400#, -0.091100#, 0.000000#, 1%**

- ' Coordinates are in master units

**startPoint.x = 0.000000#**

**startPoint.y = 0.000000#**

**startPoint.z = 0.000000#**

- ' Send a data point to the current command

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.790000#**

**point.y = startPoint.y - 0.261900#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**' Send a reset to the current command**

**MbeSendReset**

**MbeSendCommand "IGEN CONSTRAIN ANGLE LOCK "**

**MbeSendCommand "IGEN CONSTRAIN ANGLE 180.000000"**

**point.x = startPoint.x - 0.396800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 5.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**



**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.704500#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**

**point.y = startPoint.y - 5.044400#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.960800#**

**point.y = startPoint.y - 4.959000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**point.x = startPoint.x - 0.055100#**

**point.y = startPoint.y - 10.083000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**  
**point.y = startPoint.y - 10.000000#**  
**point.z = startPoint.z**  
**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.362800#**  
**point.y = startPoint.y - 9.826800#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 19.875400#**  
**point.y = startPoint.y - 9.997600#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**  
**point.y = startPoint.y - 10.083000#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.131700#**  
**point.y = startPoint.y - 9.997600#**  
**point.z = startPoint.z**  
**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.046300#**  
**point.y = startPoint.y - 9.826800#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y - 7.500000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**MbeSendReset**

**point.x = startPoint.x - 0.311400#**

**point.y = startPoint.y - 15.036200#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x**

**point.y = startPoint.y - 15.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**point.x = startPoint.x - 19.448200#**

**point.y = startPoint.y - 14.950800#**

**point.z = startPoint.z**

**MbeSendTentPoint point, 1%**

**point.x = startPoint.x - 20.000000#**

**point.y = startPoint.y - 15.000000#**

**point.z = startPoint.z**

**MbeSendDataPoint point, 1%**

**mbesendreset**

**MbeSendKeyIn "LC=0"**

**MbeSendKeyIn "CO=2"**

**MbeSendKeyIn "WT=0"**

**mbesendreset**

**open ws+"apaitorat.dat" for input as 10**

**input #10, ve,Su,Ss**

**close #10**

**i=1**

**open ws+"oratotitaStrd-a.dat" for input as 6**

**do while not eof (6)**

**input #6, a(i),ORAT(i)**

**m=Orat(i)\*10**

**SA(i)=m/10**

**i=i+1**

**loop**

**mfin=i**

**close #6**

**MbeSendKeyIn "CO=2"**

**for i=max-1 to (max-mfin+1) step-1**

**' Start a command**

**MbeSendCommand "PLACE LINE CONSTRAINED "**

**' Coordinates are in master units**

**startPoint.x = X8(i)**

**startPoint.y = 0**

**startPoint.z = 0.000000#**

**' Send a data point to the current command**

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(i)**

**point.y = SA(max-i)/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**next i**

**for i=max-1 to (max-mfin+1) step-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(XU(i))**

**MbeSendDataPoint X8(i),-7.5,0,1%**

**mbesendreset**

**next i**

**for i=max-1 to (max-mfin+1) step-1**

**MbeSendKeyIn "CO=0"**

**MbeSendCommand "PLACE DIALOGTEXT ICON "**

**MbeSendCommand "ACTIVE ANGLE "**

**MbeSendCommand "ACTIVE ANGLE 0°"**

**MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(max-i)**

**MbeSendDataPoint X8(i),-2.5,0,1%**

**mbesendreset**

**next i**

for i=max-1 to (max-mfin+1) step-1

MbeSendKeyIn "CO=0"

MbeSendCommand "PLACE DIALOGTEXT ICON "

MbeSendCommand "ACTIVE ANGLE "

MbeSendCommand "ACTIVE ANGLE 0°"

MbeSendAppMessage "TEXTEDIT","FirstLine " +cstr(SA(max-i))

MbeSendDataPoint X8(i),-12.5,0,1%

mbesendreset

next i

MbeSendCommand "PLACE DIALOGTEXT ICON "

MbeSendCommand "ACTIVE ANGLE "

MbeSendCommand "ACTIVE ANGLE 0°"

MbeSendAppMessage "TEXTEDIT","FirstLine "+"DIAGRAMMA ORATOTHTAS  
GIA STASH SE POREIA D-A"

MbeSendDataPoint 0,-16,0,1%

MbeSendKeyIn "CO=3"

'Start a command

**MbeSendCommand "PLACE LINE CONSTRAINED "**

- ' Coordinates are in master units

**startPoint.x = 0**

**startPoint.y = Ss/10**

**startPoint.z = 0.000000#**

- ' Send a data point to the current command

**point.x = startPoint.x**

**point.y = startPoint.y**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**MbeSendCommand "IGEN CONSTRAIN ANGLE UNLOCK "**

**point.x = X8(max-mfin+1)**

**point.y = Ss/10**

**point.z = startPoint.z**

**MbeSendDataPoint point, 2%**

**mbesendreset**

**End Sub**