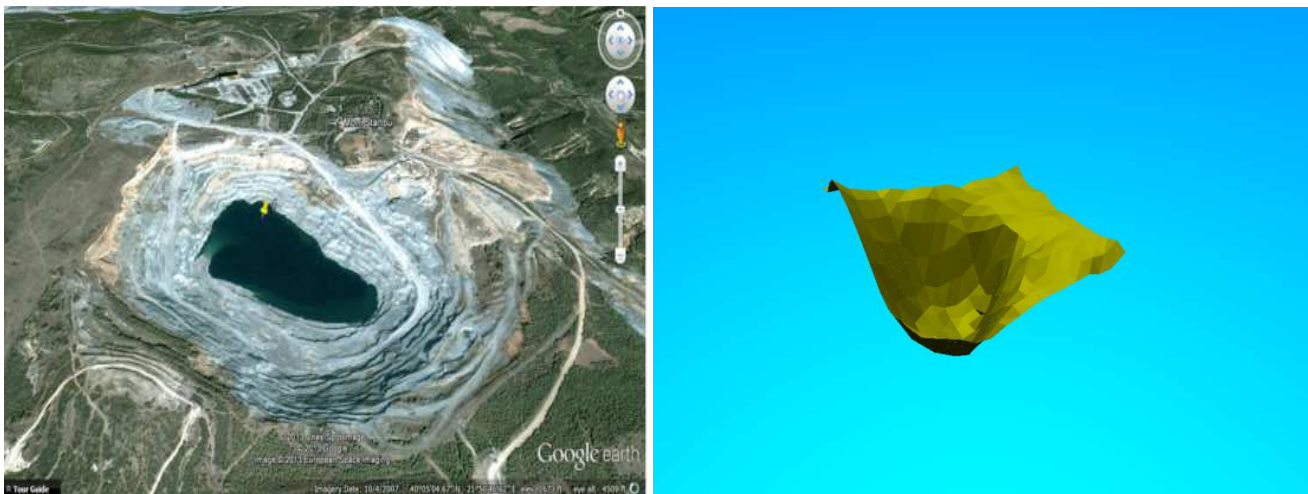




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υπολογισμός και προοπτική απεικόνιση θέσεων λιμνοδεξαμενών με
παγκόσμια ψηφιακά μοντέλα εδάφους



Εκπόνηση

Παπαδάκης Λουκάς

Επιβλέπων καθηγητής

Εμμανουήλ Βουγιούκας

Συνεπιβλέπων

Δρ. Αθανάσιος Στάμος

Αθήνα

Μάρτιος 2013

Ευχαριστίες

Η παρούσα εργασία δεν θα είχε υλοποιηθεί χωρίς την σημαντική συμβολή του καθηγητή μου. Για το λόγο αυτό θα ήθελα να εκφράσω τις ευχαριστίες μου στον Δρ. Αθανάσιο Στάμο για την άψογη καθοδήγηση και συνεργασία.

Ακόμα θα ήθελα να ευχαριστήσω του γονείς μου για όσα μου έχουν προσφέρει σε όλη τη διάρκεια των σπουδών μου και όχι μόνο καθώς και τον φίλο μου και νέο συνάδελφο Γωνιανάκη Παύλο για τη βοήθειά του και τη συνεχή στήριξή του.

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας είναι η χρήση του παγκόσμιου ψηφιακού μοντέλου εδάφους SRTM (Shuttle Radar Topography Mission) για τον υπολογισμό θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας, καθώς και για την προοπτική απεικόνισή τους.

Αναπτύχθηκαν τέσσερις μέθοδοι για την επεξεργασία των δεδομένων του SRTM και κατασκευάστηκαν προγράμματα Η/Υ σε γλώσσα FORTRAN για την εφαρμογή αυτών των μεθόδων.

Μετά τον υπολογισμό ικανοποιητικού αριθμού θέσεων πιθανών λιμνοδεξαμενών, ακολούθησε ο εντοπισμός τους στο χάρτη, χρησιμοποιώντας το πρόγραμμα Google earth, καθώς και η προοπτική απεικόνιση των θέσεων αυτών με χρήση των δεδομένων του SRTM.

ABSTRACT

The purpose of this thesis is the use of Global Digital Elevation project SRTM (Shuttle Radar Topography Mission) to locate small reservoirs in the region of Greece.

Four methods were developed for the processing of the STRM data using FORTRAN code.

Following the calculation of a sufficient number of possible places, their location on the map was found, using Google Earth. Afterwards, the above locations were perspectively depicted using the STRM data.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή	4
2. Ψηφιακό Μοντέλο Εδάφους SRTM	6
2.1 Γενικά	6
2.2 Παραγωγή SRTM	6
2.3 Διανομή SRTM	8
3. Μεθοδολογία 1	10
3.1 Γενικά	10
3.2 Μέθοδος.....	10
3.3 Ανάγνωση Υψομετρικών Δεδομένων SRTM	14
3.4 Επεξεργασία Δεδομένων – Έυρεση Λιμνοδεξαμενών	16
3.5 Αποτελέσματα Μεθόδου	18
3.6 Απεικόνιση Αποτελεσμάτων	18
4. Μεθοδολογία 2	21
4.1 Γενικά	21
4.2 Μέθοδος	21
4.3 Επεξεργασία Δεδομένων – Εύρεση Λιμνοδεξαμενών	25
4.4 Αποτελέσματα Μεθόδου	26
4.5 Απεικόνιση Αποτελεσμάτων	27
5. Μεθοδολογία 3	29
5.1 Γενικά	29
5.2 Μέθοδος	29
5.3 Επεξεργασία Δεδομένων – Εύρεση Λιμνοδεξαμενών	33
5.4 Αποτελέσματα Μεθόδου	34
5.5 Απεικόνιση Αποτελεσμάτων	35

6. Μεθοδολογία 4	40
6.1 Γενικά	40
6.2 Μέθοδος	40
6.3 Επεξεργασία Δεδομένων – Εύρεση Λιμνοδεξαμενών	42
6.4 Αποτελέσματα Μεθόδου	42
6.5 Απεικόνιση Αποτελεσμάτων	45
7. Προοπτική Απεικόνιση	52
7.1 Γενικά	52
7.2 Παρακολούθηση Οπτικής Ακτίνας Με Το Λογισμικό POV-RAY	52
7.3 Εφαρμογή	53
8. Παραδείγματα	55
8.1 Γενικά	55
8.2 Παράδειγμα1	55
8.3 Παράδειγμα2	58
8.4 Παράδειγμα3	61
8.5 Παράδειγμα4	64
9. Συμπεράσματα	68
Παράρτημα Α	
Παράρτημα Α.1.1 Αποτελέσματα Πρώτης Μεθόδου	71
Παράρτημα Α.1.2 Μηκοτομές Αποτελεσμάτων Πρώτης Μεθόδου	72
Παράρτημα Α.2.1 Αποτελέσματα Δεύτερης Μεθόδου	77
Παράρτημα Α.2.2 Μηκοτομές Αποτελεσμάτων Δεύτερης Μεθόδου	79
Παράρτημα Α.3.1 Αποτελέσματα Τρίτης Μεθόδου	82
Παράρτημα Α.3.2 Μηκοτομές Αποτελεσμάτων Τρίτης Μεθόδου	83
Παράρτημα Α.4.1 Αποτελέσματα Τέταρτης Μεθόδου	86
Παράρτημα Α.4.2 Μηκοτομές Αποτελεσμάτων Τέταρτης Μεθόδου	96

Παράρτημα Β

Παράρτημα Β.1 Πρόγραμμα limnodeksameni1	101
Παράρτημα Β.2 Πρόγραμμα limnodeksameni2	108
Παράρτημα Β.3 Πρόγραμμα limnodeksameni2	118
Βιβλιογραφία	133

1. ΕΙΣΑΓΩΓΗ

Την τελευταία δεκαετία, η έκρηξη του πληθυσμού της γης καθώς και οι αυξανόμενες απαιτήσεις σε νερό για ύδρευση αλλά και άρδευση δημιουργούν την ανάγκη για μεγαλύτερη κατανάλωση νερού. Προς αυτή την κατεύθυνση οδηγεί και η άνθιση του τουριστικού τομέα ιδιαίτερα στις Μεσογειακές χώρες και κυρίως στην Ελλάδα. Ακόμη, οι παγκόσμιες κλιματικές αλλαγές, αποτέλεσμα του φαινομένου του θερμοκηπίου, αναμένεται να επιφέρουν ποικίλες και μακροπρόθεσμες επιπτώσεις. Οι πρώτες συνέπειες παρατηρούνται ήδη, με την αύξηση στην έκταση και την ένταση της ερημοποίησης προς βορρά στις περιοχές που είναι ευάλωτες. Επιπλέον, διαπιστώνεται σημαντική μείωση των αποθεμάτων νερού σε χώρες όπως η Αίγυπτος, η Τυνησία, και οι υπόλοιπες χώρες της Μεσογείου. Συχνότερα, υδατικά αποθέματα κρίνονται ακατάλληλα για χρήση λόγω της ανόδου της στάθμης της θάλασσας αλλά και λόγω της ρύπανσης των υδάτων.

Η χρήση υδατοσυλλογών μπορεί να αποτελέσει τη λύση των παραπάνω προβλημάτων. Τα άμεσα οφέλη από τη δημιουργία μικρού αλλά και μεγάλου μεγέθους λιμνοδεξαμενών και φραγμάτων εμφανίζονται στην ύδρευση πόλεων, οικισμών αλλά και βιομηχανικών μονάδων, στην άρδευση καλλιεργειών και στην παραγωγή υδροηλεκτρικής ενέργειας. Έμμεσα, οι περιοχές δημιουργίας τέτοιων έργων αποκτούν αντιπλημμυρική προστασία ενώ επιπλέον σημειώνεται ανάπτυξη της ευρύτερης περιοχής.

Με αφορμή τα παραπάνω, στην παρούσα διπλωματική εργασία επιχειρείται η εύρεση πιθανών θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας καθώς και η προοπτική απεικόνισή τους χρησιμοποιώντας δεδομένα από ένα παγκόσμιο ψηφιακό μοντέλο εδάφους.

Στόχος της εργασίας είναι να βρεθούν περιοχές στην Ελλάδα που έχουν μορφολογία εδάφους που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή, χρησιμοποιώντας τα δεδομένα που έχουμε στη διάθεσή μας από το παγκόσμιο ψηφιακό μοντέλο επιφάνειας SRTM (Shuttle Radar Topography Mission). Για το σκοπό αυτό κατασκευάζουμε ένα πρόγραμμα Η/Υ σε γλώσσα FORTRAN το οποίο επεξεργάζεται με κατάλληλο τρόπο τα δεδομένα.

Κατά την εκπόνηση της εργασίας προέκυψαν απρόβλεπτες δυσκολίες, κυρίως διότι δεν έχει επιχειρηθεί ο εντοπισμός λιμνοδεξαμενών από το SRTM. Για να αντιμετωπιστούν οι δυσκολίες, αναπτύχθηκαν διάφορες μέθοδοι οι οποίες περιγράφουν πως επεξεργαζόμαστε τα δεδομένα του SRTM με σκοπό τον υπολογισμό θέσεων λιμνοδεξαμενών.

Αφού εντοπιστούν οι κατάλληλες θέσεις, ακολουθεί η απεικόνισή τους με διάφορους τρόπους, όπως με τη δημιουργία μηκοτομών και την προοπτική απεικόνισή τους έτσι ώστε να επιβεβαιωθεί ότι η περιοχή έχει τη ζητούμενη μορφολογία. Για την απεικόνιση χρησιμοποιούνται ξανά τα δεδομένα του ψηφιακού μοντέλου επιφάνειας SRTM.

Τέλος, γίνεται εντοπισμός των θέσεων στο πρόγραμμα Google earth ώστε να διαπιστωθεί η ορθότητα των αποτελεσμάτων. Παράλληλα, με τον τρόπο αυτό, γίνεται και έλεγχος της ορθότητας των δεδομένων του SRTM και επαληθεύεται αν αυτά συμφωνούν με τα πραγματικά.

2. ΤΟ ΠΑΓΚΟΣΜΙΟ ΨΗΦΙΑΚΟ ΜΟΝΤΕΛΟ ΕΛΑΦΟΥΣ SRTM

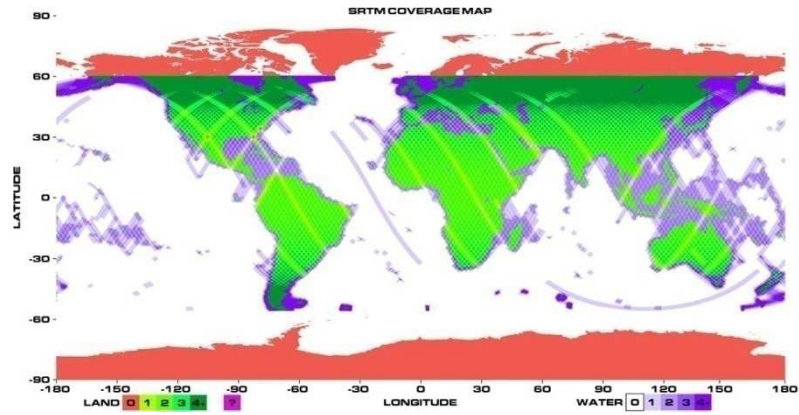
2.1 ΓΕΝΙΚΑ

Η διαστημική αποστολή SRTM (Shuttle Radar Topography Mission) [1] είναι αποτέλεσμα διακρατικής συνεργασίας των διαστημικών υπηρεσιών NASA (National Aeronautics and Space Administration), DLR (German Aerospace Center) και ASI (Italian Space Agency). Αντικειμενικός σκοπός της αποστολής ήταν η δημιουργία ενός ενιαίου και ομοιογενούς παγκόσμιου ΨΜΕ για σχεδόν το σύνολο (80%) της γήινης επιφάνειας (από 56° N έως 60° B), πράγμα το οποίο δεν είχε πραγματοποιηθεί μέχρι τότε.

2.2 ΠΑΡΑΓΩΓΗ SRTM

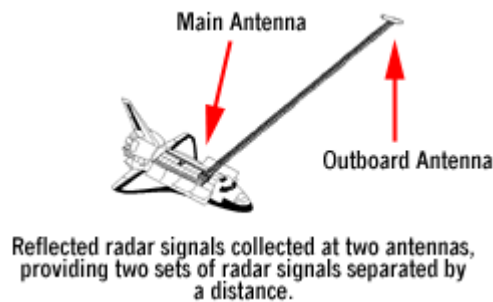
Το πρόγραμμα ξεκίνησε τον Φεβρουάριο του 2000 με την εκτόξευση του διαστημόπλοιου Space Shuttle Endeavour. Το διαστημόπλοιο, επανδρωμένο και κινούμενο σε υψόμετρο 233 Km, με ταχύτητα 7.5Km/sec πραγματοποίησε 176 περιφορές γύρω από τη Γη, κατά τη διάρκεια 11 ημερών το Φεβρουάριο του 2000. Η αποστολή ήταν πρωτοποριακή για πολλούς λόγους που αφορούν τόσο στην τεχνολογία που αναπτύχθηκε για την επιτυχία του εγχειρήματος (μεγαλύτερη κατασκευή που πέταξε ποτέ στο διάστημα, πρώτο μονής διέλευσης συμβολομετρικό δορυφορικό σύστημα) όσο και στο τελικό αποτέλεσμα (ενιαίο και ομοιογενές ΨΜΕ της γήινης επιφάνειας) [1].

Ένα ειδικά σχεδιασμένο σύστημα ραντάρ λειτουργούσε πάνω στο διαστημόπλοιο επί 11 μέρες συλλέγοντας δεδομένα και οδηγώντας στην καταγραφή πληροφοριών για το μεγαλύτερο μέρος της επιφάνειας της γης που εκτείνεται μεταξύ του βόρειου γεωγραφικού πλάτους των 60 μοιρών και του νότιου γεωγραφικού πλάτους των 54 μοιρών. Η έκταση αυτή ισοδυναμεί σχεδόν με το 80% της επιφάνειας της γης, ενώ σε αυτή διαμένει σχεδόν το 95% του παγκόσμιου πληθυσμού (σχήμα1).



Σχήμα 1. Η κάλυψη της γήινης επιφάνειας. Πηγή:<http://www2.jpl.nasa.gov/srtm>

Η επεξεργασία των μετρήσεων έγινε με τεχνικές συμβολομετρίας με χρήση ραντάρ (radar interferometry). Σύμφωνα με αυτές συλλέγονται δυο εικόνες ραντάρ από θέσεις με μικρή απόσταση και η σύγκριση αυτών μας επιτρέπει τον υπολογισμό του υψομέτρου της επιφάνειας. Δύο τύποι δεκτών ραντάρ SAR [3], στο κανάλι C και στο κανάλι X, τοποθετήθηκαν στο διαστημόπλοιο Space Shuttle Endeavour. Οι κύριες κεραίες και των δύο τύπων δεκτών ήταν τοποθετημένες στην άτρακτο του Endeavour ενώ οι δευτερεύουσες ήταν αναρτημένες στην άκρη μεταλλικού δικτύωματος μήκους 60m, το οποίο αναπτύχθηκε όταν η αποστολή βρισκόταν στο διάστημα (σχήμα 2). Ο αρχικός σχεδιασμός προέβλεπε και την τοποθέτηση δεκτών τύπου L, κάτι το οποίο τελικά δεν πραγματοποιήθηκε για την αποφυγή υπερφόρτωσης του διαστημοπλοίου.



Σχήμα 2. Το διαστημόπλοιο Space Shuttle Endeavour με τους δέκτες

Η χρήση ραντάρ πλεονεκτεί σε σχέση με τις οπτικές κάμερες, καθώς μπορούν να λειτουργούν και κατά τη διάρκεια της νύχτας, ενώ μπορούν να διαπερνούν τα σύννεφα. Τέλος με την τοποθέτηση των ραντάρ στο διαστημόπλοιο ξεπερνάμε τα εμπόδια προσέγγισης στην τοποθεσία καταγραφής.

Τα αρχικά δεδομένα του SRTM περιέχουν κενά χωρίς δεδομένα, σε περιοχές όπου υπήρχε νερό ή ισχυρή νέφωση που εμπόδιζε την καταγραφή. Για τον λόγο αυτό έχουν περάσει ορισμένα στάδια επεξεργασίας ώστε να παρέχουν συνεχή και χωρίς κενά δεδομένα για το υψόμετρο και τη μορφή της επιφάνειας της γης [7]. Τα πρώτα δεδομένα του προγράμματος SRTM δημοσιεύτηκαν το 2003 και είχαν μορφή μονοβάθμιου ψηφιακού μοντέλου εδάφους. Η δημοσίευση γινόταν για κάθε ήπειρο μόλις τελείωνε η επεξεργασία τους από τη NASA και την USGS. Τα δεδομένα που αφορούσαν τις Ηνωμένες Πολιτείες είχαν ακρίβεια προσεγγιστικά 30 μέτρα στον ισημερινό, ενώ για τον υπόλοιπο κόσμο η ακρίβεια μειώθηκε σε 90 μέτρα στον ισημερινό προσεγγιστικά. Το ψηφιακό μοντέλο εδάφους που δημιουργήθηκε διατίθεται ελεύθερα και δωρεάν και καλύπτει το 80% της γήινης επιφάνειας.

2.3 ΔΙΑΝΟΜΗ SRTM

Το SRTM καλύπτει σχεδόν όλη την επιφάνεια της γης. Ακόμα και με ένα σημείο ανά 90 μέτρα, το πλήθος των σημείων είναι τεράστιο πράγμα που δυσχεραίνει την διανομή τους σε όλους τους ενδιαφερόμενους. Μια σημαντική παρατήρηση για τη διευκόλυνση της διανομής ήταν ότι η χρήση του SRTM στην συντριπτική πλειοψηφία των περιπτώσεων είναι τοπική. Για παράδειγμα ένας ερευνητής συνήθως ενδιαφέρεται για το τμήμα του SRTM που αντιστοιχεί στη χώρα του ή ακόμα και στην μόνο σε ένα νομό της χώρας του. Έτσι το SRTM χωρίστηκε σε “τετραγωνικές” περιοχές που καλύπτουν γεωγραφικό μήκος και πλάτος 5’ x 5’ (πρώτα λεπτά της μοίρας) της γήινης επιφάνειας. Κάθε περιοχή δίνεται ως ξεχωριστό αρχείο με την ονομασία `srtm_<i><j>.tif` όπου <i> και <j> ακέραιοι αριθμοί που εξαρτώνται από το γεωγραφικό μήκος και πλάτος

$$i = \left[\frac{\lambda + 180}{5} + 1 \right] \quad j = \left[\frac{60 - \phi}{5} + 1 \right]$$

Όπου λ, φ δίνονται σε μοίρες και [...] συμβολίζει αποκοπή δεκαδικών.

Ακόμα και με το χωρισμό σε περιοχές, το κάθε αρχείο έχει σημαντικό αριθμό σημείων: $N = 6001 \times 6001 = 36 \cdot 10^6$

Θεωρώντας 8 bytes για κάθε συντεταγμένη (x, y, h) το μέγεθος του αρχείου είναι : $M = 3 \times 8 \times 36 \cdot 10^6 = 864 \cdot 10^6 \text{ bytes} = 864 \text{ Mbytes}$

Στην Ελλάδα χρειάζονται 6 αρχεία: $\Sigma M = 6M = 6 \times 864 \text{ Mbytes} = 5.2 \text{ GBytes}$. Για να μειωθεί ο όγκος του αρχείου παρατηρούμε ότι τα σημεία είναι σε κানাβο και έτσι δεν χρειάζεται να αποθηκευτούν οι συντεταγμένες x και y. Επιπλέον, τα υψόμετρα του SRTM μοιάζουν με τα

εικονοστοιχεία (pixels) μιας εικόνας. Η διαφορά έγκειται ότι στο SRTM τα pixels δεν εκφράζουν αποχρώσεις του γκρι, αλλά υψόμετρα και τα υψόμετρα είναι πραγματικοί αριθμοί και όχι ακέραιοι. Έτσι τα υψόμετρα αποθηκεύονται σε αρχεία εικόνας .tif ως πραγματικοί αριθμοί. Επειδή τα υψόμετρα δεν είναι μεγάλης ακρίβειας, αρκούν οι μισής ακρίβειας πραγματικοί αριθμοί (half precision floating point) που χρειάζονται 2 bytes. Έτσι το μέγεθος των αρχείων γίνεται $M = 2 \times 36 \cdot 10^6 = 72$ MBytes που είναι πολύ πιο εύκολο να διανεμηθεί.

3. ΜΕΘΟΔΟΛΟΓΙΑ 1

3.1 ΓΕΝΙΚΑ

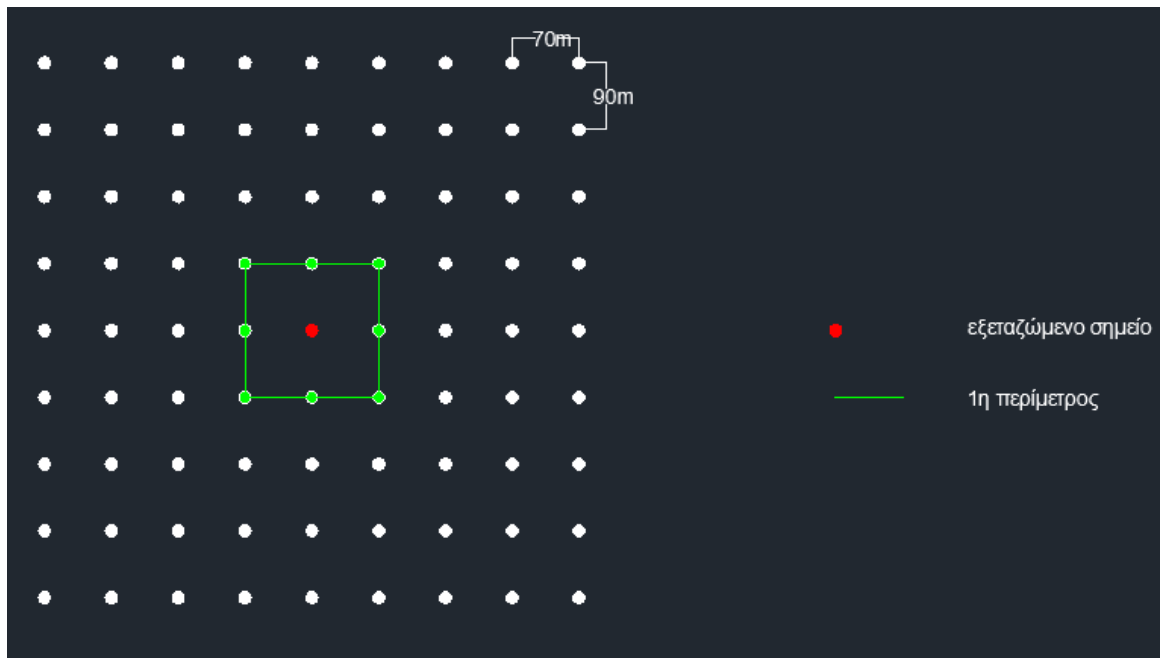
Στο κεφάλαιο αυτό αναπτύσσεται η πρώτη μεθοδολογία για την εύρεση θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας με χρήση των υψομετρικών δεδομένων που έχουμε στη διάθεσή μας από το παγκόσμιο ψηφιακό μοντέλο SRTM. Στην πρώτη μέθοδο κάθε σημείο μακριά από το κέντρο της (πιθανής) λιμνοδεξαμενής πρέπει να έχει μεγαλύτερο υψόμετρο από σημεία που είναι πιο κοντά στο κέντρο.

3.2 ΜΕΘΟΔΟΣ

Το παγκόσμιο ψηφιακό μοντέλο εδάφους SRTM δίνει τα υψόμετρα του εδάφους σε συγκεκριμένα σημεία. Τα σημεία αυτά βρίσκονται σε κανάβο βήματος $\Delta\lambda=\Delta\phi=3''$ (δεύτερα λεπτά μοίρας). Τα $\Delta\lambda$ και $\Delta\phi$ αναφέρονται σε γεωγραφικό μήκος και γεωγραφικό πλάτος αντίστοιχα. Στο Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς 1987 (ΕΓΣΑ87), το βήμα κανάβου αντιστοιχεί σε περίπου $\Delta X=70m$ κατά X και $\Delta Y=90m$ κατά Y . Η διαφορά μεταξύ ΔX και ΔY οφείλεται στο γεγονός ότι οι παράλληλοι στο γεωγραφικό πλάτος της Ελλάδας είναι μικρότεροι από τον ισημερινό, ενώ όλοι οι μεσημβρινοί έχουν το ίδιο μήκος.

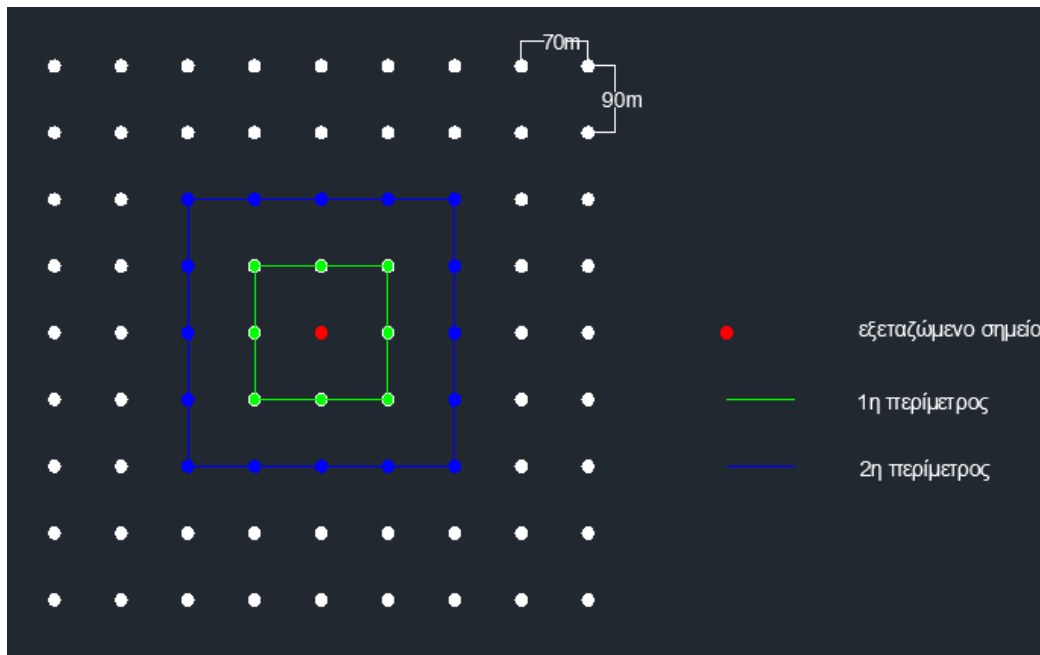
Στην πρώτη μέθοδο γίνεται αναζήτηση για φυσικές κοιλάτητες οι οποίες μπορούν θεωρητικά να γίνουν λιμνοδεξαμενές με πολύ χαμηλό κόστος, επειδή αποφεύγεται η κατασκευή φράγματος. Ως κοιλάτητα θεωρείται μία περιοχή της οποίας όλα τα σημεία της περιμέτρου έχουν μεγαλύτερο υψόμετρο από τα εσωτερικά σημεία της, δηλαδή από όλα τα σημεία που περικλείονται από την περίμετρο.

Στην πρώτη μέθοδο αναζητούμε κοιλάτητες στις οποίες το υψόμετρο αυξάνει μονότονα από το μέσο της κοιλάτητας προς την περίμετρό της. Προκειμένου να βρούμε μία τέτοια κοιλάτητα στο έδαφος, ξεκινούμε από τυχαίο σημείο του κανάβου και ελέγχουμε το υψόμετρο των 8 γειτονικών του σημείων. Αν το υψόμετρο και των 8 γειτονικών σημείων είναι μεγαλύτερο ή ίσο από το αρχικό, τότε εκ των πραγμάτων το αρχικό σημείο είναι το κέντρο “τετραγωνικής” κοιλάτητας με μήκος $\Delta\lambda=6''$ και πλάτος $\Delta\phi=6''$. Η περίμετρος της κοιλάτητας ονομάζεται 1η περίμετρος αφού γειτνιάζει άμεσα με το αρχικό κεντρικό σημείο (σχήμα 3.1).



Σχήμα 3.1. πρώτη περίμετρος

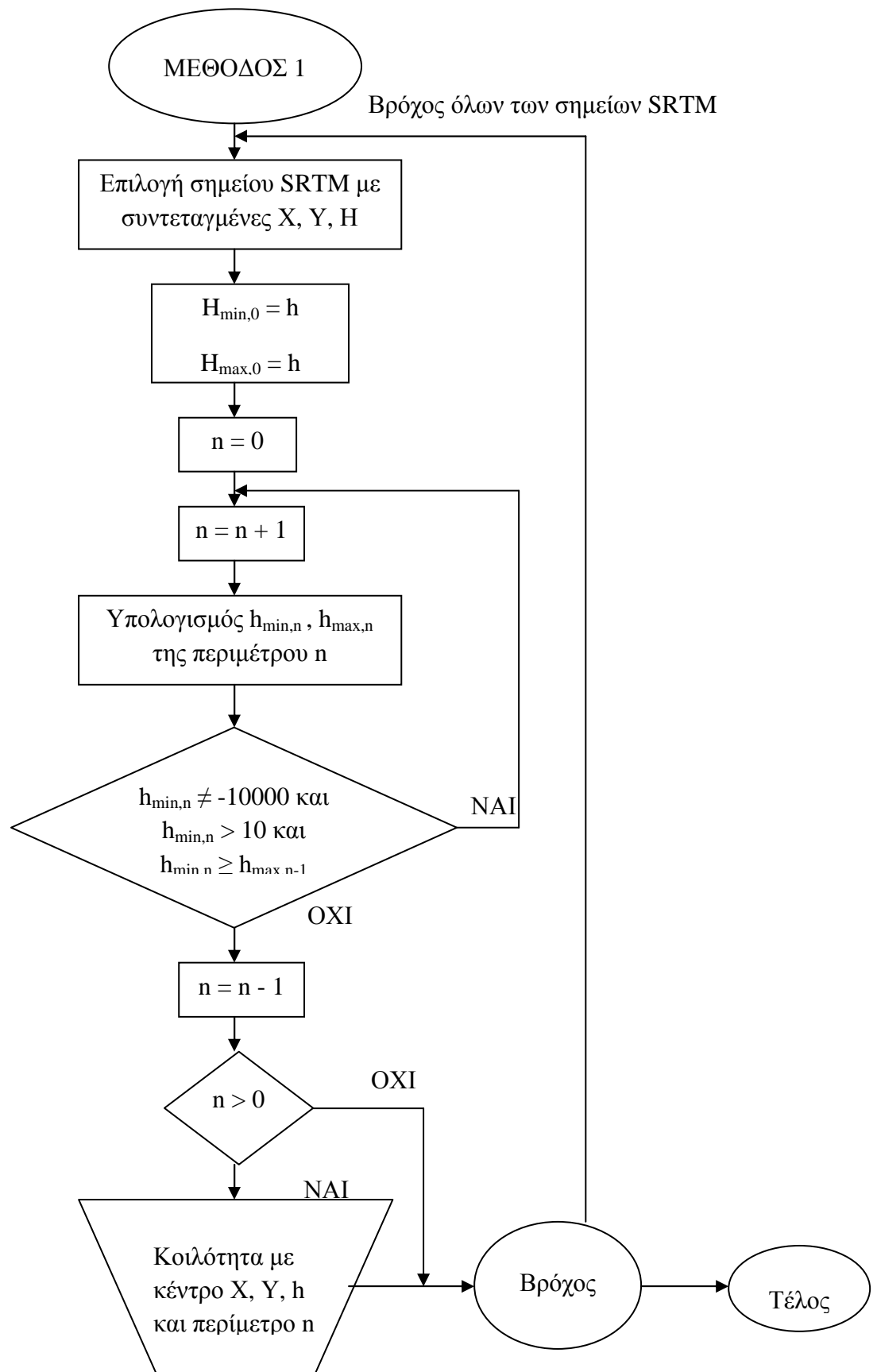
Στη συνέχεια ελέγχουμε τα γειτονικά σημεία όλων των σημείων της 1ης περιμέτρου, εξαιρουμένου του κεντρικού σημείου (που ήδη γνωρίζουμε ότι έχει χαμηλότερο υψόμετρο). Η περίμετρος που αντιστοιχεί σε αυτά τα 16 σημεία ονομάζεται 2η περίμετρος επειδή γειτνιάζει με την πρώτη περίμετρο, αλλά βρίσκεται μακρύτερα από το αρχικό σημείο απ' ότι τα σημεία της πρώτης περιμέτρου (σχήμα 3.2). Αν το υψόμετρο και των 16 σημείων της 2ης περιμέτρου είναι μεγαλύτερα ή ίσα από το υψόμετρο όλων των σημείων της 1ης περιμέτρου, τότε το κεντρικό σημείο είναι κοιλότητα με περίμετρο τη 2η. Η κοιλότητα έχει διαστάσεις $\Delta\lambda=12''$ και $\Delta\phi=12''$ δηλαδή είναι κατά 6'' μεγαλύτερη από την προηγούμενη.



Σχήμα 3.2. δεύτερη περίμετρος

Για να εξακριβώσουμε αν τα σημεία μιας περιμέτρου έχουν μεγαλύτερο υψόμετρο από τα σημεία της προηγούμενης περιμέτρου, υπολογίζουμε το ελάχιστο υψόμετρο της περιμέτρου. Αυτό το συγκρίνουμε με το μέγιστο υψόμετρο της προηγούμενης περιμέτρου.

Παρομοίως ελέγχουμε τα σημεία της 3ης περιμέτρου, δηλαδή για τα σημεία που γειτνιάζουν με τα σημεία της 2ης περιμέτρου αλλά δεν περικλείονται από αυτή. Αν τα υψόμετρα και των 24 σημείων της 3ης περιμέτρου είναι μεγαλύτερα και από τα 16 σημεία της προηγούμενης (2ης) περιμέτρου, τότε η κοιλότητα περικλείεται από την 3η περίμετρο. Η ίδια διαδικασία επαναλαμβάνεται για την 4, 5, 6 κλπ περιμέτρους. Η διαδικασία τερματίζεται όταν το υψόμετρο ενός σημείου μίας περιμέτρου είναι μικρότερο από το υψόμετρο ενός σημείου της αμέσως προηγούμενης περιμέτρου, οπότε και η κοιλότητα περικλείεται από την αμέσως προηγούμενη περίμετρο. Η μέθοδος περιπλέκεται από το γεγονός ότι το SRTM σε μερικά σημεία δεν έχει υψόμετρο για λόγους που οφείλονται στον ίδιο το δέκτη SAR (στο όργανο μέτρησης). Σε αυτά τα σημεία το υψόμετρο έχει τη συμβολική τιμή -10000.0. Η λύση που επιλέχθηκε είναι όταν βρεθεί ένα σημείο με υψόμετρο -10000.0 να τερματίζεται η διαδικασία για τις περιμέτρους. Παρομοίως αν κάποιο υψόμετρο μιας περιμέτρου είναι κάτω από 10m θεωρούμε ότι η πιθανή λιμνοδεξαμενή είναι πάρα πολύ κοντά στη θάλασσα για να είναι χρήσιμη. Ο αλγόριθμος συνοψίζεται στο σχήμα 3.3.



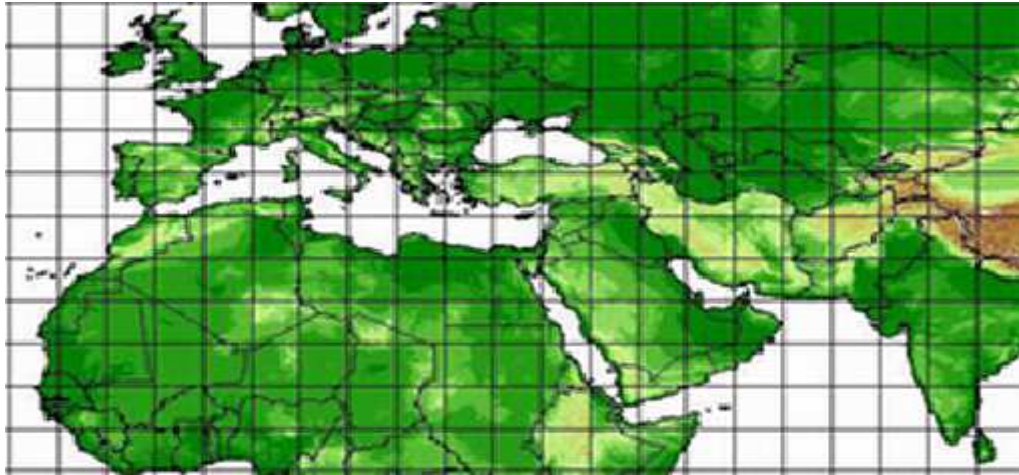
Θεωρώντας ότι το όριο της κοιλότητας είναι η n-οστή περίμετρος, η κοιλότητα σε κάτοψη είναι “τετραγωνική” με πλευρά μήκους $\Delta l = \Delta \varphi = 6n$ " και εμβαδό $36n^2$ τετραγωνικά δεύτερα της μοίρας. Στο ΕΓΣΑ87 η κοιλότητα είναι ορθογωνική με πλευρές $\Delta X \approx 140n$ m και $\Delta Y \approx 180n$ m και εμβαδό $E \approx 25200n^2$ m².

Για τη μέθοδο κατασκευάστηκε πρόγραμμα H/Y (limnodeksameni1) σε γλώσσα FORTRAN που εφαρμόζει τον παραπάνω αλγόριθμο και υπολογίζει τις πιθανές θέσεις λιμνοδεξαμενών. Ο κώδικας του προγράμματος βρίσκεται στο παράρτημα Β.1.

3.3 ΑΝΑΓΝΩΣΗ ΥΨΟΜΕΤΡΙΚΩΝ ΔΕΔΟΜΕΝΩΝ SRTM

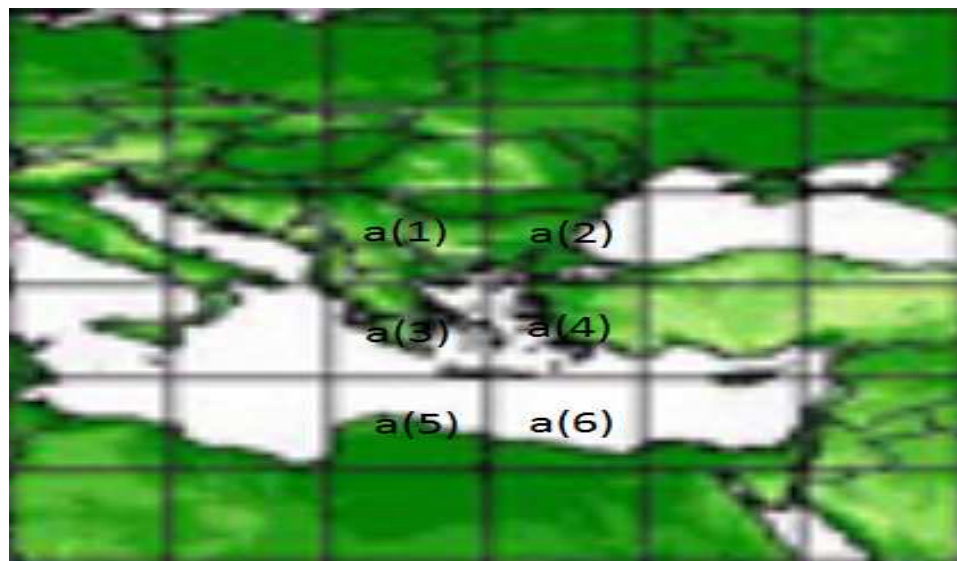
Η εισαγωγή των δεδομένων για τα υψόμετρα των σημείων γίνεται με τη βοήθεια των υπορουτίνων thanSet, openBmp και closeBmp. Οι υπορουτίνες openBmp και closeBmp χρησιμοποιούνται για να ανοίξουν και να κλείσουν αντίστοιχα το αρχείο στο οποίο είναι αποθηκευμένα τα δεδομένα από το πρόγραμμα SRTM. Η υπορουτίνα thanSet διαβάζει ένα αρχείο του SRTM και αποθηκεύει τα υψόμετρα των σημείων σε ένα πίνακα πραγματικών αριθμών (HIM). Αποθηκεύει το πλήθος των γραμμών (NYROWS) και των στηλών (NXCOLS) του πίνακα, τις γεωγραφικές συντεταγμένες του πρώτου σημείου του πίνακα και πιο συγκεκριμένα το γεωγραφικό μήκος (λ) (X0) και το γεωγραφικό πλάτος (φ) (Y0) του σημείου. Τέλος, αποθηκεύεται η διαφορά στο γεωγραφικό μήκος (λ) μεταξύ διαδοχικών γραμμών (DX) και η διαφορά στο γεωγραφικό πλάτος (φ) μεταξύ διαδοχικών στηλών (DY) του πίνακα.

Όλες αυτές οι πληροφορίες, που διαβάζονται από ένα αρχείο του SRTM, αποθηκεύονται σε μια μεταβλητή τύπου DEMUSGS. Ο τύπος αυτός έχει οριστεί στην αρχή του προγράμματος και περιέχει όλες τις μεταβλητές που αναφέρθηκαν παραπάνω. Έτσι, κάθε φορά που καλείται το υποπρόγραμμα thanSet αντιστοιχίζεται ένα αρχείο του SRTM, δηλαδή ένα κομμάτι του χάρτη, με μια μεταβλητή τύπου DEMUSGS την οποία εμείς ορίζουμε.



Εικόνα 3.3. θέσεις των αρχείων του SRTM στο χάρτη

Η εικόνα 3.3 είναι από την ιστοσελίδα με τα αρχεία του SRTM και χρησιμοποιείται για την επιλογή του κατάλληλου αρχείου δεδομένων για την περιοχή ενδιαφέροντος. Κάθε τετράγωνο στο χάρτη απεικονίζει την περιοχή που καλύπτει ένα αρχείο του SRTM. Όπως φαίνεται από την εικόνα για να καλύψουμε πλήρως την Ελλάδα απαιτείται η επιλογή και επεξεργασία έξι αρχείων και επομένως η υπορουτίνα thanSet πρέπει να κληθεί έξι φορές και να τα αντιστοιχήσει με έξι διαφορετικές μεταβλητές τύπου DEMUSGS. Ωστόσο για να επιτύχουμε ευκολότερη πρόσβαση στα δεδομένα και να αποφευχθούν λάθη, δημιουργείται ένας πίνακας μεταβλητών τύπου DEMUSGS έξι θέσεων με όνομα A. Στην υπορουτίνα INIT καλείται η thanSet έξι φορές και αντιστοιχίζονται τα αρχεία του SRTM και οι αντίστοιχες περιοχές του χάρτη με τις θέσεις του πίνακα A όπως φαίνεται στο σχήμα 3.4.



Σχημα 3.4. θέσεις του πίνακα A στο χάρτη

Έχουμε επομένως έξι πίνακες με τα υψόμετρα των σημείων πάνω στο χάρτη όπως παρουσιάστηκαν παραπάνω. Για να διευκολυνθούμε όμως στη συγγραφή του προγράμματος αντί της χρήσης έξι πινάκων υψομέτρων (έναν αποθηκευμένο σε κάθε μεταβλητή DEMUSGS σε κάθε θέση του πίνακα A) δουλεύουμε σαν να έχουμε ένα μεγάλο πίνακα που περιέχει τα υψόμετρα όλων των σημείων. Επειδή οι διαστάσεις όλων των πινάκων υψομέτρων είναι ίσες και ισούνται με 6001 γραμμές και 6001 στήλες (δηλαδή τα NXCOLS και NYROWS ισούνται με 6001 σε όλες τις θέσεις του πίνακα A) οι διαστάσεις του μεγάλου αυτού πίνακα θα είναι 12002 στήλες και 18003 γραμμές. Ωστόσο ο πίνακας αυτός δεν δηλώνεται και δεν αντιγράφονται τα υψόμετρα των σημείων από τους πίνακες σε αυτόν, καθώς κάτι τέτοιο θα δέσμευε επιπλέον μνήμη χωρίς να υπάρχει ουσιαστικός λόγος, αντίθετα ο πίνακας είναι πλασματικός και αντί αυτού χρησιμοποιείται η συνάρτηση HIM. Στη συνάρτηση HIM εισάγεται η θέση ενός σημείου στον μεγάλο (πλασματικό) πίνακα και αυτή επιστρέφει το υψόμετρο του σημείου από τον μικρό πίνακα όπου αυτό είναι αποθηκευμένο. Η συνάρτηση HIM καλεί την υπορουτίνα ijriiaka, αυτή επιστρέφει σε ποιόν από τους έξι πίνακες ανήκει το σημείο που αναζητάμε καθώς και την θέση του στον πίνακα αυτόν. Στη συνέχεια λαμβάνεται το υψόμετρο που είναι αποθηκευμένο στον πίνακα και την θέση που προέκυψαν και επιστρέφεται από την συνάρτηση HIM. Τέλος, σε περίπτωση εισαγωγής σημείων εκτός του μεγάλου πίνακα, δηλαδή, εκτός του χάρτη (για παράδειγμα σημείο που ανήκει στη γραμμή μηδέν) επιστρέφεται υψόμετρο ίσο με -1. Αυτό που συμβολίζει αυτό το σφάλμα θα χρησιμεύσει παρακάτω στη συγγραφή του προγράμματος.

Με τον τρόπο που περιγράφηκε παραπάνω έχουμε εισάγει τα υψομετρικά δεδομένα για όλη την Ελλάδα ως ένα μεγάλο πίνακα διαστάσεων 12002 στηλών και 18003 γραμμών, δηλαδή 216.072.006 υψόμετρα, με την απόσταση μεταξύ διαδοχικών γραμμών του πίνακα περίπου ίση με 90 μέτρα και μεταξύ διαδοχικών στηλών περίπου ίση με 70 μέτρα.

3.4 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ – ΕΥΡΕΣΗ ΛΙΜΝΟΔΕΞΑΜΕΝΩΝ

Έχοντας στη διάθεση μας τα υψομετρικά στοιχεία για 216.000.000 σημεία που καλύπτουν όλη την Ελλάδα σε μορφή ενός πίνακα διαστάσεων 12002 στηλών και 18003 γραμμών προσπαθούμε να βρούμε πιθανές θέσεις όπου μπορεί να υπάρχουν λιμνοδεξαμενές. Για να το διαπιστώσουμε εξετάζουμε κάθε ένα από αυτά τα σημεία ως κέντρο της πιθανής λιμνοδεξαμενής. Για να είναι όντως κέντρο το εξεταζόμενο σημείο θα πρέπει τα υψόμετρα να αυξάνουν καθώς απομακρυνόμαστε από αυτό. Έτσι θεωρούμε διαδοχικές περιμέτρους γύρω

από το εξεταζόμενο σημείο όπως παρουσιάστηκαν στην παράγραφο 3.2 και απαιτούμε όσο αυξάνει η περίμετρος, δηλαδή όσο απομακρυνόμαστε από το σημείο, τα υψόμετρα να μεγαλώνουν. Συγκεκριμένα όλα τα υψόμετρα της πρώτης περιμέτρου πρέπει να είναι μεγαλύτερα από το υψόμετρο του εξεταζόμενου σημείου, ενώ όλα τα υψόμετρα μια τυχαίας περιμέτρου πρέπει να είναι μεγαλύτερα από όλα τα υψόμετρα της προηγούμενης της.

Για να γίνει ο έλεγχος των σημείων χρησιμοποιείται η υπορουτίνα `perimetroi`. Αυτή αφού κληθεί τυπώνει σε ένα αρχείο τη θέση κάθε σημείου στον πίνακα και το πλήθος των διαδοχικών περιμέτρων γύρω από το σημείο που ικανοποιούν τις προϋποθέσεις που ορίσαμε παραπάνω. Το πλήθος των περιμέτρων μπορεί να χρησιμοποιηθεί για να εκτιμήσουμε το μέγεθος της λιμνοδεξαμενής με κέντρο το εξεταζόμενο σημείο.

Η υπορουτίνα `perimetroi` εξετάζει ένα προς ένα όλα τα σημεία και τυπώνει τα αποτελέσματα για καθένα. Ωστόσο στην περίπτωση που ένα σημείο έχει υψόμετρο μικρότερο των 10 μέτρων ο έλεγχος γύρω από το σημείο αυτό δεν γίνεται διότι θεωρούμε ότι η λιμνοδεξαμενή είναι πολύ κοντά στη θάλασσα για να είναι χρήσιμη. Επίσης με αυτόν τον περιορισμό απορρίπτονται σημεία που βρίσκονται πάνω στη θάλασσα και όπως είναι λογικό δε μπορούν να αποτελέσουν σημεία λιμνοδεξαμενής. Τα σημεία αυτά έχουν υψόμετρο μικρότερο του μηδενός στα δεδομένα του SRTM.

Αρχικά ελέγχονται τα σημεία της πρώτης περιμέτρου γύρω από το εξεταζόμενο σημείο και βρίσκουμε το μέγιστο και το ελάχιστο υψόμετρο αυτών. Επειδή δεν υπάρχουν προηγούμενες περιμέτροι αρκεί το μικρότερο υψόμετρο της περιμέτρου να είναι μεγαλύτερο από το υψόμετρο του εξεταζόμενου σημείου. Αυτό σημαίνει πως όλα τα σημεία της περιμέτρου είναι ψηλότερα από το εξεταζόμενο και άρα ο έλεγχος μπορεί να προχωρήσει σε επόμενες περιμέτρους. Το μέγιστο υψόμετρο χρησιμεύει για τη σύγκριση στην επόμενη περίμετρο. Σε αυτήν αφού βρούμε ξανά το μέγιστο και το ελάχιστο υψόμετρο ελέγχουμε αν το ελάχιστο υψόμετρο είναι μεγαλύτερο από το μέγιστο της προηγούμενης περιμέτρου, πράγμα που σημαίνει ότι όλα τα σημεία της περιμέτρου είναι ψηλότερα από όλα τα σημεία της προηγούμενης περιμέτρου και άρα μπορούμε να προχωρήσουμε στην επόμενη. Η διαδικασία σταματάει όταν το ελάχιστο της περιμέτρου είναι μικρότερο από το μέγιστο της προηγούμενης της ή αν βρεθεί σημείο στην εξεταζόμενη περίμετρο με υψόμετρο μικρότερο των 10 μέτρων, καθώς αυτό δε μπορεί να είναι σημείο λιμνοδεξαμενής. Τέλος, αφού η διαδικασία διακοπεί για το εξεταζόμενο σημείο τυπώνεται το πλήθος των αποδεκτών περιμέτρων που αντιστοιχούν σε αυτό.

3.5 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΥ

Η εκτέλεση του προγράμματος, το οποίο χρησιμοποιεί τις υπορουτίνες που αναφέρθηκαν, δίνει ένα αρχείο αποτελεσμάτων που περιέχει το πλήθος των περιμέτρων που ικανοποιούν την πρώτη μεθοδολογία, όπως αναλύθηκε, γύρω από κάθε σημείο. Από τα αποτελέσματα αυτά μπορούμε να βγάλουμε συμπεράσματα για την πιθανότητα ύπαρξης λιμνοδεξαμενής, συσχετίζοντας την έκταση που αυτή θα κάλυπτε με το πλήθος των περιμέτρων. Προκύπτει ότι από τα 216.000.000 σημεία που εξετάστηκαν, 25 σημεία βρέθηκαν να έχουν μόλις δυο περιμέτρους με υψόμετρα να αυξάνουν σύμφωνα με τη μεθοδολογία 1. Το πλήθος των δυο περιμέτρων αντιστοιχεί προσεγγιστικά σε έκταση 100.800 τετραγωνικών μέτρων, όπως περιγράφηκε στην παράγραφο 3.2. Ακόμα βρέθηκε ένα σημείο με τρεις περιμέτρους που αντιστοιχούν προσεγγιστικά σε έκταση 226.800 τετραγωνικών μέτρων. Τέλος, σημεία με πλήθος περιμέτρων μεγαλύτερο των τριών δεν εντοπίστηκαν.

Οι λιμνοδεξαμενές που υπολογίστηκαν από το πρόγραμμα `limnodeksameni1` αντιστοιχούν σε μικρές εκτάσεις και αυτό οδηγεί στην απαίτηση για μεγάλο βάθος προκειμένου να εξασφαλίσουμε τον απαιτούμενο όγκο ύδατος στη λιμνοδεξαμενή, που είναι περίπου 4.000.000 κυβικά μέτρα. Απαιτείται επομένως ένας έστω προσεγγιστικός υπολογισμός του όγκου νερού για κάθε θέση που υπολογίστηκε για να διαπιστώσουμε αν μπορούμε να τη χρησιμοποιήσουμε. Αυτό θα γίνει σε επόμενο στάδιο λόγω του μικρού πλήθους των αποτελεσμάτων.

Τα αποτελέσματα βρίσκονται στο παράρτημα A.1.1 όπου σε κάθε γραμμή παρουσιάζεται η θέση του κεντρικού σημείου στον πίνακα και πιο συγκεκριμένα η γραμμή και η στήλη στην οποία αυτό ανήκει και το πλήθος των αποδεκτών περιμέτρων γύρω από το σημείο.

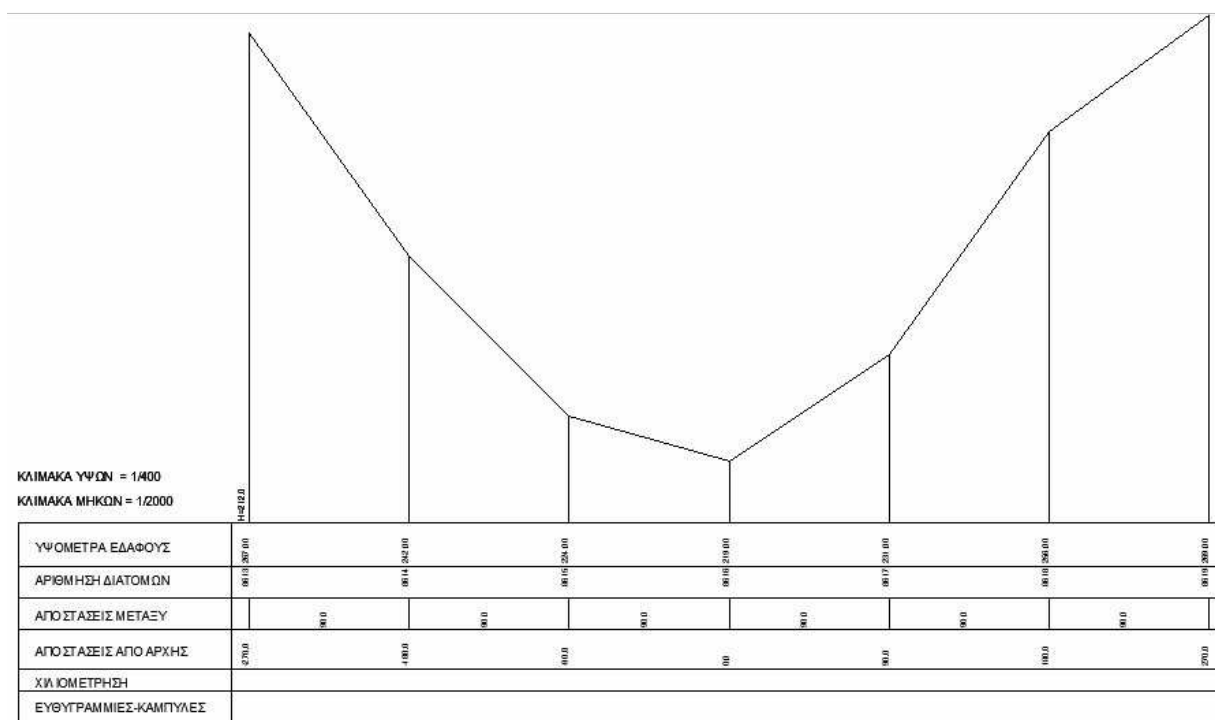
3.6 ΑΠΕΙΚΟΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Για την καλύτερη ανάλυση των αποτελεσμάτων της μεθόδου 1, καθώς και για να διαπιστωθεί η ορθότητά τους, προχωράμε στην απεικόνισή τους με τη δημιουργία μηκοτομών. Για να το επιτύχουμε χρησιμοποιούμε την υπορουτίνα `mikotomi` και το πρόγραμμα `c_mhk`, που περιγράφονται παρακάτω, για τη δημιουργία μιας μηκοτομής η οποία θα διέρχεται από το σημείο που εξετάζεται ως κέντρο της λιμνοδεξαμενής.

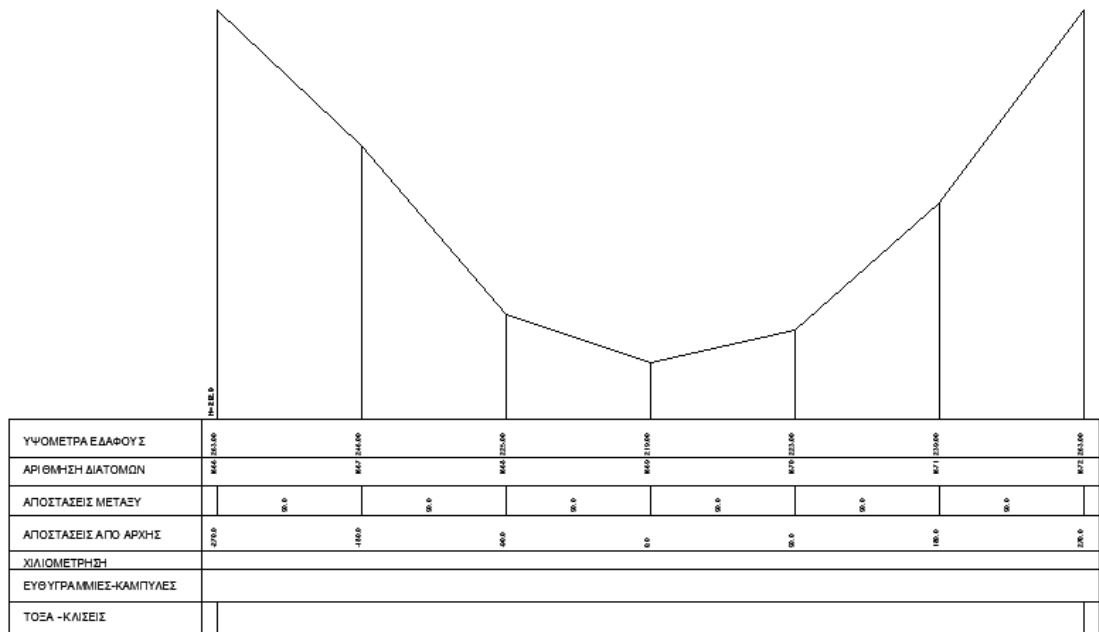
Στην υπορουτίνα mikotomi εισάγεται η θέση του κέντρου της λιμνοδεξαμενής στον πίνακα, το πλήθος των περιμέτρων γύρω από το κέντρο, των οποίων σημεία θα σχεδιαστούν στη μηκοτομή και ένας κωδικός αριθμός που ορίζει αν η μηκοτομή θα περιέχει σημεία της ίδιας γραμμής ή της ίδιας στήλης με το κεντρικό. Η υπορουτίνα δημιουργεί ένα αρχείο και σε αυτό να τυπώνει την χιλιομετρική θέση, το υψόμετρο και έναν χαρακτηριστικό αριθμό για κάθε σημείο που θα σχεδιαστεί στη μηκοτομή. Γνωρίζοντας ότι η απόσταση μεταξύ διαδοχικών σημείων της ίδιας γραμμής και της ίδιας στήλης είναι 70 και 90 μέτρα αντίστοιχα καθώς και το πλήθος των περιμέτρων των οποίων τα σημεία θα τυπωθούν, δίνουμε στο πρώτο σημείο χιλιομετρική θέση -70N ή -90N, όπου N το πλήθος των περιμέτρων και σε κάθε επόμενο προσθέτουμε 70 ή 90 μέτρα αντίστοιχα. Έτσι το κεντρικό σημείο έχει χιλιομετρική θέση μηδέν και μπορεί να εντοπίζεται εύκολα.

Το αρχείο που έχει δημιουργηθεί από την υπορουτίνα perimetroι χρησιμοποιείται στη συνέχεια από το πρόγραμμα c_mhk το οποίο μας δίνει τη μηκοτομή σε ένα αρχείο dxf το οποίο μπορούμε να ανοίξουμε με οποιοδήποτε σχεδιαστικό πρόγραμμα (CAD) ηλεκτρονικού υπολογιστή.

Στα σχήματα 3.5 και 3.6 παρουσιάζονται δυο μηκοτομές, η πρώτη στη ίδια γραμμή και η δεύτερη στην ίδια στήλη με το σημείο που προέκυψε στα αποτελέσματα της μεθόδου να έχει 3 περιμέτρους.



Σχήμα 3.5. μηκοτομή αποτελέσματος της μεθοδολογίας 1



Σχήμα 3.6. μηκοτομή αποτελέσματος της μεθοδολογίας 1

Παρατηρούμε ότι τόσο στη μηκοτομή με τα σημεία της ίδιας γραμμής, όσο και σε αυτή με τα σημεία της ίδιας στήλης με το κεντρικό, τα υψόμετρα αυξάνουν καθώς απομακρυνόμαστε από το κεντρικό σημείο, το οποίο είναι το πέμπτο κατά σειρά με χιλιομετρική θέση 0.

Περισσότερες μηκοτομές θέσεων λιμνοδεξαμενών που υπολογίστηκαν από το πρόγραμμα `limnodeksamen1` βρίσκονται στο παράρτημα A.1.2. Σε όλες η μορφή είναι όπως στο παράδειγμα που παρουσιάστηκε όπως άλλωστε αναμενόταν. Συμπεραίνουμε λοιπόν ότι τα αποτελέσματα της μεθόδου που εξετάστηκε στο κεφάλαιο αυτό, είναι ορθά, παρά το περιορισμένο πλήθος τους, αφού αντιστοιχούν στη μορφολογία εδάφους που περιγράφηκε.

4. ΜΕΘΟΔΟΛΟΓΙΑ 2

4.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό αναπτύσσεται η δεύτερη μεθοδολογία για την εύρεση θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας με χρήση των υψομετρικών δεδομένων που έχουμε στη διάθεσή μας από το παγκόσμιο ψηφιακό μοντέλο SRTM. Στη δεύτερη μέθοδο δεχόμαστε ορισμένα σημεία μακριά από το κέντρο της (πιθανής) λιμνοδεξαμενής να έχουν μικρότερο υψόμετρο από σημεία που είναι πιο κοντά στο κέντρο.

4.2 ΜΕΘΟΔΟΣ

Με την πρώτη μεθοδολογία γίνεται αναζήτηση για φυσικές κοιλότητες στις οποίες το υψόμετρο αυξάνει μονότονα από το μέσο της κοιλότητας προς την περίμετρό της. Όπως είδαμε τα αποτελέσματα της μεθόδου αυτής ήταν περιορισμένα. Με τη δεύτερη μεθοδολογία προσπαθούμε να αυξήσουμε το πλήθος των αποτελεσμάτων και να εντοπίσουμε περισσότερες μορφές εδάφους που μπορούν να αντιστοιχούν σε λιμνοδεξαμενή.

Στη δεύτερη μεθοδολογία αναζητούμε φυσικές κοιλότητες που μπορούν να διαμορφωθούν χωρίς την κατασκευή φράγματος. Αναζητούμε κοιλότητες στις οποίες τα υψόμετρα δεν είναι απαραίτητα μονότονα αυξανόμενα, αντίθετα μπορούν σε μια περιοχή να παραμένουν σταθερά ή ακόμα και να μειώνονται, αρκεί στη συνέχεια να συνεχίσουν να αυξάνουν. Δεχόμαστε επομένως ένα πλήθος περιμέτρων να έχουν σημεία με μικρότερα υψόμετρα, αρκεί να εντοπιστεί επόμενη, πιο απομακρυσμένη περίμετρος της οποίας τα σημεία έχουν μεγαλύτερο υψόμετρο από όλα τα εσωτερικά της σημεία.

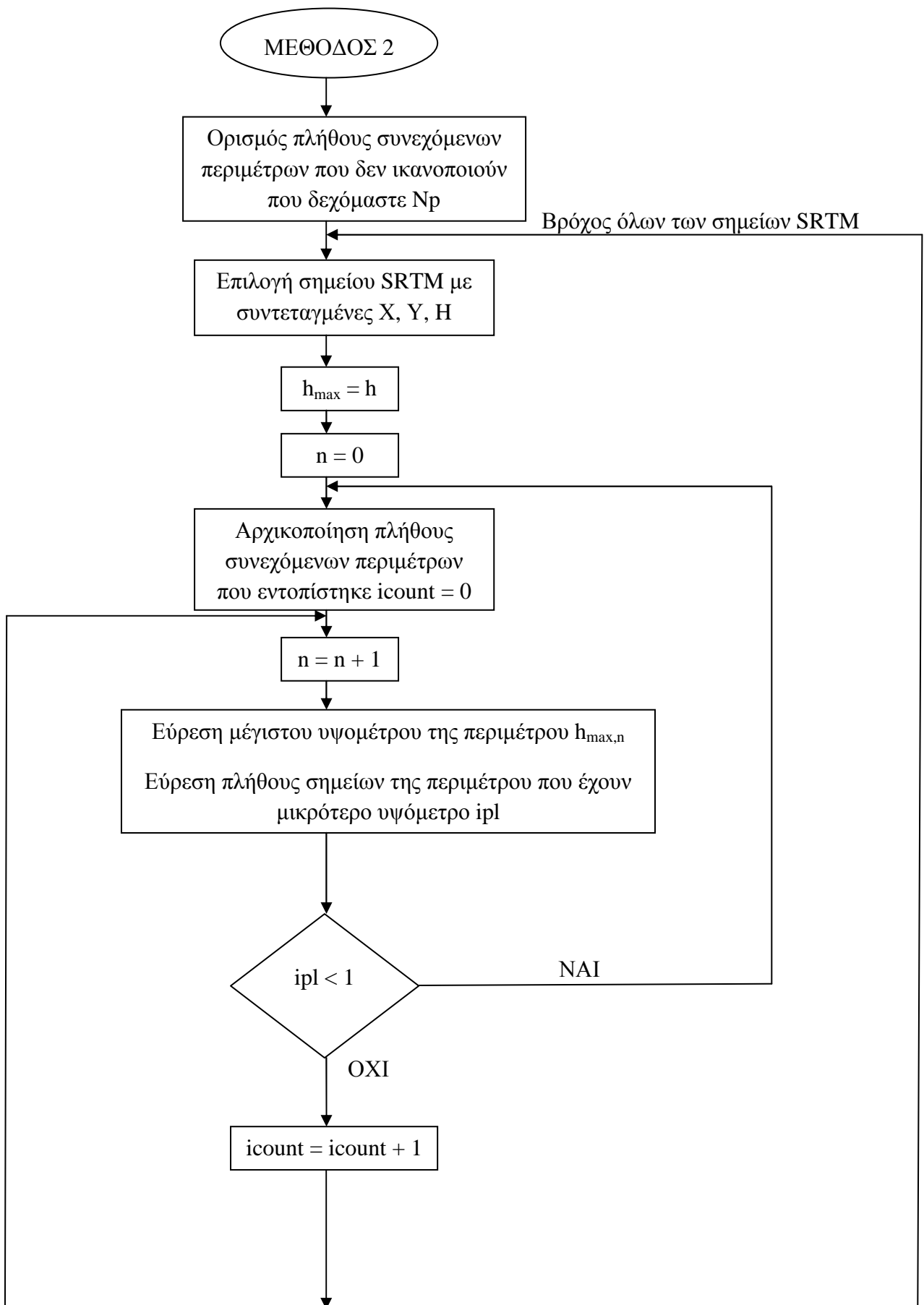
Προκειμένου να βρούμε τέτοιες περιοχές ξεκινάμε από τυχαίο σημείο του κανάβου και ελέγχουμε το υψόμετρο των σημείων της πρώτης περιμέτρου. Αν δε βρεθεί κανένα σημείο να έχει μικρότερο υψόμετρο από το αρχικό τότε το αρχικό σημείο είναι κέντρο τετραγωνικής κοιλότητας 1ης περιμέτρου.

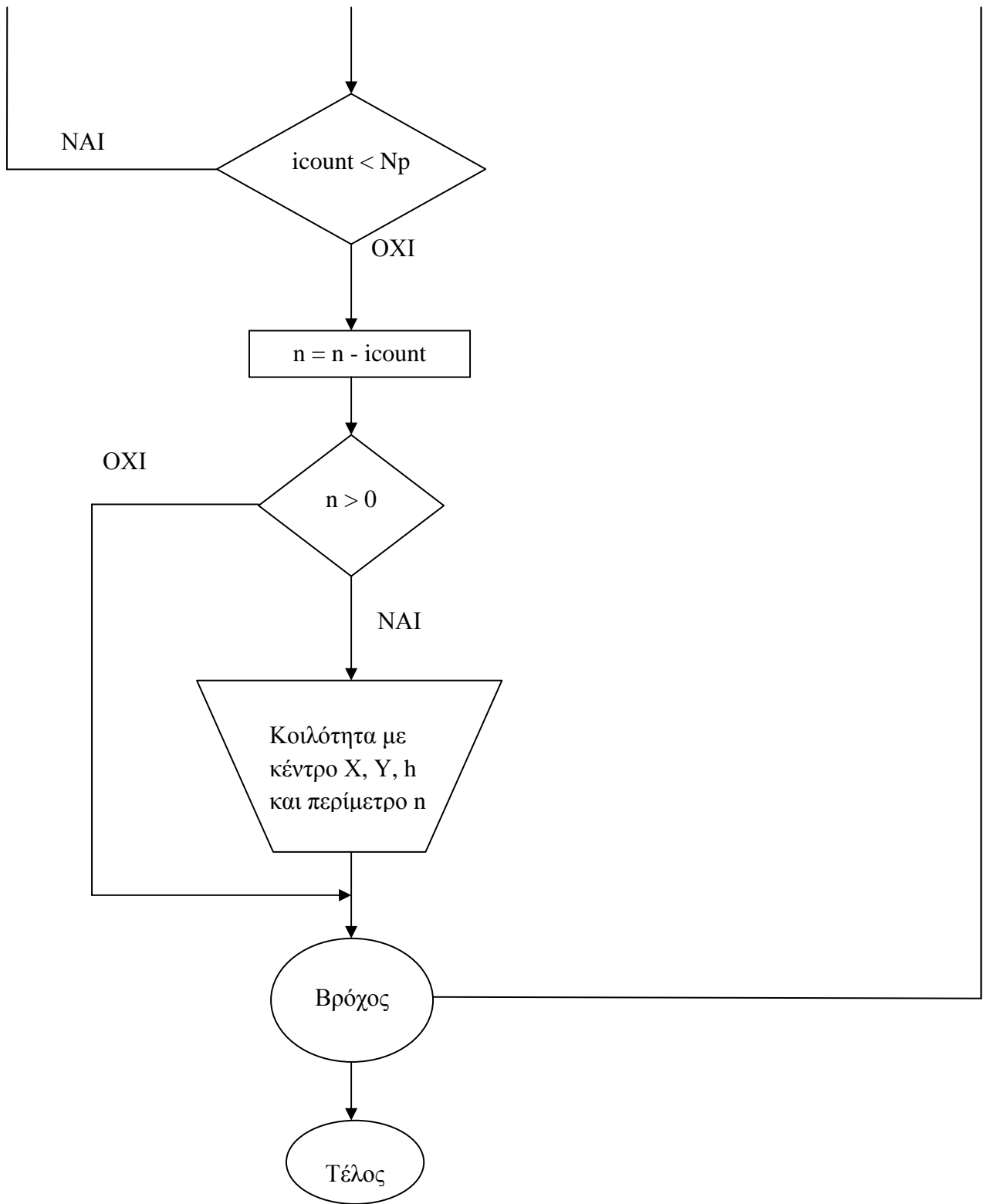
Στη συνέχεια ελέγχονται τα υψόμετρα της δεύτερης περιμέτρου. Αν το πλήθος των σημείων με μικρότερο υψόμετρο από αυτό όλων των σημείων της πρώτης περιμέτρου ισούται με μηδέν, τότε το κεντρικό σημείο είναι κοιλότητα με περίμετρο τη 2η. Η ίδια διαδικασία επαναλαμβάνεται για την 3, 4, 5 κλπ περίμετρο.

Για να βρούμε αν υπάρχει σημείο μιας περιμέτρου που έχει μικρότερο υψόμετρο από τα σημεία των προηγούμενων περιμέτρων, συγκρίνουμε το υψόμετρο κάθε σημείου της περιμέτρου με το μέγιστο υψόμετρο των προηγούμενων περιμέτρων.

Σε αυτή τη μέθοδο δεχόμαστε κάποιο υψόμετρο μιας περιμέτρου να είναι μικρότερο από της προηγούμενη, με την προϋπόθεση ότι όλα τα σημεία της επόμενης περιμέτρου θα έχουν μεγαλύτερο υψόμετρο από όλες τα προηγούμενες περιμέτρους που περικλείονται σε αυτή. Η πρόταση αυτή γενικεύεται αν δεχτούμε N_p διαδοχικές περιμέτρους με μικρότερο υψόμετρο από προηγούμενες, με την προϋπόθεση ότι η περίμετρος N_{p+1} έχει μεγαλύτερο υψόμετρο από όλες τις προηγούμενες. Αν συναντήσουμε περισσότερες από N_p διαδοχικές περιμέτρους με μικρότερα υψόμετρα η διαδικασία σταματάει. Το πλήθος N_p είναι μια παράμετρος που μπορεί να μεταβληθεί ανάλογα τις απαιτήσεις της διερεύνησης. Ο αλγόριθμος συνοψίζεται στο σχήμα 4.1.

Για τη δεύτερη μέθοδο κατασκευάστηκε πρόγραμμα Η/Υ (limnodeksameni2) σε γλώσσα FORTRAN που εφαρμόζει τον παραπάνω αλγόριθμο και υπολογίζει πιθανές θέσεις λιμνοδεξαμενών. Ο κώδικας του προγράμματος δίνεται στο παράρτημα Β.2.





Σχήμα 4.1. Αλγόριθμος της δεύτερης μεθοδολογίας

4.3 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ – ΕΥΡΕΣΗ ΛΙΜΝΟΔΕΞΑΜΕΝΩΝ

Η εισαγωγή των υψομετρικών δεδομένων των σημείων γίνεται με τον ίδιο τρόπο, όπως στο πρόγραμμα `limnodeksameni1`, όπως περιγράφηκε στην παράγραφο 3.3. Χρησιμοποιούμε δηλαδή και εδώ την συνάρτηση `him` για να πάρουμε το υψόμετρο οποιουδήποτε σημείου δίνοντας τη θέση του στο μεγάλο πίνακα.

Για τον έλεγχο των σημείων χρησιμοποιούμε την υπορουτίνα `perimetroi`, η οποία λειτουργεί διαφορετικά απ'ότι στο πρόγραμμα `limnodeksameni1`, καθώς δέχεται δυο μεταβλητές, μια για κάθε πρόσθετη λειτουργία της.

Η μεταβλητή που δίνουμε στην υπορουτίνα αναφέρεται στον μέγιστο αριθμό των διαδοχικών περιμέτρων που δύνανται να έχουν σημεία με μικρότερο υψόμετρο και έχει όνομα `iper`.

Η υπορουτίνα `perimetroi` ελέγχει κάθε σημείο του πίνακα με υψόμετρο μεγαλύτερο των δέκα μέτρων και εξετάζει διαδοχικές περιμέτρους γύρω από αυτό σύμφωνα με τον αλγόριθμο που περιγράφηκε στην προηγούμενη παράγραφο. Για κάθε μια αρχικά καλεί την υπορουτίνα `find_pl`, στην οποία δίνει την θέση του κεντρικού σημείου που ελέγχεται, τον αριθμό της περιμέτρου γύρω από το σημείο που εξετάζουμε και το μέγιστο υψόμετρο που έχει προκύψει από τις προηγούμενες περιμέτρους. Αυτή με τη σειρά της καλεί την υπορουτίνα `points_pos` η οποία επιστρέφει τις θέσεις στον πίνακα των σημείων της περιμέτρου που εξετάζεται. Στη συνέχεια συγκρίνοντας τα υψόμετρα των σημείων της περιμέτρου με το μέγιστο των προηγούμενων περιμέτρων επιστρέφει το πλήθος των σημείων που έχουν μικρότερο υψόμετρο, τη θέση τους καθώς και το μέγιστο υψόμετρο που προκύπτει από την εξεταζόμενη περίμετρο.

Στη συνέχεια η υπορουτίνα `perimetroi` χρησιμοποιεί την συνάρτηση `check_per` για να ελέγξει αν η εξεταζόμενη περίμετρος ικανοποιεί ή όχι τις προϋποθέσεις της δεύτερης μεθοδολογίας, αν δηλαδή υπάρχει σε αυτήν σημείο με μικρότερο υψόμετρο.

Στην περίπτωση που μια περίμετρος δεν ικανοποιεί τις προϋποθέσεις η διαδικασία δεν τερματίζεται αμέσως όπως συνέβαινε στο πρόγραμμα `limnodeksameni1`, αλλά συνεχίζεται με τις επόμενες περιμέτρους μέχρι το πλήθος των συνεχόμενων περιμέτρων που δεν ικανοποιούν τις προϋποθέσεις να ξεπεράσει το μέγιστο που ορίζεται με τη μεταβλητή `istor`.

Αν όλα τα σημεία της περιμέτρου έχουν μεγαλύτερο υψόμετρο από τις προηγούμενες περιμέτρους η διαδικασία συνεχίζεται κατά τα γνωστά με την επόμενη περίμετρο, ενώ

επιπλέον μηδενίζεται το πλήθος των συνεχόμενων περιμέτρων που δεν ικανοποιούν τις προϋποθέσεις (το οποίο αρχίζει να αυξάνει ξανά για κάθε επόμενη περίμετρο που δεν ικανοποιεί τις προϋποθέσεις).

4.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΥ

Για να διερευνήσουμε τη μέθοδο έγιναν πολλές εκτελέσεις του προγράμματος limnodeksameni2 δίνοντας διάφορες τιμές, κατά την κλήση της υπορουτίνας perimetroί, στην παράμετρο που ορίζουν το μέγιστο πλήθος των διαδοχικών περιμέτρων που δεν ικανοποιούν τις προϋποθέσεις.

Τα αποτελέσματα που παρουσιάζονται προέκυψαν εκτελώντας το πρόγραμμα με την τιμή 5 για την μεταβλητή που αναφέρεται στο μέγιστο πλήθος των διαδοχικών περιμέτρων που δεν ικανοποιούν που πρέπει να εντοπιστούν για να σταματήσει ο έλεγχος, καθώς μετά από δοκιμές καταλήξαμε ότι για τιμή μεγαλύτερη από αυτήν δεν υπάρχουν διαφορές στα αποτελέσματα, ενώ αυξάνει πολύ ο υπολογιστικός χρόνος του προγράμματος.

Εντοπίστηκαν 76 σημεία με δύο περιμέτρους και μόλις ένα σημεία με 3 περιμέτρους, ενώ όπως ήταν αναμενόμενο το σημείο αυτό είναι το ίδιο που είχε προκύψει από τη μεθοδολογία 1. Παρατηρούμε ότι εντοπίστηκαν επιπλέον 51 σημεία με 2 περιμέτρους να ικανοποιούν σε σχέση με την πρώτη μεθοδολογία και περιμένουμε σε αυτές να υπάρχουν σημεία της πρώτης περιμέτρου που έχουν υψόμετρο μικρότερο από το κεντρικό σημείο. Αυτός είναι ο λόγος που δεν εντοπίστηκαν με την πρώτη μέθοδο στην οποία ο έλεγχος σταμάτησε εκεί. Ακόμα, όλα τα σημεία της δεύτερης περιμέτρου έχουν μεγαλύτερο υψόμετρο από την πρώτη περίμετρο και το κεντρικό σημείο και έτσι τα σημεία αυτά εντοπίστηκαν να έχουν δυο περιμέτρους να ικανοποιούν τις προϋποθέσεις της δεύτερης μεθόδου.

Ακόμα παρατηρούμε ότι δεν εντοπίστηκαν επιπλέον σημεία με περισσότερες από 2 περιμέτρους να ικανοποιούν τη δεύτερη μεθοδολογία, και αυτός είναι ο λόγος που τα αποτελέσματα δεν αλλάζουν για τιμές μεγαλύτερες της μονάδας στο πλήθος των διαδοχικών περιμέτρων που δύνανται να έχουν σημεία με μικρότερο υψόμετρο. Αυτό συμβαίνει γιατί δεν εντοπίζονται επόμενες περίμετροι των οποίων όλα τα σημεία έχουν μεγαλύτερο υψόμετρο από όλα τα σημεία που περικλείονται από αυτές.

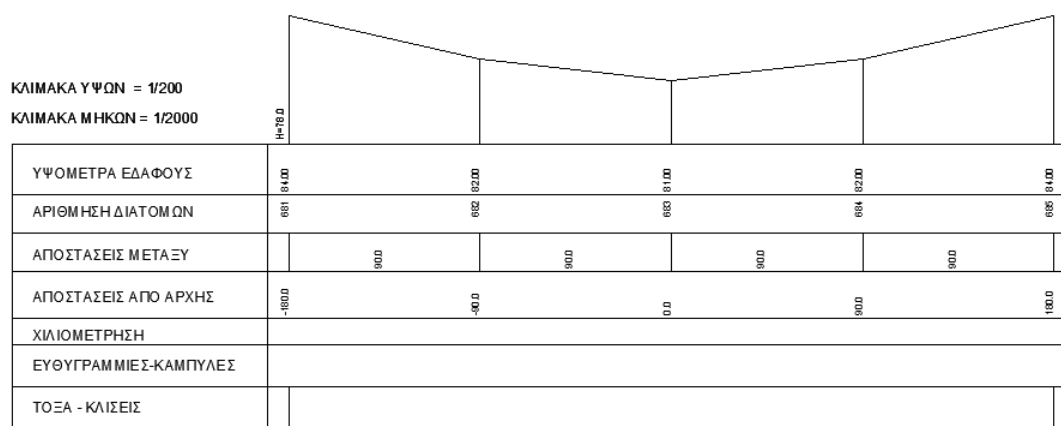
Τα αποτελέσματα βρίσκονται στο παράρτημα Α.2.1 όπου σε κάθε γραμμή παρουσιάζονται η γραμμή και η στήλη του σημείου στον πίνακα και το πλήθος των περιμέτρων γύρω από το σημείο που ικανοποιούν τη δεύτερη μεθοδολογία.

4.5 ΑΠΕΙΚΟΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

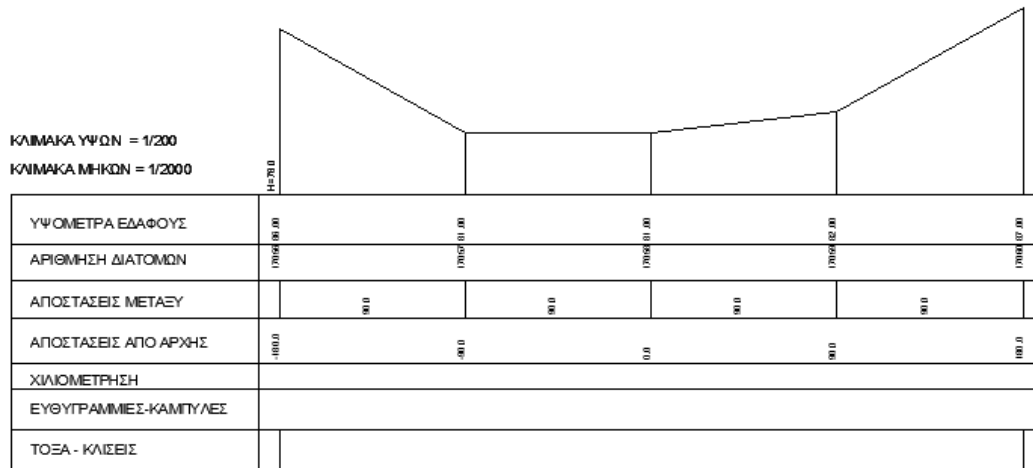
Όπως στη μεθοδολογία1 έτσι και εδώ προχωράμε στην απεικόνιση των αποτελεσμάτων για να εξακριβώσουμε την ορθότητά τους αλλά και για να διερευνήσουμε τους λόγους που οδηγούν σε περιορισμένο αριθμό αποτελεσμάτων.

Χρησιμοποιούμε πάλι την υπορουτίνα mikotomi και το πρόγραμμα c_mhk και σχεδιάζουμε τις μηκοτομές των σημείων της ίδιας γραμμής και της ίδιας στήλης με το κεντρικό σημείο, με την διαδικασία που περιγράφηκε στην ενότητα 3.6.

Στα σχήματα 4.2 και 4.3 παρουσιάζονται δυο τέτοιες μηκοτομές με κέντρο ένα σημείο που προέκυψε στα αποτελέσματα της μεθόδου να έχει 2 περιμέτρους.



σχήμα 4.2. μηκοτομή αποτελέσματος της μεθοδολογίας 2



σχήμα 4.3. μηκοτομή αποτελέσματος της μεθοδολογίας 2

Στο σχήμα 4.2 παρουσιάζεται η μηκοτομή που περιλαμβάνει σημεία της ίδιας γραμμής. Το κεντρικό μας σημείο είναι το τρίτο και παρατηρούμε ότι εκατέρωθεν του σημείου αυτού τα υψόμετρα αυξάνουν καθώς απομακρυνόμαστε από αυτό. Στο σχήμα 4.2 έχουμε τη μηκοτομή όπου περιλαμβάνονται σημεία της ίδιας στήλης. Παρατηρούμε ότι καθώς κινούμαστε προς τη μια πλευρά τα υψόμετρα αυξάνουν συνεχώς, όπως θα περιμέναμε, προς την άλλη ωστόσο αρχικά μειώνονται για μια περίμετρο και στη συνέχεια συνεχίζουν να αυξάνουν. Αυτός είναι και ο λόγος που το σημείο αυτό δεν έχει δυο περιμέτρους να ικανοποιούν σύμφωνα με την πρώτη μεθοδολογία, αφού σύμφωνα με αυτή η διαδικασία θα σταματήσει να ελέγχει μόλις διαπιστωθεί ότι η πρώτη περίμετρος δεν ικανοποιεί τις προϋποθέσεις. Αντίθετα στη δεύτερη μεθοδολογία ο έλεγχος συνεχίζεται και αφού διαπιστώνεται ότι η επόμενη περίμετρος έχει μεγαλύτερα υψόμετρα, όπως φαίνεται και στη μηκοτομή γίνεται δεκτή και η προηγούμενη.

Περισσότερες μηκοτομές θέσεων λιμνοδεξαμενών που υπολογίστηκαν με το πρόγραμμα limnodeksameni2 που εφαρμόζει τον αλγόριθμο της δεύτερης μεθόδου βρίσκονται στο παράρτημα Α.2.2. Σε όλες η μορφή είναι όπως στο παράδειγμα που παρουσιάστηκε όπως άλλωστε αναμενόταν. Συμπεραίνουμε λοιπόν ότι τα αποτελέσματα της μεθόδου που εξετάστηκε στο κεφάλαιο αυτό, είναι ορθά, παρά το περιορισμένο πλήθος τους, αφού αντιστοιχούν στη μορφολογία εδάφους που περιγράφηκε.

5. ΜΕΘΟΔΟΛΟΓΙΑ 3

5.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό αναπτύσσεται η τρίτη μεθοδολογία για την εύρεση θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας με χρήση των υψομετρικών δεδομένων που έχουμε στη διάθεσή μας από το παγκόσμιο ψηφιακό μοντέλο SRTM. Στην τρίτη μέθοδο αναζητούμε κοιλάτητες οι οποίες μπορούν να γίνουν λιμνοδεξαμενές με την κατασκευή φράγματος.

5.2 ΜΕΘΟΔΟΣ

Με τις δυο προηγούμενες μεθόδους αναζητάμε φυσικές κοιλάτητες με τα υψόμετρα να αυξάνουν από το μέσο προς την περίμετρό τους. Τα αποτελέσματα της δεύτερης μεθόδου είναι περισσότερα, ενώ σε αυτά περιλαμβάνονται και τα αποτελέσματα της πρώτης μεθόδου. Με την τρίτη μέθοδο προσπαθούμε να υπολογίσουμε περισσότερες θέσεις λιμνοδεξαμενών. Έτσι προσπαθούμε να εντοπίσουμε διαφορετικές κοιλάτητες που μπορούν να γίνουν λιμνοδεξαμενές.

Στην τρίτη μεθοδολογία αναζητούμε ακόμα κοιλάτητες, οι οποίες μπορούν να γίνουν λιμνοδεξαμενές με την κατασκευή φράγματος. Το φράγμα που απαιτείται θα πρέπει να έχει μικρό μήκος, καθώς σκοπός μας είναι ο υπολογισμός θέσεων λιμνοδεξαμενών χαμηλού κόστους. Αναζητούνται κοιλάτητες στις οποίες δεν απαιτείται όλα τα σημεία της περιμέτρου να έχουν μεγαλύτερο υψόμετρο από τα εσωτερικά της σημεία. Δεχόμαστε ένας αριθμός σημείων μιας περιμέτρου να έχουν μικρότερο υψόμετρο. Το σκεπτικό είναι ότι αν μόνο ένα μικρό τμήμα μιας περιμέτρου δεν έχει το απαιτούμενο υψόμετρο μπορεί να το αποκτήσει με την κατασκευή φράγματος στο συγκεκριμένο τμήμα. Τότε όλα τα σημεία της περιμέτρου θα έχουν το απαιτούμενο υψόμετρο και άρα θα έχουμε μια κοιλάτη που μπορεί να γίνει λιμνοδεξαμενή. Τα σημεία που δεν έχουν το απαραίτητο υψόμετρο θα πρέπει να είναι σε συνεχόμενες θέσεις της περιμέτρου προκειμένου να δημιουργηθεί φράγμα που να τα ενώνει.

Στην τρίτη μεθοδολογία αναζητάμε κοιλάτητες όπως αυτές που αναζητάμε με την δεύτερη μεθοδολογία, όπου τα υψόμετρα μπορούν σε μια περιοχή να παραμένουν σταθερά ακόμα και να μειώνονται, αρκεί στη συνέχεια να συνεχίσουν να αυξάνουν. Δεχόμαστε ένα πλήθος περιμέτρων να έχουν σημεία με μικρότερο υψόμετρο, αρκεί να εντοπιστεί επόμενη

περίμετρος της οποίας όλα τα σημεία έχουν μεγαλύτερο υψόμετρο από όλα τα εσωτερικά της σημεία. Επιπλέον λόγω της δημιουργία φράγματος, δεχόμαστε ένα πλήθος σημείων της πλέον απομακρυσμένης περιμέτρου να έχουν μικρότερο υψόμετρο όπως περιγράφηκε προηγουμένως.

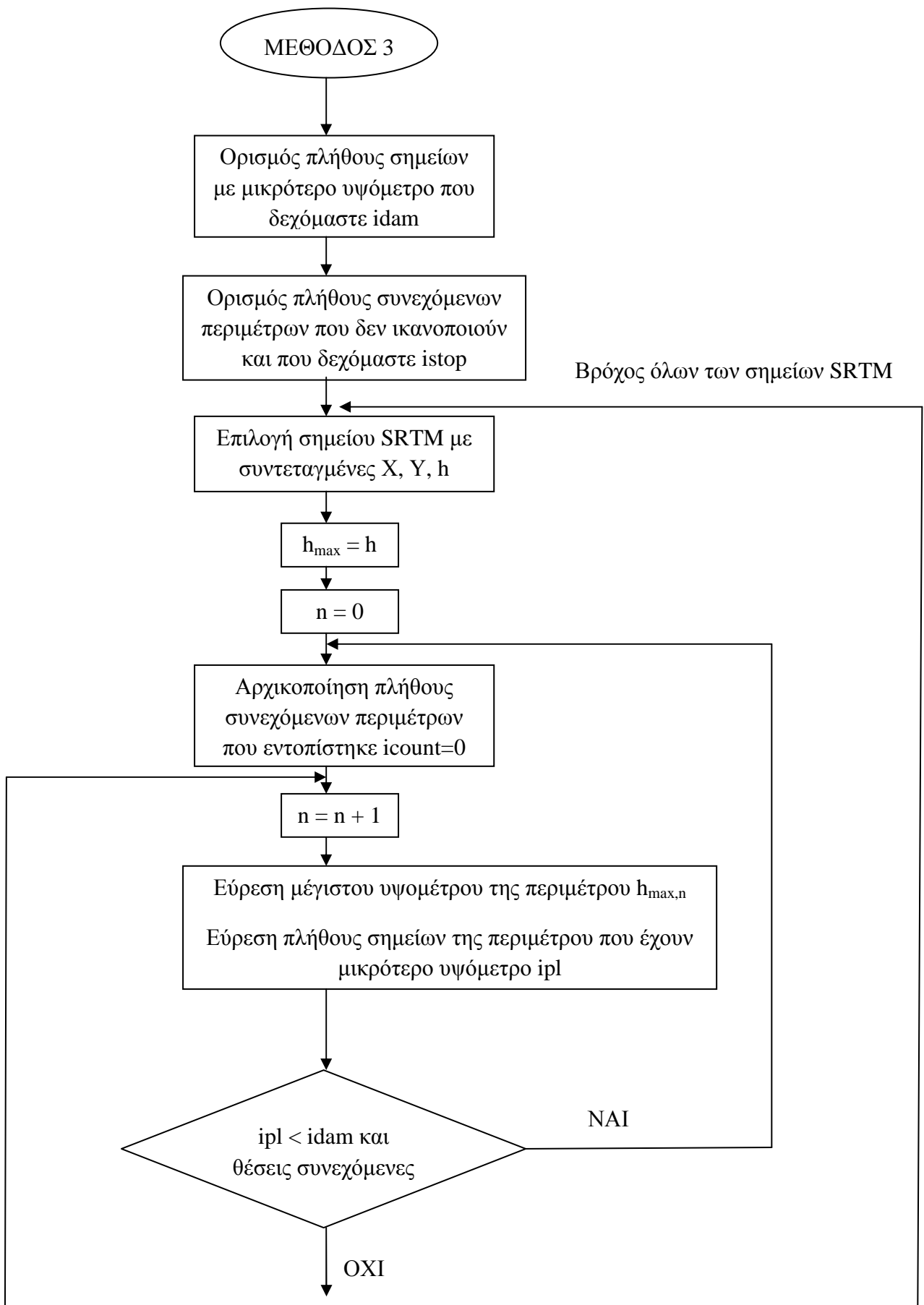
Προκειμένου να βρούμε τέτοιες περιοχές ξεκινάμε από τυχαίο σημείο του κανάβου και ελέγχουμε το υψόμετρο των σημείων της πρώτης περιμέτρου. Βρίσκουμε το πλήθος των σημείων που έχουν μικρότερο υψόμετρο από το αρχικό και αν το πλήθος αυτό είναι μικρότερο από αυτό που δεχόμαστε και οι θέσεις τους είναι γειτονικές το αρχικό σημείο είναι κέντρο τετραγωνικής κοιλότητας 1ης περιμέτρου.

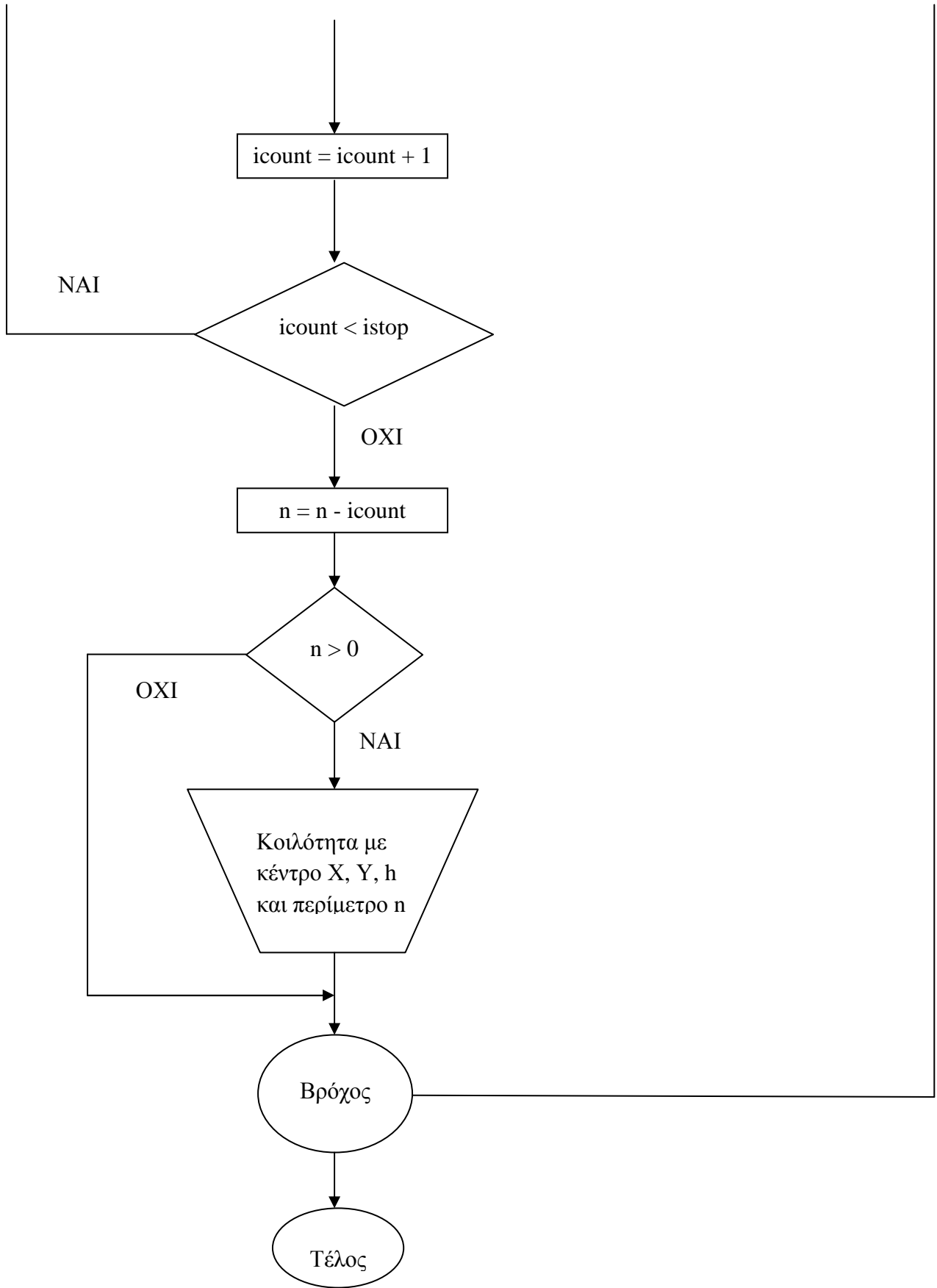
Στη συνέχεια εξετάζονται τα υψόμετρα της δεύτερης περιμέτρου. Αν το πλήθος των σημείων με μικρότερο υψόμετρο από αυτό όλων των σημείων της πρώτης περιμέτρου είναι μικρότερο από αυτό που δεχόμαστε, τότε το κεντρικό σημείο είναι κοιλότητα με περίμετρο τη 2η. Η ίδια διαδικασία επαναλαμβάνεται για την 3, 4, 5 κλπ περίμετρο.

Για να βρούμε το πλήθος των σημείων μιας περιμέτρου που έχουν μικρότερο υψόμετρο από τα σημεία των προηγούμενων περιμέτρων, συγκρίνουμε το υψόμετρο κάθε σημείου της περιμέτρου με το μέγιστο υψόμετρο των προηγούμενων περιμέτρων.

Όπως στη δεύτερη μέθοδο έτσι και εδώ δεχόμαστε N_p περιμέτρους με μικρότερο υψόμετρο από προηγούμενες, με την προϋπόθεση ότι όλα τα σημεία της περιμέτρου N_{p+1} να έχουν μεγαλύτερο υψόμετρο από όλες τις προηγούμενες περιμέτρους. Επιπλέον σε αυτή τη μέθοδο δεχόμαστε ένα πλήθος σημείων μιας περιμέτρου να μην έχουν το απαιτούμενο υψόμετρο ($idam$), αρκεί τα σημεία αυτά να βρίσκονται σε γειτονικές θέσεις της περιμέτρου. Το πλήθος $idam$ καθώς και το N_p είναι μια παράμετροι που μπορεί να μεταβάλλονται ανάλογα με τις απαιτήσεις της διερεύνησης. Ο αλγόριθμος συνοψίζεται στο σχήμα 5.1.

Όπως παρατηρούμε ο αλγόριθμος της τρίτης μεθοδολογίας έχει ελάχιστες διαφορές με αυτόν της δεύτερης. Για το λόγο αυτό το πρόγραμμα `limnodeksameni2` εφαρμόζει και τον αλγόριθμο της τρίτης μεθόδου. Ο κώδικας του προγράμματος δίνεται στο παράρτημα Β.2.





5.3 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ – ΕΥΡΕΣΗ ΛΙΜΝΟΔΕΞΑΜΕΝΩΝ

Η επεξεργασία των δεδομένων γίνεται με παρόμοιο τρόπο όπως αναλύθηκε στη δεύτερη μέθοδο (παράγραφος 4.3), καθώς οι αλγόριθμοι και των δυο μεθόδων εφαρμόζονται από το ίδιο πρόγραμμα (limnodeksameni2). Για την τρίτη μέθοδο η υπορουτίνα perimetroί που ελέγχει όλα τα σημεία ως κέντρα πιθανής λιμνοδεξαμενής δέχεται μια πρόσθετη μεταβλητή σε σχέση με τη δεύτερη μέθοδο.

Η πρόσθετη μεταβλητή που δίνουμε στην υπορουτίνα perimetroί ορίζει το πλήθος των σημείων μιας περιμέτρου που δύνανται να έχουν μικρότερο υψόμετρο και έχει το όνομα idam. Ας σημειωθεί ότι η μεταβλητή αυτή κατά την εκτέλεση του προγράμματος limnodeksameni2 για τον υπολογισμό θέσεων λιμνοδεξαμενών σύμφωνα με τη δεύτερη μέθοδο παίρνει την τιμή μηδέν, καθώς δεν δεχόμαστε κανένα σημείο μιας περιμέτρου να έχει μικρότερο υψόμετρο από τα σημεία που περικλείονται από αυτήν.

Η υπορουτίνα perimetroί δέχεται ακόμα τη μεταβλητή που ορίζει τον μέγιστο αριθμό διαδοχικών περιμέτρων που δύνανται να έχουν μικρότερο υψόμετρο και έχει όνομα istor, όπως περιγράφηκε στην παράγραφο 4.3.

Η υπορουτίνα perimetroί λειτουργεί με τον τρόπο που αναπτύχθηκε στην παράγραφο 4.3 με μόνη διαφορά τη συνάρτηση check_per. Η συνάρτηση αυτή χρησιμοποιείται για να ελέγξει αν η εξεταζόμενη περίμετρος ικανοποιεί ή όχι τις προϋποθέσεις της τρίτης μεθοδολογίας, συγκρίνοντας το πλήθος των σημείων που έχουν μικρότερο υψόμετρο με αυτό που έχει οριστεί ότι δεχόμαστε (idam), καθώς και την θέση των σημείων αυτών πάνω στην περίμετρο, αφού όπως είναι φυσικό όλα τα σημεία που δεν έχουν το απαιτούμενο υψόμετρο θα πρέπει να είναι συνεχόμενα προκειμένου να είναι δυνατή η δημιουργία φράγματος σε αυτά. Ας σημειωθεί ότι στη δεύτερη μέθοδο η μεταβλητή idam έχει την τιμή μηδέν και αυτός είναι ο λόγος που δεν απαιτείται ο τελευταίος έλεγχος αλλά εξετάζεται μόνο αν υπάρχει κάποιο σημείο με μικρότερο υψόμετρο.

Στην περίπτωση που μια περίμετρος δεν ικανοποιεί τις προϋποθέσεις της μεθόδου, ο έλεγχος δεν τερματίζεται αμέσως αλλά συνεχίζουμε να εξετάζουμε επόμενες περιμέτρους μέχρι το πλήθος των συνεχόμενων περιμέτρων που δεν ικανοποιούν υπερβεί αυτό που δεχόμαστε και ορίζεται με τη μεταβλητή istor.

Αντίθετα αν η περίμετρος ικανοποιεί συνεχίζουμε κατά τα γνωστά τον έλεγχο στην επόμενη περίμετρο και επιπλέον μηδενίζεται το πλήθος των συνεχόμενων περιμέτρων που δεν ικανοποιούν και αρχίζει να αυξάνει ξανά για κάθε επόμενη περίμετρο που δεν ικανοποιεί.

5.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΥ

Για να διερευνήσουμε την τρίτη μέθοδο, όπως στη δεύτερη, έγιναν πολλές εκτελέσεις του προγράμματος limnodeksameni2 δίνοντας διάφορες τιμές στις δυο παραμέτρους που ορίζουν το πλήθος των σημείων μιας περιμέτρου που μπορούν να έχουν μικρότερο υψόμετρο και το μέγιστο πλήθος των διαδοχικών περιμέτρων που δεν ικανοποιούν τις προϋποθέσεις. Αν για παράδειγμα τρέξουμε το πρόγραμμα δίνοντας την τιμή 0 και στις δυο αυτές μεταβλητές θα πάρουμε τα ίδια αποτελέσματα της πρώτης μεθοδολογίας. Η τελευταία παρατήρηση μας βοηθάει να επαληθεύσουμε την σωστή λειτουργία του προγράμματος.

Τα αποτελέσματα που παρουσιάζονται προέκυψαν τρέχοντας το πρόγραμμα με την τιμή 3 για τη μεταβλητή που ορίζει το πλήθος των σημείων μιας περιμέτρου που δεχόμαστε να έχουν μικρότερο υψόμετρο, αφού ένα φράγμα που θα ενώνει τα 3 αυτά συνεχόμενα σημεία θα έχει μήκος ($2 \cdot 90 =$) 180 μέτρα, απόσταση που κρίνεται αρκετά μεγάλη για ένα φράγμα χαμηλού κόστους. Για την μεταβλητή που αναφέρεται στο πλήθος των διαδοχικών περιμέτρων που δεν ικανοποιούν που πρέπει να εντοπιστούν για να σταματήσει ο έλεγχος έγιναν δοκιμές και καταλήξαμε ότι για τιμή μεγαλύτερη από 5 δεν υπάρχουν διαφορές στα αποτελέσματα, ενώ αυξάνει πολύ ο υπολογιστικός χρόνος του προγράμματος.

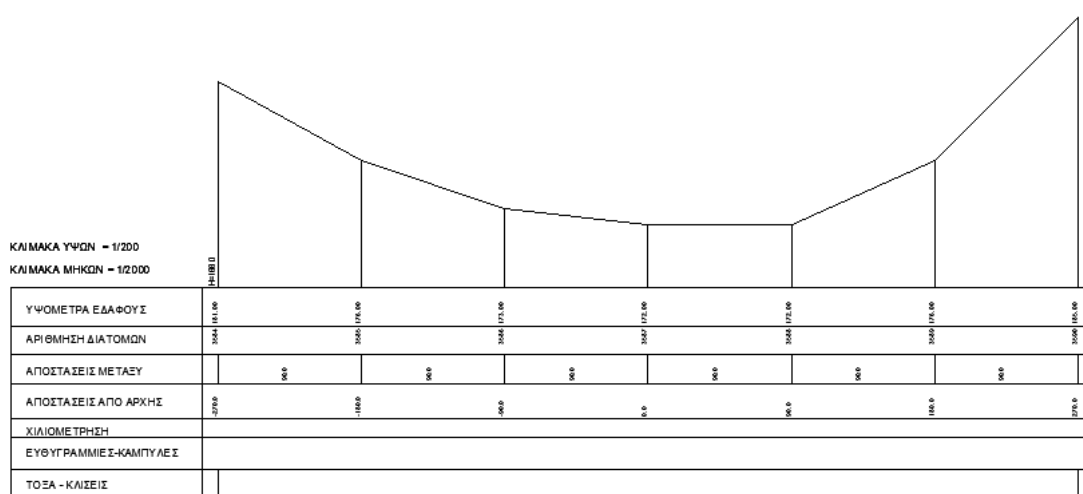
Εντοπίστηκαν 18053 σημεία με δύο περιμέτρους και μόλις 4 σημεία με 3 περιμέτρους, ενώ όπως ήταν αναμενόμενο το ένα από τα 4 αυτά σημεία είναι το ίδιο που είχε προκύψει από τις προηγούμενες δυο μεθόδους. Παρατηρούμε μεγάλη αύξηση στο πλήθος των σημείων με 2 περιμέτρους και αναμένουμε στα περισσότερα από αυτά να υπάρχουν έως 3 συνεχόμενα σημεία της δεύτερης περιμέτρου που να μην έχουν το απαιτούμενο υψόμετρο και αυτός είναι ο λόγος που δεν ικανοποιούσαν την πρώτη μεθοδολογία και εντοπίζονται τώρα. Σχετικά με την τιμή που δόθηκε στη μεταβλητή που ορίζει το πλήθος των διαδοχικών περιμέτρων που δύνανται να έχουν μικρότερο υψόμετρο, αυτή επηρεάζει τα αποτελέσματα μέχρι την τιμή 2, καθώς για μεγαλύτερες τιμές τα αποτελέσματα είναι τα ίδια (τουλάχιστον για την Ελλάδα). Τέλος, παρόλο που τα αποτελέσματα αυξήθηκαν συνεχίζουν να αντιστοιχούν σε μικρές εκτάσεις που απαιτούν μεγάλα βάθη προκειμένου να πετύχουμε τον απαιτούμενο όγκο ύδατος. Σαν αποτέλεσμα μόνο μικρό μέρος των αποτελεσμάτων αντιστοιχεί πραγματικά σε λιμνοδεξαμενή και αυτός είναι ο λόγος που δεν προχωράμε περαιτέρω τη διερεύνηση σχετικά με τον όγκο και τις θέσεις τους.

Τα αποτελέσματα βρίσκονται στο παράρτημα Α.3.1 όπου σε κάθε γραμμή παρουσιάζονται η γραμμή και η στήλη του σημείου στον πίνακα και το πλήθος των περιμέτρων γύρω από το σημείο που ικανοποιούν τη δεύτερη μεθοδολογία. Ωστόσο λόγω του μεγάλου πλήθους των αποτελεσμάτων παρουσιάζονται αρχικά τα τρία σημεία γύρω από τα οποία ικανοποιούν τρεις περίμετροι και στη συνέχεια ορισμένα μόνο από τα σημεία που έχουν δυο περιμέτρους να ικανοποιούν.

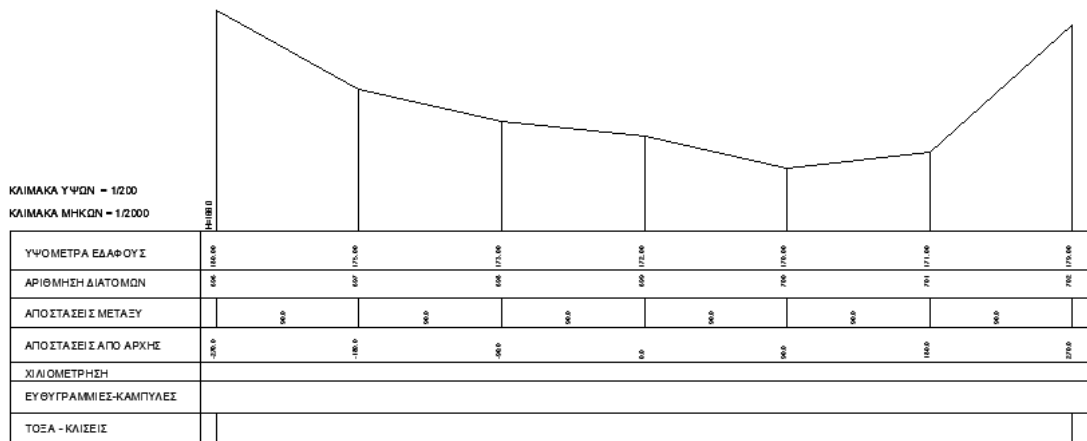
5.5 ΑΠΕΙΚΟΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Όπως στις προηγούμενες μεθοδολογίες έτσι και εδώ απεικονίζουμε τα αποτελέσματά μας με τη δημιουργία μηκοτομών που διέρχονται από αυτά χρησιμοποιώντας την υπορουτίνα mikotomi και το πρόγραμμα c_mhk με την διαδικασία που περιγράφηκε στην παράγραφο 3.6.

Στα σχήματα 5.2 και 5.3 παρουσιάζονται δυο τέτοιες μηκοτομές με κέντρο ένα σημείο που προέκυψε στα αποτελέσματα της μεθόδου να έχει 3 περιμέτρους.



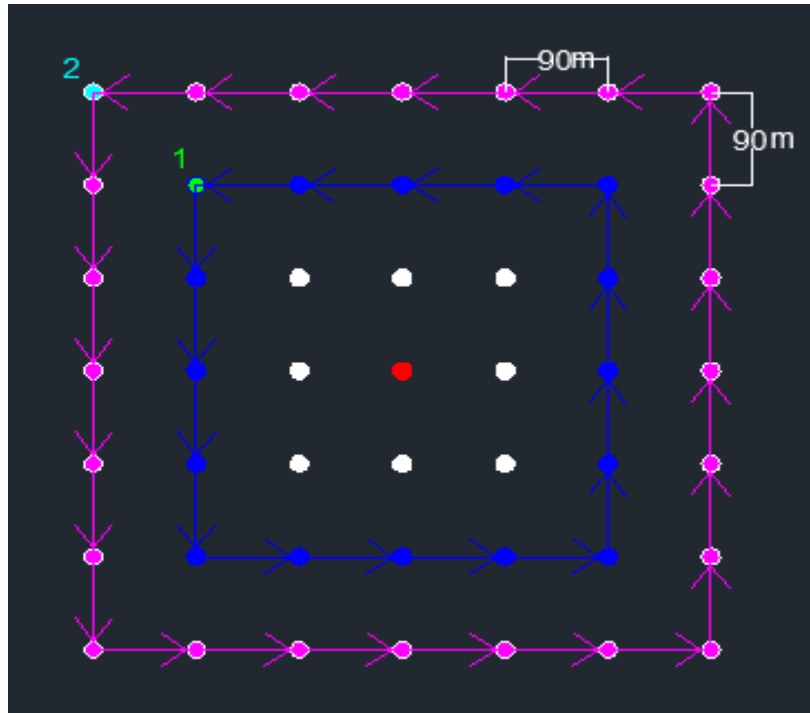
σχήμα 5.2. μηκοτομή αποτελέσματος της μεθοδολογίας 3



σχήμα 5.3. μηκοτομή αποτελέσματος της μεθοδολογίας 3

Βλέπουμε ότι και οι δυο μηκοτομές συμφωνούν με την μορφολογία που περιγράφηκε στη δεύτερη και την τρίτη μέθοδο καθώς τα υψόμετρα μπορεί για ένα διάστημα λίγων περιμέτρων να μειώνονται όμως στη συνέχεια συνεχίζουν να αυξάνουν και όπως περιγράφεται στις δυο αυτές μεθόδους. Περισσότερες μηκοτομές βρίσκονται στο παράρτημα Α.3.2 και σε όλα η μορφή είναι όπως περιμένουμε σύμφωνα με την τρίτη μεθοδολογία.

Εκτός των μηκοτομών στην ίδια γραμμή και στην ίδια στήλη με το κεντρικό σημείο, στην υπορουτίνα *mikotomi* προστίθεται ένα επιπλέον χαρακτηριστικό, αυτό της δημιουργίας μηκοτομής με όλα τα σημεία δυο περιμέτρων που ορίζουμε, γύρω από ένα σημείο. Η υπορουτίνα δέχεται τη θέση του κεντρικού σημείου και τον αριθμό της πρώτης περιμέτρου που θα σχεδιαστεί και δημιουργεί το αρχείο το οποίο χρησιμοποιείται από το πρόγραμμα *c_mhk* για το σχεδιασμό της μηκοτομής. Σε αυτήν το πρώτο σημείο αντιστοιχεί στο πάνω αριστερά σημείο της περιμέτρου (σημείο 1 στο σχήμα 5.4) και ακολουθούν τα επόμενα με ανθρωπολογιακή φορά, όπως φαίνεται στο σχήμα 5.4. Στη συνέχεια στην ίδια μηκοτομή σχεδιάζονται τα σημεία της επόμενης περιμέτρου ξεκινώντας ξανά από το πάνω αριστερά σημείο (σημείο 2 στο σχήμα 5.4) και με την ίδια φορά.



Σχήμα 5.4. Θέσεις και φορά τύπωσης των σημείων στη μηκοτομή περιμέτρου

Η n -οστή περίμετρος έχει $8n$ σημεία και κάθε πλευρά της έχει $2n$ σημεία. Έτσι τα πρώτα $3n$ σημεία ανήκουν στην πρώτη πλευρά. Τα επόμενα $2n$ στη δεύτερη κ.ο.κ.

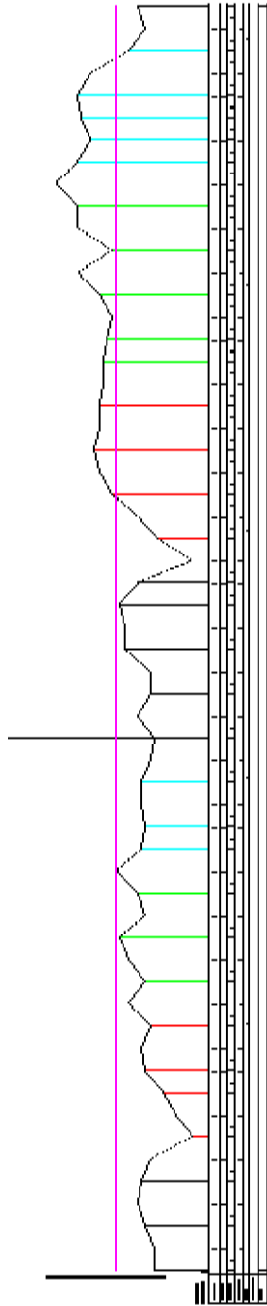
Αυτού του είδους η μηκοτομή χρησιμοποιείται για να προσπαθήσουμε να εξετάσουμε καλύτερα τη μορφολογία του εδάφους και να προσδιορίσουμε τους λόγους για τους οποίους στα αποτελέσματα δεν υπάρχουν σημεία με περισσότερες από τρεις περιμέτρους και να διορθώσουμε τυχόν προβλήματα του προγράμματος αν υπάρχουν.

Στο σημείο γύρω από το οποίο παρουσιάστηκαν προηγουμένως οι μηκοτομές γραμμής και στήλης, παρουσιάζεται τώρα η μηκοτομή της τρίτης και τέταρτης περιμέτρου για να εντοπίσουμε τους λόγους για τους οποίους η τέταρτη περίμετρος δεν ικανοποιεί (σχήμα 5.5).

Αφού πρόκειται για την τρίτη και τέταρτη περίμετρο, τα πρώτα $3 \times 8 = 24$ σημεία ανήκουν στην τρίτη περίμετρο και επειδή σχεδιάζεται δυο φορές το υψόμετρο του πρώτου σημείου τα σημεία είναι 25, και τα επόμενα $4 \times 8 + 1 = 33$ σημεία στην τέταρτη, ενώ κάθε πλευρά της τρίτης περιμέτρου έχει $3 \times 2 = 6$ σημεία. Δίνουμε ίδια χρώματα στα σημεία που ανήκουν στην ίδια πλευρά για να γίνει πιο κατανοητή η μηκοτομή.

Παρατηρούμε ότι το μέγιστο υψόμετρο της τρίτης περιμέτρου είναι 186 ενώ τα σημεία της τέταρτης περιμέτρου με υψόμετρο μικρότερο από 186 είναι πολύ περισσότερα από τρία. Η

τέταρτη περίμετρος δεν ικανοποιεί τις προϋποθέσεις της τρίτη μεθόδου όπως βρέθηκε και στα αποτελέσματά της. Καταλήγουμε στο συμπέρασμα ότι η τρίτη μεθοδολογία όπως και η δεύτερη θέτουν πολύ απαιτητικές προϋποθέσεις για μια περίμετρο. Έτσι παρόλο που τα αποτελέσματά της είναι σωστά, είναι περιορισμένα σε πλήθος και απαιτείται μια νέα προσέγγιση.



Σχήμα 5.5. Μηκοτομή 3ης και 4ης περιμέτρου σημείου αποτελεσμάτων μεθοδολογίας2

6. ΜΕΘΟΔΟΛΟΓΙΑ 4

6.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό αναπτύσσεται η τέταρτη μεθοδολογία για την εύρεση θέσεων λιμνοδεξαμενών στην περιοχή της Ελλάδας με χρήση των υψομετρικών δεδομένων που έχουμε στη διάθεσή μας από το παγκόσμιο ψηφιακό μοντέλο SRTM. Στην τέταρτη μέθοδο όπως και στις δυο προηγούμενες δεχόμαστε ορισμένα σημεία μακριά από το κέντρο της (πιθανής) λιμνοδεξαμενής να έχουν μικρότερο υψόμετρο από σημεία που είναι πιο κοντά στο κέντρο. Αλλάζει όμως το υψόμετρο με το οποίο συγκρίνουμε τα υψόμετρα μιας περιμέτρου.

6.2 ΜΕΘΟΔΟΣ

Μετά από παρατήρηση των μηκοτομών των περιμέτρων για τα αποτελέσματα της τρίτης μεθοδολογίας καταλήξαμε στο συμπέρασμα ότι ο λόγος για τον οποίο υπολογίζεται περιορισμένος αριθμός θέσεων λιμνοδεξαμενών είναι οι αυστηρές προϋποθέσεις που έχουμε θέσει σχετικά με την αύξηση των υψομέτρων από μια περίμετρο στην επόμενη. Με την τέταρτη μεθοδολογία προσπαθούμε να χαλαρώσουμε τις προϋποθέσεις αυτές ώστε να υπολογιστούν περισσότερες θέσεις λιμνοδεξαμενών.

Στην τέταρτη μεθοδολογία, όπως και στις δυο προηγούμενες αναζητούμε φυσικές κοιλότητες στις οποίες τα υψόμετρα δεν είναι απαραίτητα μονότονα αυξανόμενα, αντίθετα μπορούν σε μια περιοχή να παραμένουν σταθερά ή ακόμα και να μειώνονται, αρκεί στη συνέχεια να συνεχίσουν να αυξάνουν. Ένα πλήθος περιμέτρων δύνανται να έχουν σημεία με μικρότερο υψόμετρο, αρκεί να εντοπιστεί επόμενη περίμετρος της οποίας τα σημεία έχουν αρκετά μεγάλα υψόμετρα.

Στην τέταρτη μεθοδολογία αναζητούμε ακόμα κοιλότητες, οι οποίες μπορούν να γίνουν λιμνοδεξαμενές με την κατασκευή φράγματος. Όπως στην τρίτη μέθοδο έτσι και εδώ δεχόμαστε ένας αριθμός σημείων μιας περιμέτρου να έχουν μικρότερο υψόμετρο από τα εσωτερικά της σημεία, αρκεί αυτά να είναι σε γειτονικές θέσεις.

Προκειμένου να βρούμε τέτοιες περιοχές δουλεύουμε όπως στην τρίτη μέθοδο με μόνη διαφορά ότι για να βρούμε το πλήθος των σημείων μιας περιμέτρου που δεν έχουν το απαιτούμενο υψόμετρο συγκρίνουμε το υψόμετρο κάθε σημείου της περιμέτρου με το ελάχιστο υψόμετρο της προηγούμενης της αντί του μέγιστου. Αν το ελάχιστο υψόμετρο μιας

περιμέτρου είναι μεγαλύτερο από το ελάχιστο υψόμετρο της προηγούμενης της αυτό σημαίνει ότι το νερό μπορεί να φτάσει μέχρι το ελάχιστο υψόμετρο της τελευταίας περιμέτρου χωρίς να υπάρχουν διαφυγές. Το μέγιστο βάθος της λιμοδεξαμενής που μπορεί θεωρητικά να προκύψει από την κοιλότητα, ισούται επομένως με τη διαφορά Δh μεταξύ του υψομέτρου του κεντρικού σημείου και του ελάχιστου υψομέτρου της n-οστής περιμέτρου.

Αρχικά ελέγχουμε το υψόμετρο των σημείων της πρώτης περιμέτρου και βρίσκουμε το πλήθος των σημείων που έχουν μικρότερο υψόμετρο από το αρχικό. Αν το πλήθος των σημείων είναι μικρότερο από το μέγιστο που δεχόμαστε (idam) και οι θέσεις τους είναι συνεχόμενες το αρχικό σημείο είναι το κέντρο τετραγωνικής κοιλότητας 1ης περιμέτρου.

Για να βρούμε το πλήθος των σημείων μιας επόμενης περιμέτρου που δεν έχουν τα απαιτούμενα υψόμετρα συγκρίνουμε το υψόμετρο κάθε σημείου της περιμέτρου με το ελάχιστο υψόμετρο της προηγούμενης της. Με τον τρόπο αυτόν εξασφαλίζεται ότι από κάποια περίμετρο στην επόμενη τα υψόμετρα συνεχώς αυξάνουν, τουλάχιστον για Δh .

Η ίδια διαδικασία επαναλαμβάνεται για τις επόμενες περιμέτρους. Αν το πλήθος των σημείων μιας περιμέτρου που δεν έχουν το απαιτούμενο υψόμετρο είναι μεγαλύτερο από το μέγιστο για τη δημιουργία φράγματος, ή τα σημεία αυτά δεν είναι σε συνεχόμενες θέσεις, ή τέλος βρεθεί ένα σημείο με υψόμετρο μικρότερο του μηδενός η περίμετρος δεν ικανοποιεί τις προϋποθέσεις της μεθοδολογίας. Η διαδικασία θα τερματιστεί όταν βρεθούν περισσότερες συνεχόμενες μη συμμορφούμενες περιμέτροι από το μέγιστο (istor). Κάθε φορά που βρίσκουμε μια περίμετρο που ικανοποιεί τη μεθοδολογία το πλήθος των συνεχόμενων μη συμμορφούμενων περιμέτρων που έχει εντοπιστεί μέχρι αυτήν μηδενίζεται.

Ο αλγόριθμος είναι παρόμοιος με αυτόν της τρίτης μεθοδολογίας και συνοψίζεται στο σχήμα 5.1 της παραγράφου 5.2 με τη διαφορά ότι αντί για τον υπολογισμό του hmax υπολογίζουμε το hmin για κάθε περίμετρο, ενώ και η σύγκριση για να βρεθεί το πλήθος των σημείων μιας περιμέτρου που δεν έχουν το απαιτούμενο υψόμετρο γίνεται με το ελάχιστο αντί για το μέγιστο της προηγούμενης περιμέτρου όπως περιγράφηκε παραπάνω.

Για την τέταρτη μέθοδο κατασκευάστηκε πρόγραμμα H/Y (limnodeksameni3) σε γλώσσα FORTRAN που εφαρμόζει τον παραπάνω αλγόριθμο και υπολογίζει πιθανές θέσεις λιμοδεξαμενών. Ο κώδικας του προγράμματος δίνεται στο παράρτημα Β.3.

6.3 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ – ΕΥΡΕΣΗ ΛΙΜΝΟΔΕΞΑΜΕΝΩΝ

Η εισαγωγή των υψομετρικών δεδομένων των σημείων γίνεται κατά τα γνωστά με τις υπορουτίνες `openBmp`, `closeBmp` και `thanSet`, ενώ χρησιμοποιείται η συνάρτηση `him` για να πάρουμε το υψόμετρο ενός σημείου, δίνοντας τη θέση του στο μεγάλο πίνακα, όπως περιγράφηκε στην παράγραφο 3.3.

Για τον έλεγχο των σημείων χρησιμοποιείται ξανά η υπορουτίνα `perimetroi`. Ψάχνουμε ίδιας μορφής περιοχές, όπως στην τρίτη μέθοδο. Έτσι και εδώ εισάγονται δυο μεταβλητές, μια που ορίζει το μέγιστο πλήθος των συνεχόμενων σημείων μιας περιμέτρου που δύνανται να μην έχουν το απαιτούμενο υψόμετρο και έχει το όνομα `idam` και μια που αναφέρεται στο πλήθος των διαδοχικών περιμέτρων που δύνανται να μην ικανοποιούν και έχει το όνομα `istor`.

Η υπορουτίνα `perimetroi` ελέγχει κάθε σημείο με υψόμετρο μεγαλύτερο των 10 μέτρων ως κέντρο πιθανής λιμνοδεξαμενής, εξετάζοντας διαδοχικές περιμέτρους γύρω από αυτό. Η υπορουτίνα έχει την ίδια δομή όπως στο πρόγραμμα `limnodeksameni2`, αυτό που αλλάζει είναι ότι πλέον ο έλεγχος των υψομέτρων των σημείων μιας περιμέτρου δεν γίνεται με το μέγιστο υψόμετρο της προηγούμενης περιμέτρου αλλά με το ελάχιστο. Έτσι η υπορουτίνα `find_pl` δέχεται τώρα το ελάχιστο υψόμετρο της προηγούμενης περιμέτρου. Στη συνέχεια η υπορουτίνα `find_min`, η οποία δίνει το ελάχιστο υψόμετρο της περιμέτρου που είναι μεγαλύτερο από ένα υψόμετρο που εμείς ορίζουμε, καλείται διαδοχικές φορές έως ότου το υψόμετρο που θα πάρουμε είναι μεγαλύτερο από το ελάχιστο της προηγούμενης περιμέτρου, οπότε η περίμετρος θα ικανοποιεί. Αντίθετα αν το πλήθος των σημείων με μικρότερο υψόμετρο υπερβεί το μέγιστο που ορίζεται από τη μεταβλητή `idam` η περίμετρος δεν ικανοποιεί. Τέλος σταματάμε να καλούμε την υπορουτίνα `find_min` αν βρεθεί ένα σημείο με υψόμετρο μικρότερο του μηδενός, καθώς στην περίπτωση αυτή η περίμετρος δε μπορεί να ανήκει σε λιμνοδεξαμενή αφού ανήκει στην θάλασσα ή είμαστε σε περιοχή εκτός του πίνακα δηλαδή εκτός Ελλάδας.

6.4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΘΟΔΟΛΟΓΙΑΣ

Όπως στις προηγούμενες δυο μεθοδολογίες έτσι και τώρα το πρόγραμμα `limnodeksameni3` εκτελέστηκε για διάφορες τιμές στις μεταβλητές που ορίζουν το πλήθος των συνεχόμενων σημείων μιας περιμέτρου που δύνανται να μην έχουν το απαιτούμενο υψόμετρο και τον μέγιστο αριθμό των διαδοχικών μη συμμορφούμενων περιμέτρων που πρέπει να βρεθούν για

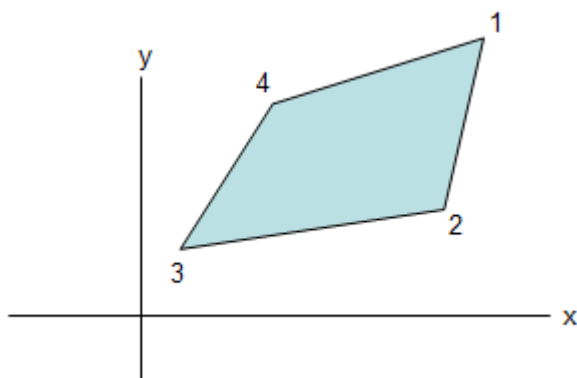
να σταματήσουμε τον έλεγχο και για κάθε συνδυασμό προέκυψαν διαφορετικά αποτελέσματα.

Στα αποτελέσματα που κρίνονται ως πιο ορθά, δεχόμαστε μέχρι τρία συνεχόμενα σημεία μιας περιμέτρου να έχουν μικρότερο υψόμετρο, καθώς ένα φράγμα που θα ένωνε τα σημεία αυτά για να τους δώσει το απαιτούμενο υψόμετρο θα είχε μήκος περίπου 180 μέτρα, το οποίο κρίνεται αρκετά μεγάλο για φράγμα μικρού κόστους. Όσον αφορά το πλήθος των συνεχόμενων περιμέτρων που δεν ικανοποιούν η τιμή 5 που δόθηκε για τα αποτελέσματα των δυο προηγούμενων μεθόδων κρίνεται επαρκής, καθώς μεγαλύτερες τιμές αυξάνουν κατά πολύ τον χρόνο που τρέχει το πρόγραμμα ενώ και η αύξηση στα αποτελέσματα αφορά σε σημεία γειτονικά αυτών που έχουν ήδη εντοπιστεί και τα οποία ουσιαστικά ανήκουν στην ίδια λιμνοδεξαμενή. Άλλωστε ήδη με την τιμή 5 παρατηρούμε στα αποτελέσματα που παρουσιάζονται αρκετά τέτοια γειτονικά σημεία που αναφέρονται στην ίδια λιμνοδεξαμενή, αν όντως αυτή υπάρχει στο συγκεκριμένο σημείο. Τέλος, το πρόγραμμα εξετάζει ως κέντρα λιμνοδεξαμενών μόνο τα σημεία μεταξύ της 4200ης και 12500ης γραμμής και της πρώτης έως την 7700η στήλη, αφού αυτές προέκυψαν από το πρόγραμμα google earth (με τον τρόπο που θα περιγραφεί παρακάτω) να ανήκουν στην Ελλάδα.

Η μεθοδολογία αυτή, σε αντίθεση με τις προηγούμενες, έδωσε πολύ μεγάλο αριθμό αποτελεσμάτων, με σημεία να έχουν ακόμα και 20 περιμέτρους γύρω τους να ικανοποιούν. Μάλιστα το πρόγραμμα σταματάει τον έλεγχο μετά την 20η περίμετρο, αφού ένα σημείο που έχει 20 περιμέτρους να ικανοποιούν προφανώς αποτελεί πιθανή θέση λιμνοδεξαμενής και δεν απαιτείται περαιτέρω διερεύνηση. Λόγω του μεγάλου πλήθους τους, προσπαθούμε να φιλτράρουμε τα αποτελέσματα με βάση τον όγκο της κάθε λιμνοδεξαμενής. Για το σκοπό αυτό χρησιμοποιούμε τη λειτουργία volume, με την οποία παίρνουμε για κάθε σημείο τον όγκο της λιμνοδεξαμενής που σχηματίζεται γύρω του. Στη λειτουργία εισάγεται η θέση του κεντρικού σημείου και το ελάχιστο υψόμετρο της τελευταίας περιμέτρου που είναι ουσιαστικά το μέγιστο υψόμετρο μέχρι το οποίο μπορεί να φτάσει το νερό χωρίς να έχουμε διαφυγές. Το μέγιστο βάθος της λιμνοδεξαμενής που μπορεί θεωρητικά να προκύψει από την κοιλότητα, είναι η διαφορά ΔH μεταξύ του υψομέτρου του κεντρικού σημείου και του ελάχιστου υψομέτρου της n-οστής περιμέτρου. Ο όγκος της λιμνοδεξαμενής δίνεται προσεγγιστικά από τον όγκο κόλουρης τετραγωνικής πυραμίδας: $E \frac{\Delta H}{3}$.

Για την εύρεση του εμβαδού της επιφάνειας του νερού χρησιμοποιείται η συνάρτηση $emvadon$, στην οποία εφαρμόζεται ο παρακάτω τύπος που αφορά εύρεση εμβαδού

χρησιμοποιώντας τις συντεταγμένες σημείων που δίνονται με ωρολογιακή ή ανθωρολογιακή φορά (σχήμα 6.1).



$$\left| \sum_{i=1}^n (x_{i+1} - x_i) \left(\frac{y_{i+1} + y_i}{2} \right) \right|$$

Σχήμα 6.1. εύρεση εμβαδού διαδοχικών σημείων χρησιμοποιώντας τις συντεταγμένες τους

Τα αποτελέσματα βρίσκονται στο παράρτημα Α.4.1 και έχουν προκύψει με την τιμή 3 ως μέγιστο πλήθος σημείων μιας περιμέτρου που δύνανται να μην έχουν το απαιτούμενο υψόμετρο και την τιμή 5 στην μεταβλητή που ορίζει το πλήθος των συνεχόμενων μη συμμορφούμενων περιμέτρων. Σε κάθε γραμμή τυπώνεται η θέση του κεντρικού σημείου (στήλη και γραμμή στον πίνακα), το πλήθος των περιμέτρων που ικανοποιούν την τέταρτη μεθοδολογία γύρω από το σημείο, το ελάχιστο και μέγιστο υψόμετρο και τέλος ο όγκος της λιμνοδεξαμενής που υπολογίζεται γύρω από το σημείο. Σε αυτά εμφανίζονται μόνο τα σημεία που αντιστοιχούν σε κέντρα λιμνοδεξαμενής με όγκο μεγαλύτερο των 4.000.000 κυβικών μέτρων, ο οποίος θεωρείται αρκετά μεγάλος. Βρέθηκαν 521 τέτοια σημεία.

Από τα σημεία αυτά πολλά έχουν κοντινές θέσεις στον πίνακα, απέχουν δηλαδή μικρή απόσταση και πρόκειται ουσιαστικά για γειτονικά σημεία, τα οποία αναφέρονται στην ίδια λιμνοδεξαμενή. Το στοιχείο αυτό μπορεί ωστόσο να χρησιμοποιηθεί αν θεωρήσουμε ότι ο εντοπισμός πολλών γειτονικών σημείων κάνει πιο πιθανή την ύπαρξη λιμνοδεξαμενής σε μια περιοχή. Είναι λογικό άλλωστε να βρεθούν πολλά σημεία μέσα σε μια λιμνοδεξαμενή που ικανοποιούν τις προϋποθέσεις που έχουμε θέσει με την τέταρτη μεθοδολογία και όχι μόνο το κεντρικό σημείο. Δεν αποτελούν επομένως και τα 521 σημεία πιθανές θέσεις λιμνοδεξαμενών, αλλά ένας μικρότερος αριθμός αυτών και απαιτείται περαιτέρω διερεύνηση των θέσεων του καθενός για να βρούμε το πραγματικό τους πλήθος. Ακόμα ορισμένες από τις

περιοχές που υπολογίστηκαν από το πρόγραμμα limnodeksameni3 να έχουν μικρότερο όγκο από 4.000.000 κυβικά μέτρα θα μπορούσαν να αποτελούν θέσεις λιμνοδεξαμενής, δεν παρουσιάζονται ωστόσο για τους λόγους που αναφέρθηκαν παραπάνω.

Ένα άλλο χαρακτηριστικό που παρατηρούμε στα αποτελέσματα είναι οι πολύ μεγάλοι όγκοι των λιμνοδεξαμενών που εντοπίζονται. Υπολογίζονται όγκοι που φτάνουν ακόμα και τα 80.000.000 κυβικά μέτρα νερού που πιθανώς να μη γεμίσουν ποτέ. Ένας λόγος που μας οδηγεί σε τόσο μεγάλους όγκους είναι ο προσεγγιστικός τύπος που χρησιμοποιείται για τον υπολογισμό τους. Ένας ακόμα λόγος είναι ο πολύ μεγάλος αριθμός συμμορφούμενων περιμέτρων που αντιστοιχούν σε έκταση μέχρι και 12.000 στρεμμάτων. Πρόκειται για πολύ μεγάλες περιοχές που έχουν μεν τα χαρακτηριστικά που αναζητάμε, σε τόσο μεγάλο μέγεθος όμως ίσως δεν μπορούν να αποτελέσουν θέση λιμνοδεξαμενής, αφού πιθανότατα είναι καλλιεργήσιμες περιοχές και κάμποι. Τα μεγέθη αυτά οδηγούν σε πολύ μεγάλους όγκους, ειδικά αν συνδυαστούν με μεγάλα υψόμετρα. Τέλος, οι μεγάλοι όγκοι εντοπίζονται συνήθως σε ορεινές θέσεις, όπου τα υψόμετρα αυξάνουν σημαντικά και οδηγούν σε μεγάλο βάθος λιμνοδεξαμενής, όπως αυτό υπολογίζεται από το πρόγραμμα limnodeksameni3 και άρα μεγάλο όγκο.

Όπως φαίνεται, λοιπόν, πολλά από τα αποτελέσματα που παίρνουμε από το πρόγραμμα limnodeksameni3 ίσως να μην είναι ρεαλιστικά όσον αφορά την έκταση και τον όγκο που υπολογίζεται, αποτελούν ωστόσο πιθανές θέσεις λιμνοδεξαμενών μικρότερων διαστάσεων, ενώ μας βοηθούν και στο να κάνουμε συγκρίσεις μεταξύ των σημείων που έχουν εντοπιστεί στα αποτελέσματα.

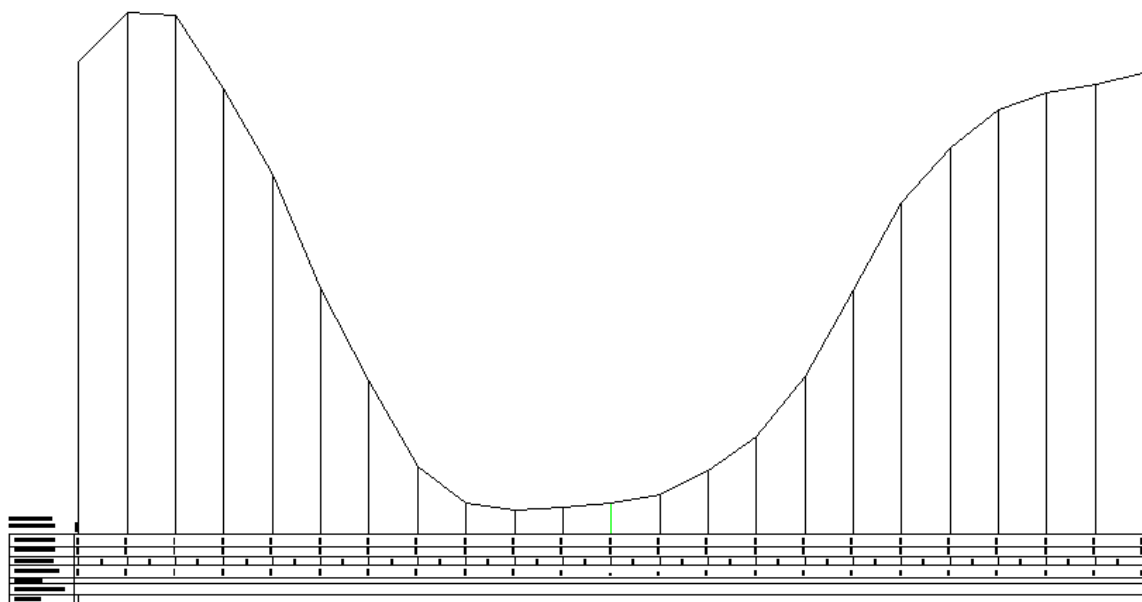
6.5 ΑΠΕΙΚΟΝΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Η παράγραφος αυτή αναφέρεται στις μεθόδους που χρησιμοποιούμε με σκοπό να απεικονίσουμε τα αποτελέσματα. Σκοπός μας είναι να εξακριβώσουμε αν σε αυτά υπάρχουν περιοχές όπως αυτές που περιγράφονται στην τέταρτη μεθοδολογία καθώς και ο εντοπισμός τους στο χάρτη, ώστε να επαληθευτούν οι θέσεις που προέκυψαν από τα δεδομένα του SRTM με τις ίδιες θέσεις στην πραγματικότητα.

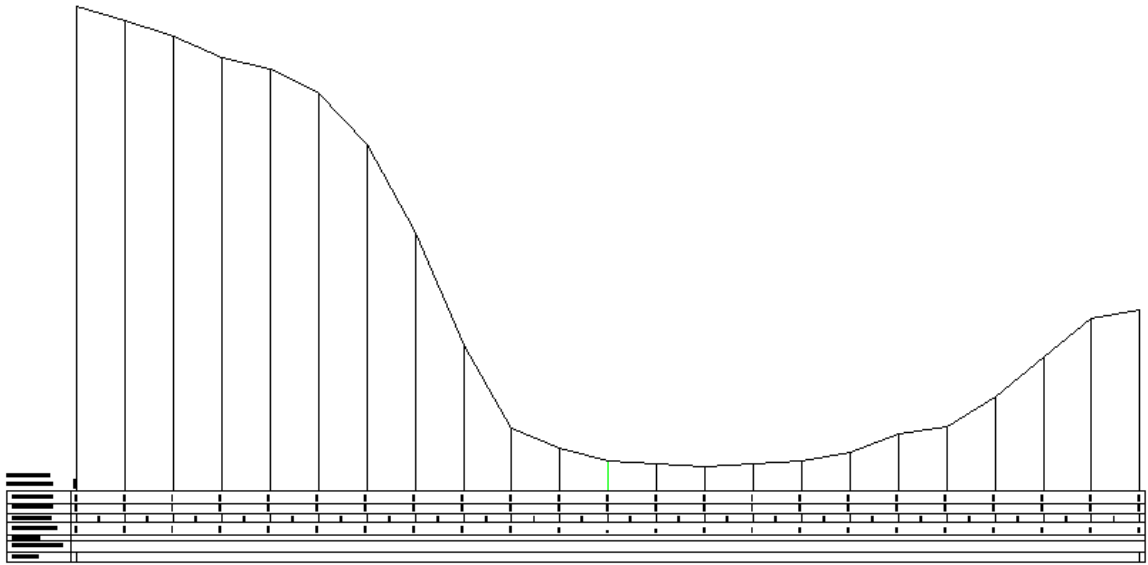
Όπως στις δυο προηγούμενες μεθόδους έτσι και σε αυτήν κάνουμε δυο μηκοτομές που περνάνε από το κεντρικό σημείο της λιμνοδεξαμενής που έχει προκύψει στα αποτελέσματα, μια με σημεία της ίδιας γραμμής και μια με σημεία της ίδιας στήλης με αυτό, προκειμένου να ελέγξουμε αν η μορφολογία του εδάφους είναι αυτή που περιμένουμε σύμφωνα με την

τέταρτη μεθοδολογία, καθώς και να εντοπίσουμε τυχόν σφάλματα του προγράμματος με το οποίο προέκυψαν αυτά. Χρησιμοποιούμε, λοιπόν, την υπορουτίνα *mikotomi*, η οποία δουλεύει με τον ίδιο τρόπο όπως στα προηγούμενα προγράμματα, και το πρόγραμμα *c_mhk* και παίρνουμε τις μηκοτομές για το σημείο που θέλουμε και για το πλήθος των περιμέτρων που ικανοποιούν γύρω από αυτό.

Παρουσιάζονται δυο τέτοιες μηκοτομές με κέντρο το σημείο που βρίσκεται στην 6777η γραμμή και 550η στήλη του πίνακα και βρέθηκε να έχει 11 περιμέτρους που ικανοποιούν τις προϋποθέσεις της τέταρτης μεθοδολογίας (σχήματα 6.2, σχήμα 6.3)



Σχήμα 6.2. μηκοτομή στην ίδια γραμμή μεαποτελεσμα της τέταρτης μεθοδολογίας



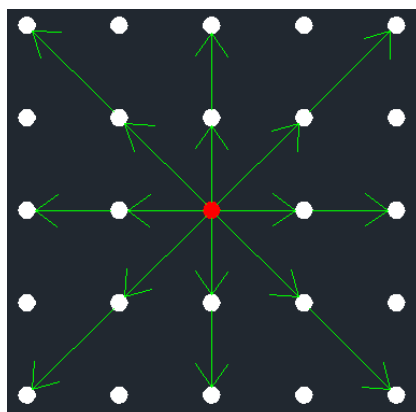
Σχήμα 6.3. μηκοτομή στην ίδια στήλη με αποτέλεσμα της τέταρτης μεθόδου

Και στις δυο μηκοτομές το κεντρικό σημείο είναι το 12ο, αυτό με χιλιομετρική θέση μηδέν και του έχουμε δώσει πράσινο χρώμα ώστε να ξεχωρίζει. Στην πρώτη μηκοτομή, που περιλαμβάνει τα σημεία της ίδιας γραμμής με το κεντρικό, βλέπουμε ότι τα υψόμετρα αυξάνουν εκατέρωθεν του κεντρικού σημείου και διαμορφώνεται έτσι τυπική μορφή εδάφους που αναζητάμε. Στη δεύτερη μηκοτομή, που περιλαμβάνει τα σημεία της ίδιας στήλης με το κεντρικό, παρατηρούμε ότι τα υψόμετρα δεξιά του κεντρικού σημείου αρχικά μειώνονται ή παραμένουν σταθερά στη συνέχεια όμως αυξάνουν. Πρόκειται για μορφή εδάφους που περιγράφηκε στις μεθοδολογίες που αναπτύχθηκαν. Παρατηρούμε ακόμα ότι τα υψόμετρα στα δεξιά του κεντρικού σημείου είναι μικρότερα από αυτά στα αριστερά, έτσι το κεντρικό σημείο δεν είχε τόσες συμμορφούμενες περιμέτρους σύμφωνα με τη δεύτερη και την τρίτη μεθοδολογία καθώς τα υψόμετρα από περίμετρο σε περίμετρο δεν αυξάνονται όσο αυτές απαιτούν, αυτός είναι και ο λόγος που το σημείο εντοπίζεται μόνο στα αποτελέσματα της τέταρτης μεθοδολογίας.

Περισσότερες μηκοτομές των αποτελεσμάτων που προέκυψαν με την τέταρτη μέθοδο βρίσκονται στο παράρτημα A.4.2 και σε όλα η μορφή είναι σύμφωνη με αυτή που περιγράφηκε στις προηγούμενες παραγράφους.

Ένας ακόμα τρόπος για να εξακριβώσουμε ότι η μορφολογία του εδάφους σε μια περιοχή κοντά στο σημείο που έχουμε εντοπίσει αντιστοιχεί σε λιμνοδεξαμενή είναι η απεικόνισή της με τη δημιουργία των ισουψών καμπυλών για διάφορα υψόμετρα γύρω από το σημείο αυτό. Χρησιμοποιούμε, λοιπόν, το πρόγραμμα akam, το οποίο δέχεται ένα αρχείο που περιέχει τις συντεταγμένες των σημείων που έχουν το ζητούμενο υψόμετρο και δίνει την ισουψή καμπύλη

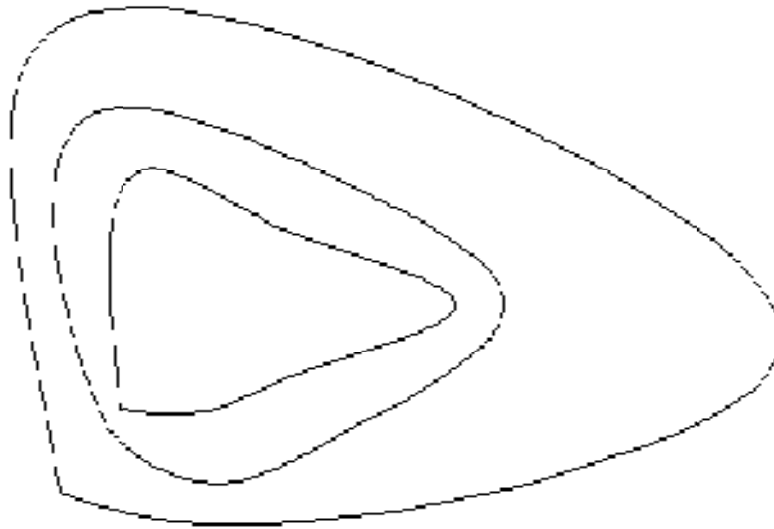
για το υψόμετρο αυτό. Η δημιουργία του αρχείου αυτού γίνεται από την υπορουτίνα `isoupsis`. Σε αυτήν εισάγεται η θέση του κεντρικού σημείου γύρω από το οποίο θέλουμε να γίνει η ισουψής και το υψόμετρό της και τυπώνει στο αρχείο τις συντεταγμένες οκτώ σημείων γύρω από το κεντρικό που έχουν το ζητούμενο υψόμετρο. Κάθε ένα σημείο είναι σε μια από τις διευθύνσεις που φαίνεται στο σχήμα 6.4.



Σχήμα 6.4. διευθύνσεις σημείων ισουψούς γύρω από το κεντρικό

Για κάθε διεύθυνση καλείται η υπορουτίνα `get_next_point` και παίρνουμε τη θέση του επόμενου σημείου σε αυτή έως ότου βρεθεί ένα σημείο με υψόμετρο μεγαλύτερο αυτού για το οποίο κάνουμε την ισουψή. Στη συνέχεια εφαρμόζεται γραμμική παρεμβολή και βρίσκουμε τη θέση του σημείου που έχει το ζητούμενο υψόμετρο. Η θέση αυτή ωστόσο είναι εικονική, αφού είναι πραγματικός αριθμός πχ 2,5 γραμμές 5,2 στήλες και χρησιμοποιείται μόνο για να βρούμε τις συντεταγμένες του σημείου σε μέτρα ως προς το κεντρικό. Για να το κάνουμε αυτό βρίσκουμε πόσες γραμμές και πόσες στήλες απέχει το σημείο αυτό από το κεντρικό μας και πολλαπλασιάζουμε τον αριθμό αυτόν με 90 και 70, όση είναι και η απόσταση μεταξύ διαδοχικών γραμμών και στηλών σε μέτρα. Οι συντεταγμένες αυτές τυπώνονται στο αρχείο για τη δημιουργία της ισουψούς από το πρόγραμμα `akam`, ενώ αποθηκεύονται και σε έναν πίνακα, ο οποίος επιστρέφεται από την υπορουτίνα και χρησιμοποιείται από τη συνάρτηση `emvadon` για τον υπολογισμό του εμβαδού που περικλείεται σε αυτά τα σημεία.

Καλώντας την υπορουτίνα `isoupsis` για υψόμετρα 20, 40 και 60 μέτρα μεγαλύτερα από αυτό του κεντρικού σημείου που ανήκει στη 6777η γραμμή και 550η στήλη παίρνουμε τις ισουψείς που φαίνονται στο σχήμα 6.5



Σχήμα 6.5. Ισοψηφείς καμπύλες γύρω από σημείο των αποτελεσμάτων της τέταρτης μεθοδολογίας

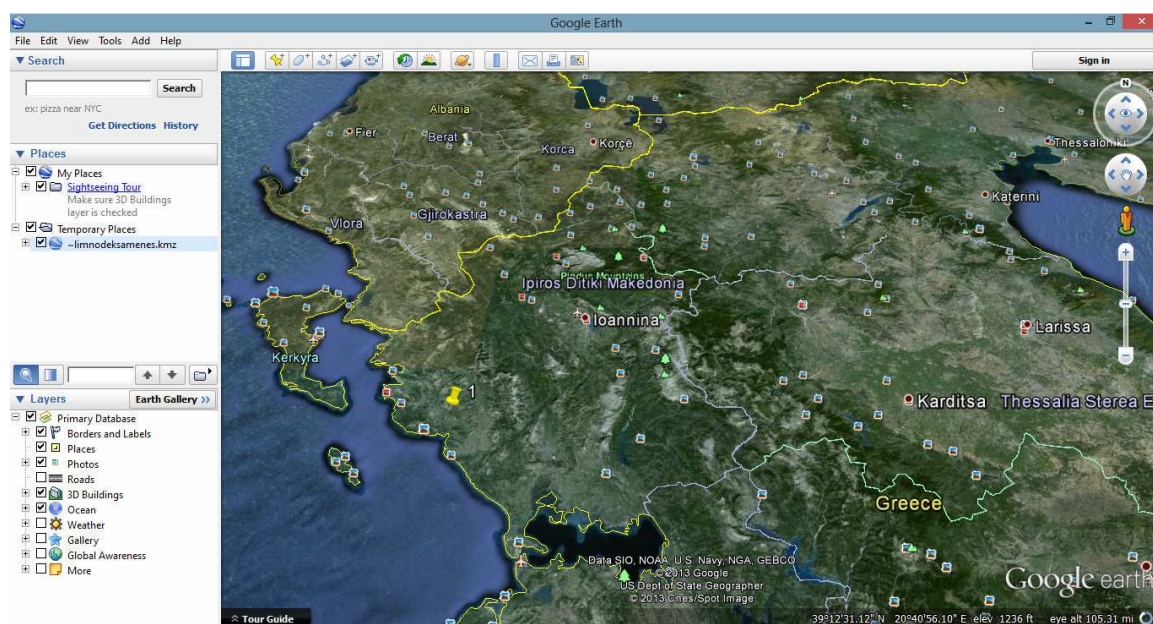
Όπως παρατηρούμε οι ισοψηφείς μικρότερων υψομέτρων περιέχονται μέσα σε αυτές μεγαλύτερων. Φαίνεται δηλαδή ότι όσο απομακρυνόμαστε από το κεντρικό σημείο τα υψόμετρα συνεχώς αυξάνουν. Έτσι το σημείο αυτό αποτελεί θέση λιμνοδεξαμενής σύμφωνα με τη μορφή του εδάφους που φαίνεται να έχει και ορθά έχει εντοπιστεί από το πρόγραμμα.

Δημιουργώντας μηκοτομές και ισοψηφείς σε περιοχές που έχουν υπολογιστεί από το πρόγραμμα επαληθεύσαμε ότι αυτές έχουν μορφή που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή, όμως όλα τα παραπάνω έχουν προκύψει από επεξεργασία των δεδομένων του SRTM. Υπάρχει ωστόσο η πιθανότητα τα δεδομένα αυτά να είναι λανθασμένα ή ακόμα θα μπορούσε να υπάρχει κάποιο συστηματικό λάθος στο πρόγραμμα, πχ τοποθέτηση των πινάκων σε λάθος θέσεις. Για το λόγο αυτό προχωράμε στον εντοπισμό των αποτελεσμάτων στο χάρτη, για να επαληθεύσουμε ότι αυτά συμφωνούν με τα συμπεράσματα από την επεξεργασία των υψομετρικών δεδομένων του SRTM. Ο εντοπισμός στο χάρτη είναι άλλωστε απαραίτητος για να δούμε σε ποια περιοχή βρίσκονται οι θέσεις και αν είναι όντως αξιοποιήσιμες, αφού μια περιοχή θα μπορούσε να έχει για παράδειγμα μορφή που αντιστοιχεί σε λιμνοδεξαμενή, να βρίσκεται όμως σε κατοικημένη περιοχή.

Για τον εντοπισμό ενός σημείου στο χάρτη αρκεί η εύρεση των γεωγραφικών του συντεταγμένων. Αυτό γίνεται με την υπορουτίνα I_f , στην οποία εισάγεται η θέση ενός σημείου στον πίνακα και επιστρέφει το γεωγραφικό μήκος (λ) και το γεωγραφικό πλάτος (φ) στα οποία αυτό αντιστοιχεί. Για τον υπολογισμό τους η υπορουτίνα χρησιμοποιεί τις γεωγραφικές συντεταγμένες του πρώτου σημείου του πίνακα και την πληροφορία για την διαφορά στο γεωγραφικό μήκος μεταξύ διαδοχικών στηλών και τη διαφορά στο γεωγραφικό πλάτος μεταξύ διαδοχικών γραμμών του πίνακα που διαθέτει από τα δεδομένα του SRTM.

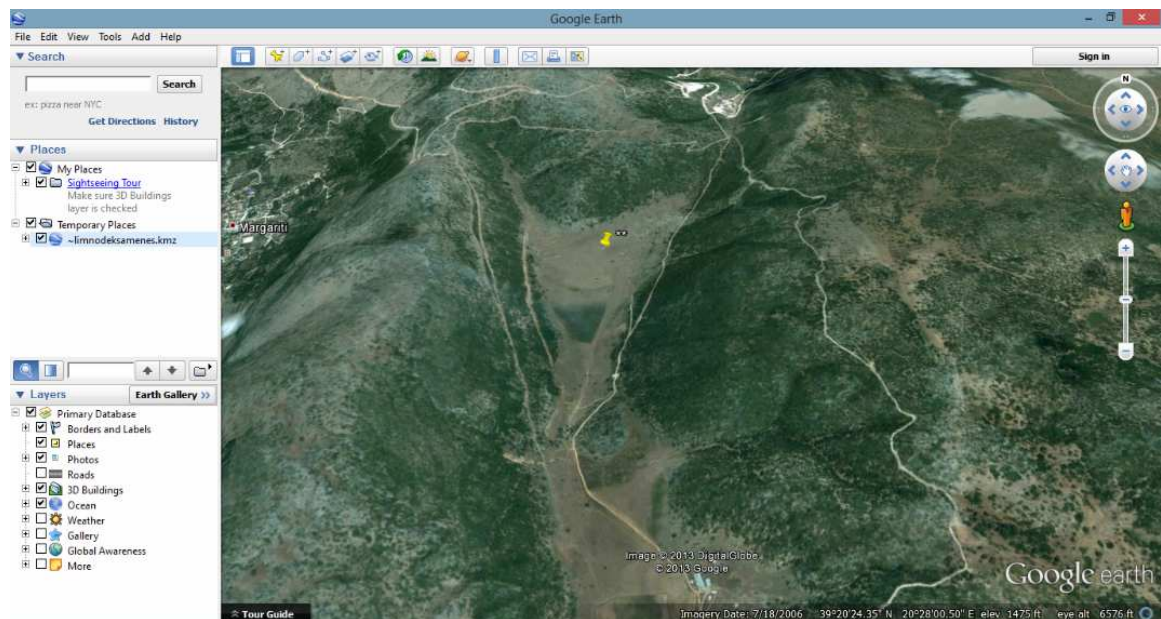
Έχοντας τις γεωγραφικές συντεταγμένες ενός σημείου μπορούμε πλέον να το εντοπίσουμε στον χάρτη. Αυτό το κάνουμε με την υπορουτίνα $entopismos$. Αυτή δημιουργεί ένα αρχείο .kml το οποίο μπορεί να εισαχθεί στο πρόγραμμα Google Earth. Στην υπορουτίνα $entopismos$ εισάγεται η θέση στον πίνακα του σημείου που θέλουμε να εντοπίσουμε και αφού καλέσει την υπορουτίνα I_f και πάρει τις γεωγραφικές συντεταγμένες του, τις τυπώνει με κατάλληλο τρόπο στο αρχείο .kml. Τέλος εισάγουμε το αρχείο αυτό στο πρόγραμμα google earth και εντοπίζουμε τη θέση του σημείου που ζητάμε σε αυτό.

Ως παράδειγμα καλούμε την υπορουτίνα $entopismos$ για το σημείο της 6777ης γραμμής και 550ης στήλης και παίρνουμε την θέση στο google earth που φαίνεται στην εικόνα 6.6.



Εικόνα 6.6. Θέση στον χάρτη ενός αποτελέσματος της τέταρτης μεθοδολογίας

Το σημείο έχει γεωμετρικό μήκος $\lambda=20,4571$ και γεωγραφικό πλάτος $\varphi=39,35458$ σε δεκαδικές μοίρες και αντιστοιχεί σε μια περιοχή της δυτικής Μακεδονίας όπως φαίνεται και στην εικόνα από το πρόγραμμα google earth.



Εικόνα 6.7. Απεικόνιση αποτελέσματος της τέταρτης μεθόδου από το πρόγραμμα google earth

Όπως βλέπουμε στην εικόνα 6.8 το σημείο βρίσκεται σε ορεινή περιοχή, αυτός είναι και ο λόγος που τα υψόμετρα αυξάνουν συνεχώς γύρω από αυτό. Η περιοχή ικανοποιεί τις προϋποθέσεις της τέταρτης μεθοδολογία και σωστά βρίσκεται στα αποτελέσματα του προγράμματος, αφού θα μπορούσε να αποτελεί θέση λιμνοδεξαμενής λόγω της μορφολογίας που έχει. Ακόμα παρατηρούμε ότι επαληθεύονται οι μηκοτομές καθώς και η μορφή των ισουψών καμπυλών που παρουσιάστηκαν παραπάνω και είχαν προκύψει από τα δεδομένα του SRTM.

Εντοπίζοντας, λοιπόν, τα σημεία των αποτελεσμάτων στο χάρτη καταλήγουμε ότι τα δεδομένα του SRTM συμφωνούν με τα πραγματικά, καθώς επίσης και στο ότι τα αποτελέσματα όντως αντιστοιχούν στη μορφή του εδάφους που αναζητάμε.

7. ΠΡΟΟΠΤΙΚΗ ΑΠΕΙΚΟΝΙΣΗ

7.1 ΓΕΝΙΚΑ

Στο κεφάλαιο αυτό παρουσιάζεται η μέθοδος που χρησιμοποιήθηκε για την προοπτική απεικόνιση των θέσεων λιμνοδεξαμενών που υπολογίστηκαν από τα δεδομένα του ψηφιακού μοντέλου εδάφους SRTM.

7.2 ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΟΠΤΙΚΗΣ ΑΚΤΙΝΑΣ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ POV-RAY

Η παρακολούθηση της πορείας οπτικής ακτίνας (ray tracing) είναι μια τεχνική για τη δημιουργία μιας εικόνας με τον εντοπισμό της διαδρομής του φωτός από την οπτική πηγή προς συγκεκριμένα εικονοστοιχεία (pixels) στο επίπεδο της εικόνας, προσομοιώνοντας τα φαινόμενα που δημιουργούνται από την πρόσκρουση του φωτός σε (εικονικά) αντικείμενα. Η τεχνική είναι σε θέση να παράγει υψηλό βαθμό οπτικού ρεαλισμού, συνήθως υψηλότερο από άλλες μεθόδους αλλά με μεγαλύτερο υπολογιστικό κόστος. Αυτό κάνει το ray tracing καταλληλότερο για εφαρμογές όπου ο χρόνος δημιουργίας της εικόνας δεν είναι κρίσιμος, αλλά ακατάλληλο για εφαρμογές σε πραγματικό χρόνο όπου η ταχύτητα είναι κρίσιμη (όπως στα βιντεοπαιχνίδια). Το ray tracing μπορεί να προσομοιώσει πολλά οπτικά φαινόμενα όπως η αντανάκλαση, η διάθλαση, η σκέδαση και η διασπορά του φωτός, επιτυγχάνοντας έτσι το φωτορεαλισμό [5]

Για λόγους οικονομίας υπολογισμών το ray tracing ξεκινά την παρακολούθηση της διαδρομής του φωτός αντίστροφα, δηλαδή από το pixel της εικόνας προς το κοντινότερο αντικείμενο, το οποίο μπορεί να φωτίζεται ή όχι. Το αντικείμενο επιδρά στο φως και το χρώμα της οπτικής πηγής ανάλογα με την υφή του, δηλαδή ανάλογα με τις φυσικές ιδιότητες που έχει η επιφάνειά του. Αναλόγως υπολογίζεται η απόχρωση και η ένταση του χρώματος στο pixel. Επειδή οι περισσότερες ακτίνες από την οπτική πηγή δεν προσκρούουν στην εικόνα, η αντίστροφη μέθοδος είναι πολύ πιο γρήγορη από την ευθεία (photon mapping). Επίσης το ray tracing μπορεί να επιταχυνθεί περαιτέρω χρησιμοποιώντας παράλληλη επεξεργασία [2].

Το πρόγραμμα H/Y Persistence of Vision Raytracer, ή POV-Ray, χρησιμοποιεί τη τεχνική ray tracing για τη δημιουργία φωτορεαλιστικών εικόνων. Είναι διαθέσιμο για διάφορες πλατφόρμες υπολογιστών όπως Linux, MacOS και Windows. Αναπτύχθηκε αρχικά από τους

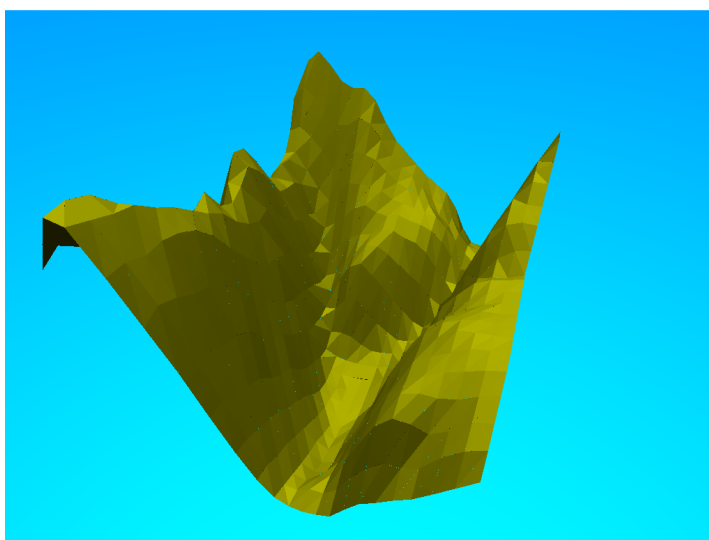
David Kirk Buck και Aaron A. Collins, και πλέον εξελίσσεται από πολλούς προγραμματιστές σε όλη τη γη [6]. Το POV-Ray διατίθεται δωρεάν μαζί με τον πηγαίο του κώδικα.

7.3 ΕΦΑΡΜΟΓΗ

Η μέθοδος που περιγράφηκε εφαρμόστηκε για την προοπτική απεικόνιση των θέσεων λιμνοδεξαμενών που υπολογίστηκαν με το πρόγραμμα limnodeksameni3. Για κάθε θέση χρησιμοποιούμε την υπορουτίνα upsometra για να δημιουργήσουμε ένα αρχείο που περιέχει τα υψόμετρα των σημείων που ανήκουν στην περιοχή της λιμνοδεξαμενής καθώς και τις γεωγραφικές τους συντεταγμένες από το ψηφιακό μοντέλο εδάφους SRTM.

Με το πρόγραμμα ron2syn τα σημεία αυτά μετατρέπονται στο Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς (ΕΓΣΑ 87) προκειμένου να είναι συμβατά με άλλα διατιθέμενα στοιχεία. Στη συνέχεια γίνεται τριγωνοποίηση Delaunary [4] προκειμένου να γίνει το πλέγμα (mesh) που χρειάζεται για το πρόγραμμα POV-RAY. Τέλος οι διαφορές στα υψόμετρα γίνονται πιο έντονες προκειμένου να είναι πιο ευκρινείς στην εικόνα που προκύπτει από το πρόγραμμα POV-RAY.

Ως παράδειγμα εφαρμογής η παραπάνω διαδικασία εφαρμόστηκε για το σημείο της 6777ης γραμμής και 550ης στήλης που υπολογίστηκε στα αποτελέσματα του προγράμματος limnodeksameni3. Η εικόνα που δημιουργήθηκε φαίνεται στο σχήμα 7.1 .



Σχήμα 7.1. προοπτική απεικόνιση ενός αποτελέσματος της τρίτης μεθοδολογίας

Πρόκειται για το ίδιο σημείο για το οποίο παρουσιάστηκαν στην παράγραφο 5.5 μηκοτομές και εικόνες από το πρόγραμμα google earth και όπως βλέπουμε η προοπτική απεικόνιση της περιοχής συμφωνεί με αυτές, αφού σχηματίζεται κοιλότητα που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή.

Περισσότερες φωτογραφίες όπου απεικονίζουν προοπτικά θέσεις που υπολογίστηκαν με το πρόγραμμα limnodeksameni3 παρουσιάζονται στο επόμενο κεφάλαιο όπου υπάρχουν ορισμένα παραδείγματα, καθώς και στο παράρτημα Α.3.3. Όπως θα δούμε και στα επόμενα παραδείγματα υπάρχει αντιστοιχία των εικόνων από το πρόγραμμα POV-RAY με αυτές από το προγράμματος google earth, όπως στην περίπτωση που παρουσιάστηκε παραπάνω. Επαληθεύεται έτσι ξανά η ορθότητα των δεδομένων του SRTM και των αποτελεσμάτων που προέκυψαν από την επεξεργασία τους.

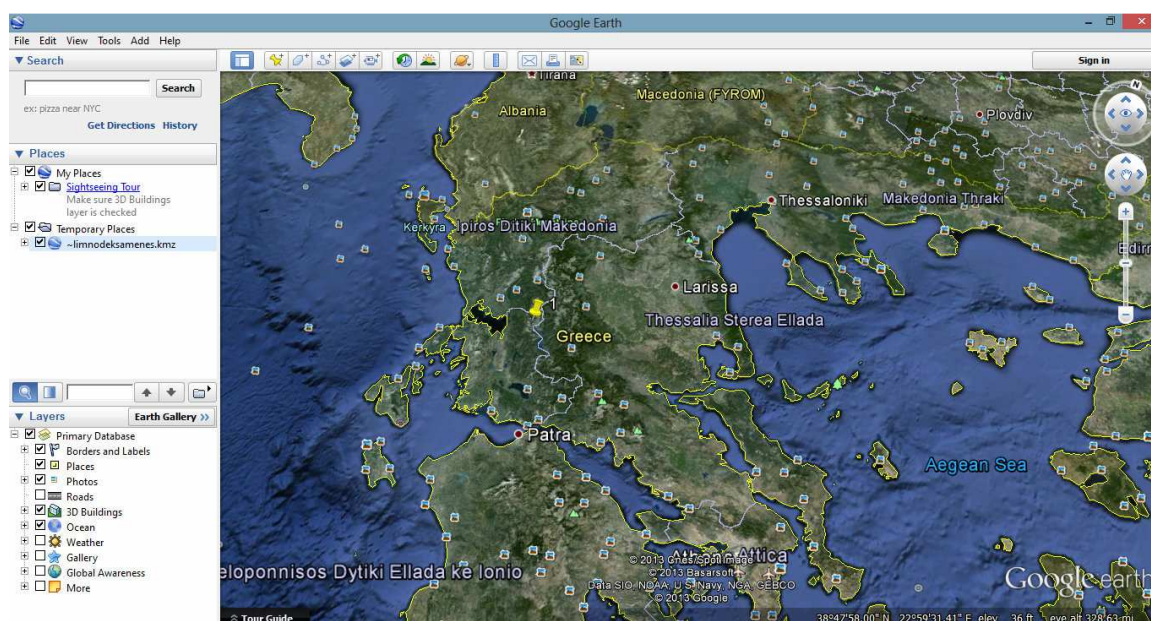
8. ΠΑΡΑΔΕΙΓΜΑΤΑ

8.1 Γενικά

Στο κεφάλαιο αυτό παρουσιάζονται ορισμένα παραδείγματα από τα αποτελέσματα που προέκυψαν με το πρόγραμμα limnodeksameni3 που υλοποιεί την τέταρτη μεθοδολογία. Για καθένα από τα παραδείγματα αυτά παρουσιάζονται δυο μηκοτομές, μια που περιέχει σημεία της ίδιας στήλης και με σημεία της ίδιας γραμμής με το κεντρικό σημείο που υπολογίστηκε. Παρουσιάζεται ακόμα η προοπτική απεικόνιση της περιοχής όπως προέκυψε από το πρόγραμμα POV-RAY χρησιμοποιώντας τα δεδομένα του SRTM και τέλος εικόνες της θέσης που εντοπίστηκε στο πρόγραμμα google earth.

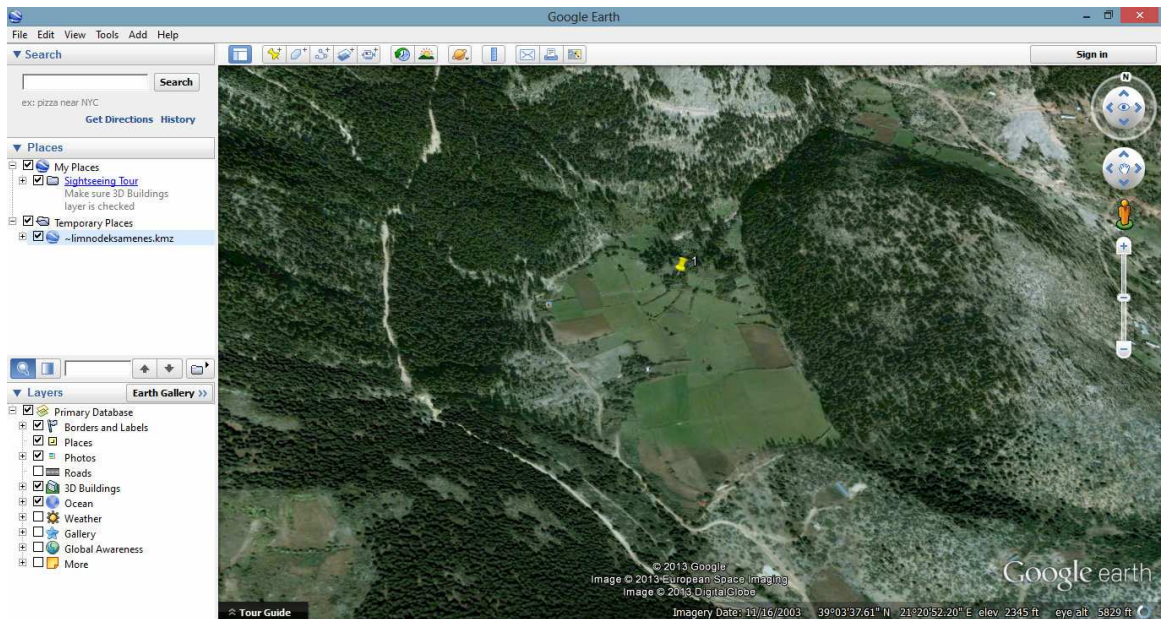
8.2 ΠΑΡΑΔΕΙΓΜΑ1

Στο πρώτο παράδειγμα παρουσιάζεται μια ορεινή περιοχή, όπως αυτή που παρουσιάστηκε στην παράγραφο 6.5. Το σημείο ανήκει στην 1621η στήλη και 7129η γραμμή του πίνακα και γύρω του βρέθηκαν 9 περίμετροι των οποίων τα υψόμετρα ικανοποιούσαν τη ζητούμενη μορφολογία, ενώ αντιστοιχεί σε όγκο 5.800.000 κυβικών μέτρων.



Εικόνα 8.1. Το πρώτο παράδειγμα στο χάρτη από το πρόγραμμα google earth

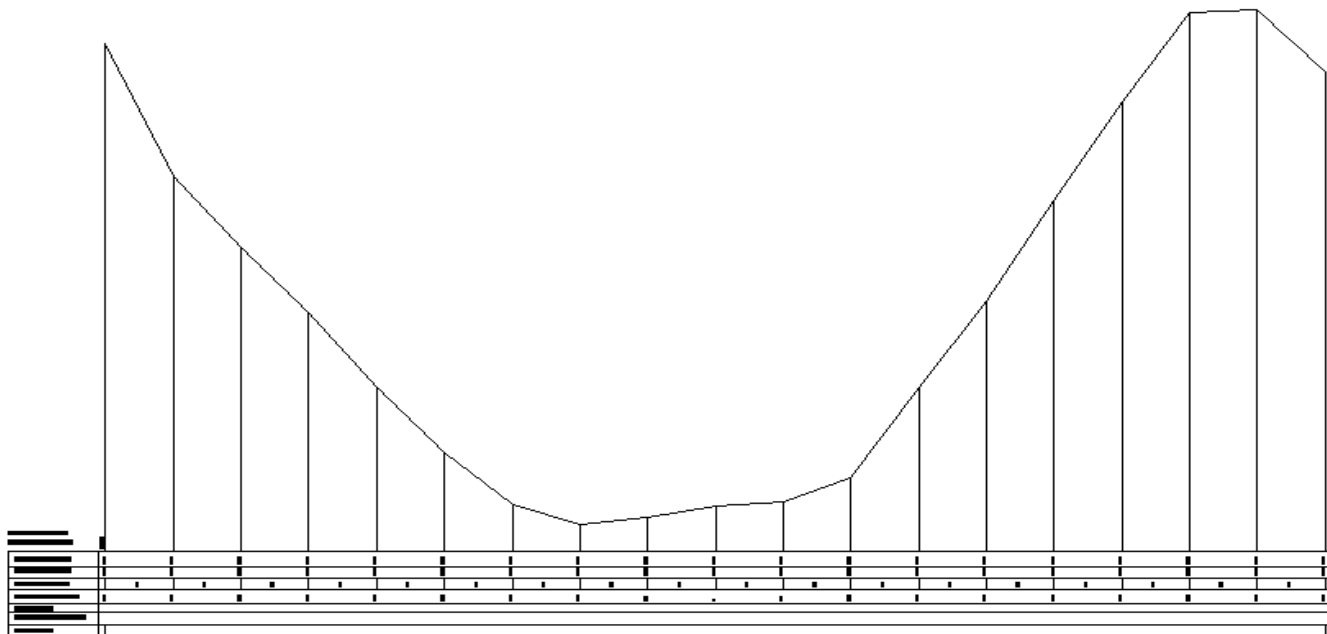
Το σημείο αυτό έχει γεωγραφικό μήκος $\lambda = 21.34958$ και γεωγραφικό πλάτος $\varphi = 39.06125$ και πρόκειται για σημείο της κεντρικής Ελλάδας όπως φαίνεται στην εικόνα 6.1 από το πρόγραμμα google earth.



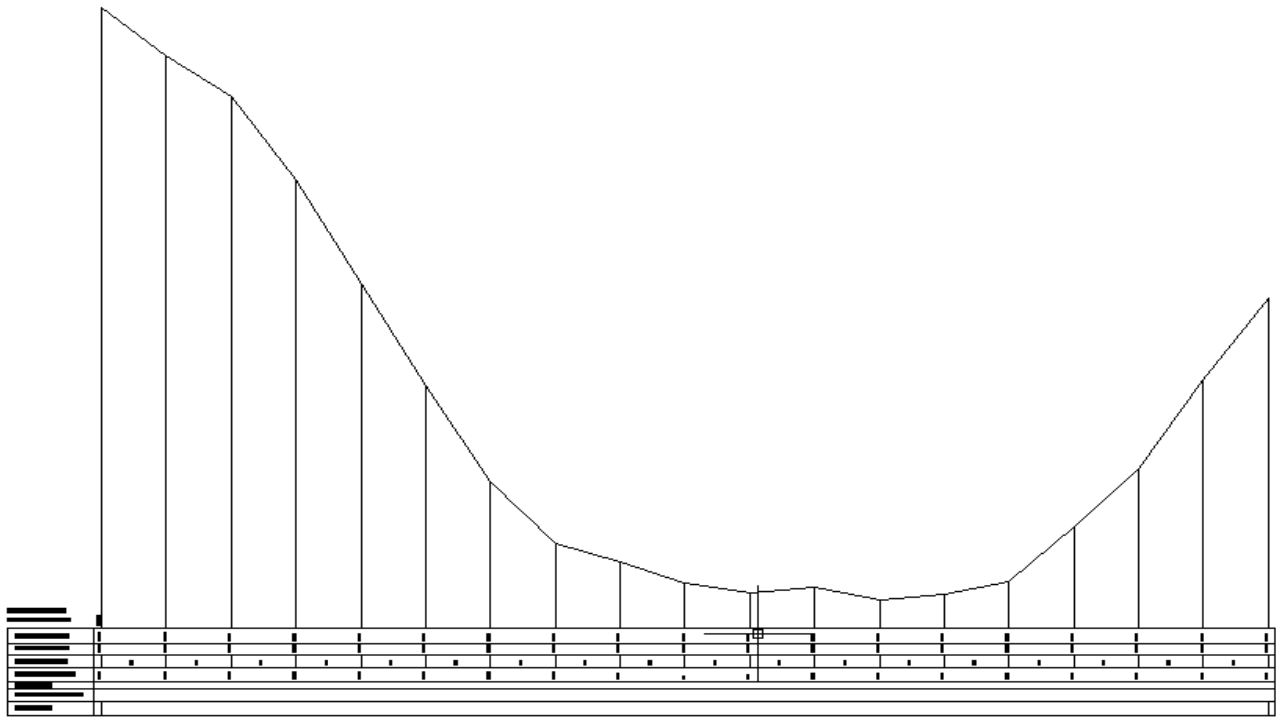
Εικόνα 8.2. Το παράδειγμα1 στο πρόγραμμα google earth

Όπως φαίνεται στην εικόνα 8.2 το κεντρικό σημείο βρίσκεται ανάμεσα σε βουνά, αυτός είναι και ο λόγος που τα υψόμετρα του συνεχώς αυξάνουν καθώς απομακρυνόμαστε από αυτό.

Στα σχήματα 8.3 και 8.4 παρουσιάζονται οι μηκοτομές των σημείων της ίδιας στήλης και της ίδιας γραμμής με το πρώτο παράδειγμα. Παρατηρούμε, λοιπόν, ότι και οι δυο μηκοτομές έχουν τη μορφή που αναμέναμε, καθώς τα υψόμετρα αυξάνουν εκατέρωθεν του κεντρικού σημείου, και καθώς απομακρυνόμαστε από αυτό.

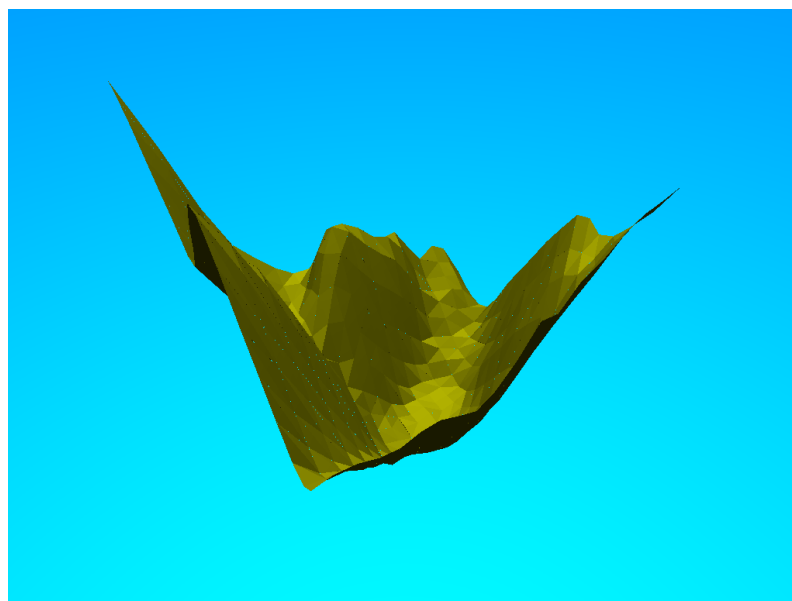


Σχήμα 8.3. Μηκοτομή σημείων της ίδιας σειράς για το πρώτο παράδειγμα

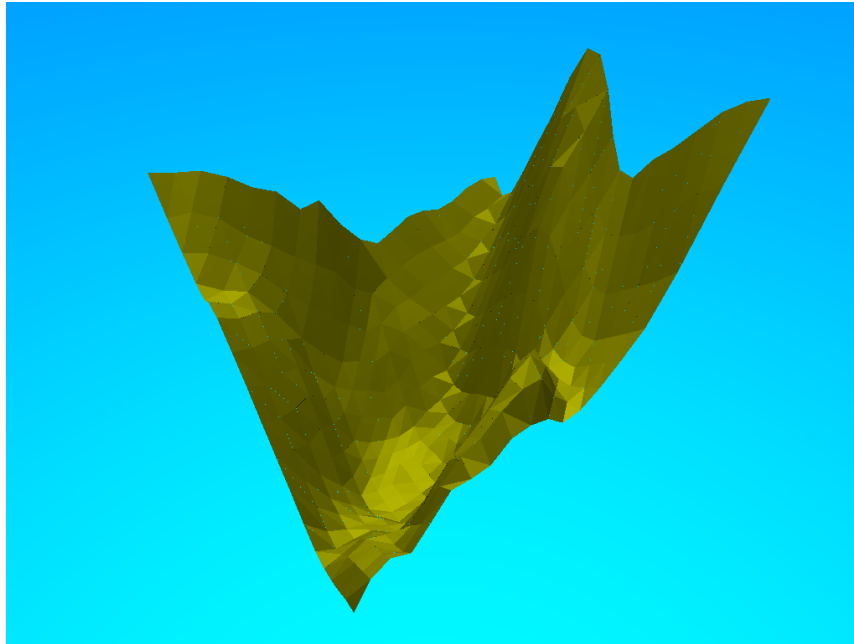


Σχήμα 8.4. Μηκοτομή σημείων της ίδιας γραμμής για το πρώτο παράδειγμα

Τέλος παρουσιάζεται η προοπτική απεικόνιση της περιοχής από φωτογραφίες που προέκυψαν από το πρόγραμμα POV-RAY(εικόνες 8.5, 8.6). Όπως βλέπουμε αυτές αντιστοιχούν με τις εικόνες από το πρόγραμμα google earth και παρουσιάζουν μορφολογία εδάφους που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή.



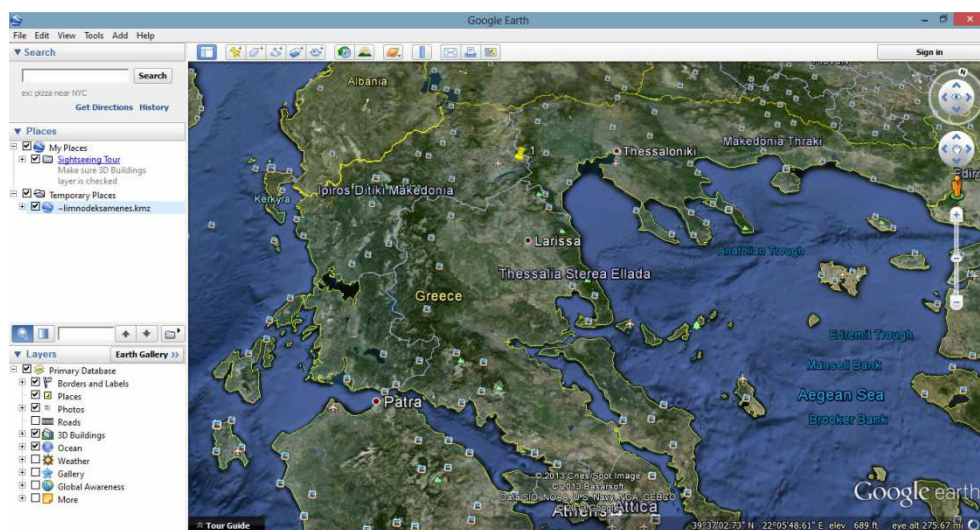
Εικόνα 8.5. προοπτική απεικόνιση της περιοχής του πρώτου παραδείγματος



Εικόνα 8.6. προοπτική απεικόνιση της περιοχής του πρώτου παραδείγματος

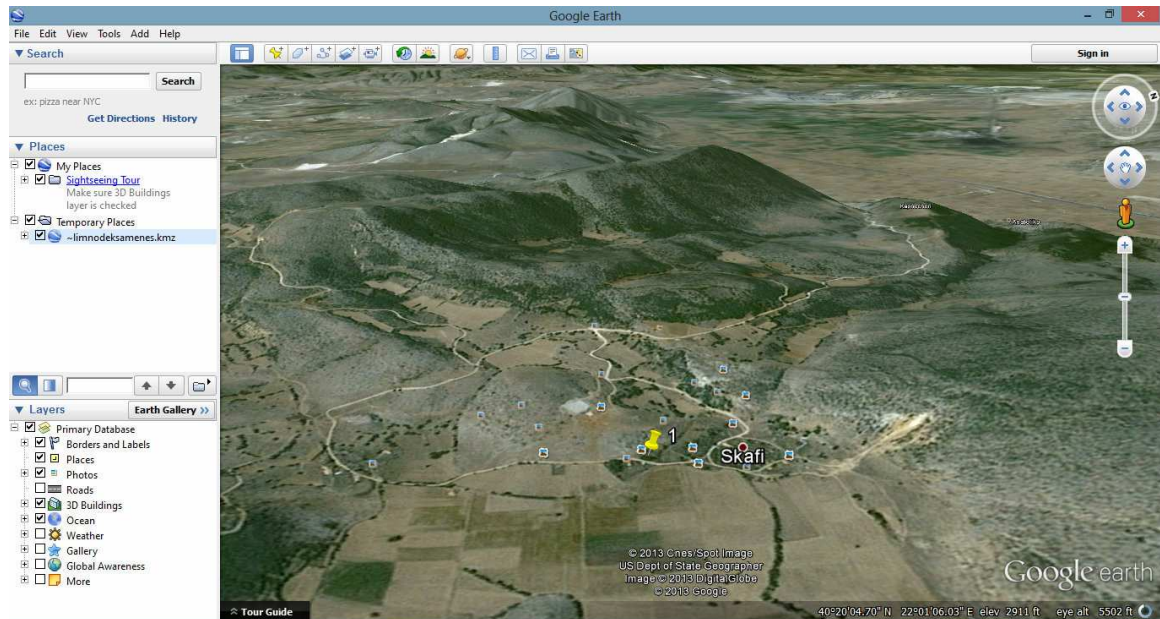
8.3 ΠΑΡΑΔΕΙΓΜΑ2

Το δεύτερο παράδειγμα που παρουσιάζεται πρόκειται για μια μεγαλύτερης έκτασης περιοχή. Πρόκειται για το σημείο της 2421^{ης} στήλης και 5600^{ης} γραμμής και βρέθηκε να έχει 12 περιμέτρους των οποίων τα σημεία ικανοποιούν τη μεθοδολογία. Ο μεγάλος αριθμός περιμέτρων είναι ο λόγος για την εύρεση μεγάλου όγκου στη συγκεκριμένη περιοχή, ίσου με 29.500.000 κυβικά μέτρα.



Εικόνα 8.7. Το δεύτερο παράδειγμα στο χάρτη από το πρόγραμμα google earth

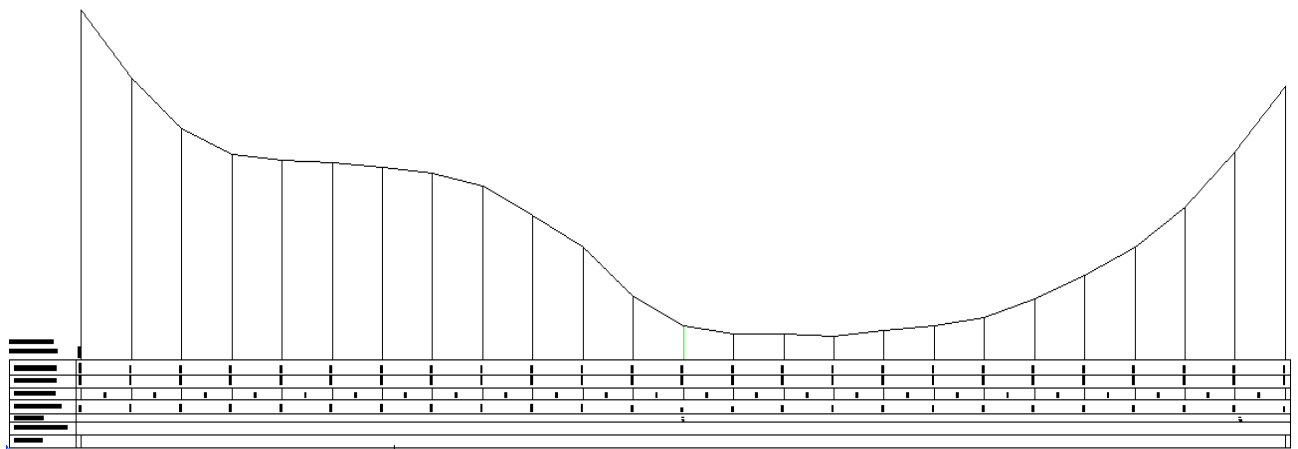
Το σημείο αυτό έχει γεωγραφικό μήκος $\lambda = 22.01625$ και γεωγραφικό πλάτος $\phi = 40.33458$ και πρόκειται για σημείο της κεντρικής Μακεδονίας όπως φαίνεται στην εικόνα 8.7 από το πρόγραμμα google earth.



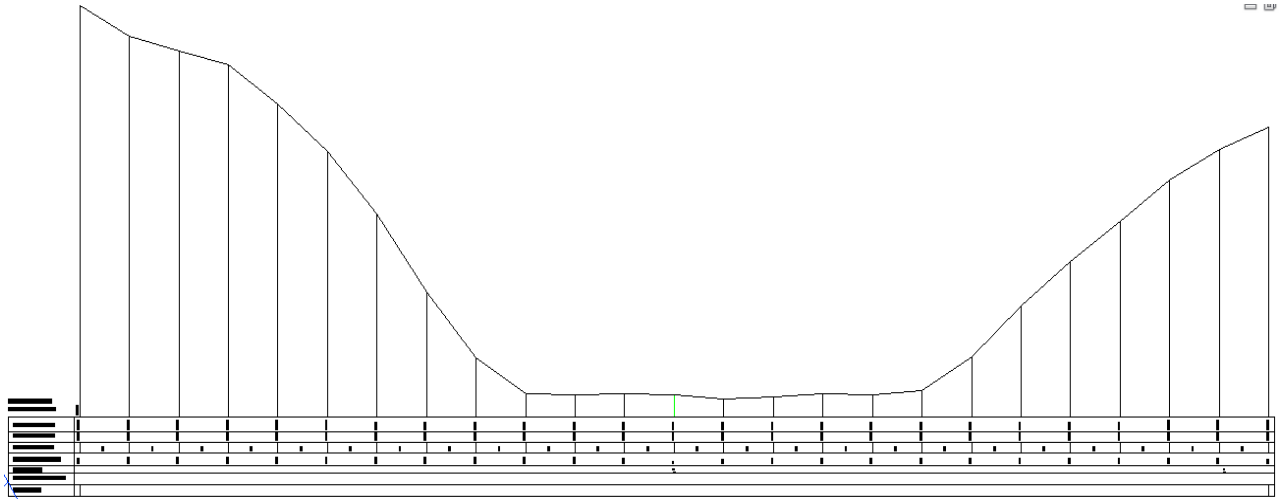
Εικόνα 8.8. Το δεύτερο παράδειγμα στο πρόγραμμα google earth

Όπως παρατηρούμε στην εικόνα 8.8 πρόκειται για ένα σημείο σε λοφώδη περιοχή με μορφολογία εδάφους που αντιστοιχεί σε λιμνοδεξαμενή. Η περιοχή άλλωστε έχει ονομασία “σκάφη” όπως βλέπουμε και στην εικόνα.

Οι μηκοτομές του σημείου φαίνονται στα σχήματα 8.9 και 8.10 και έχουν μορφή κοιλότητας με μικρές διακυμάνσεις.

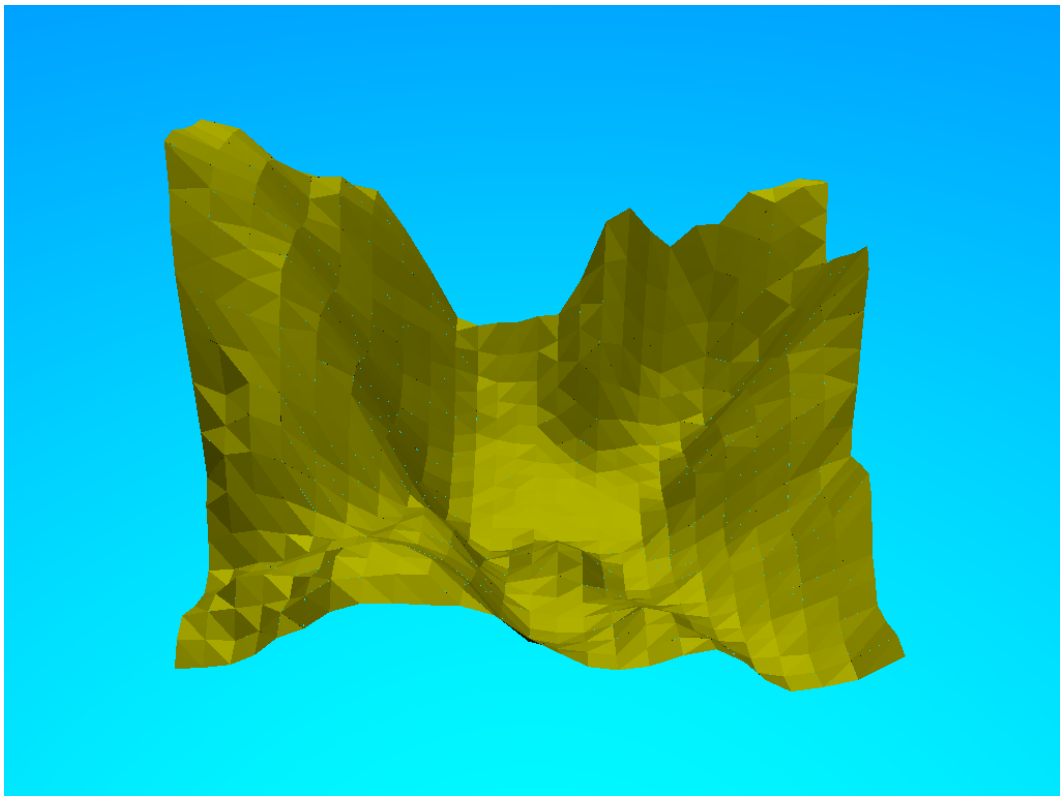


Σχήμα 8.9. Μηκοτομή σημείων της ίδιας σειράς για το δεύτερο παράδειγμα

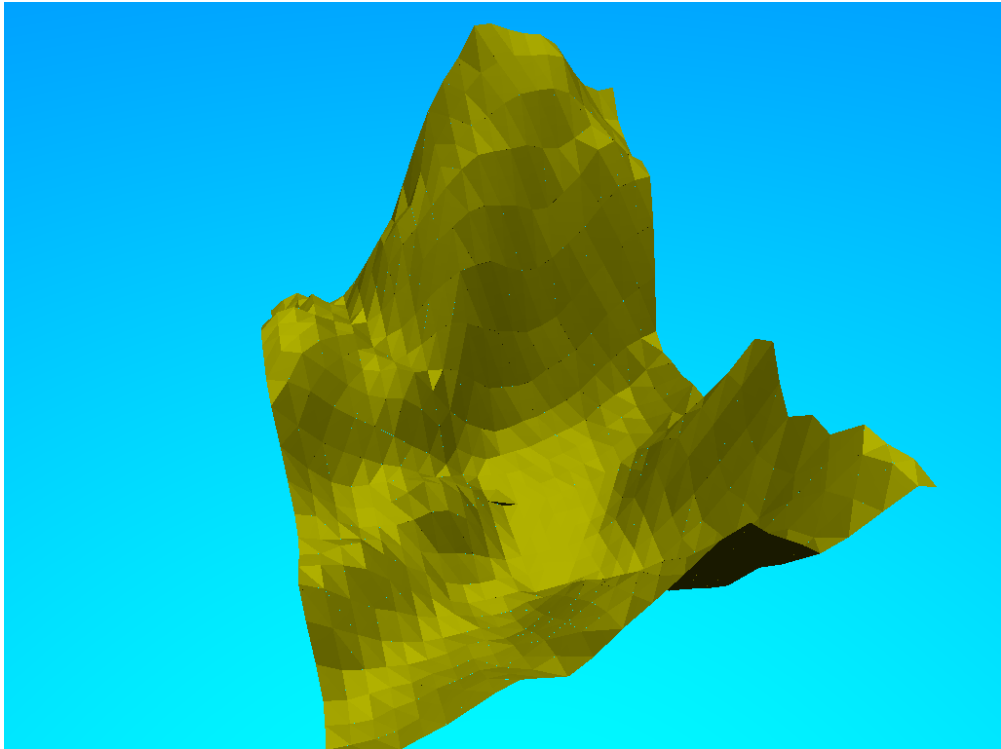


Σχήμα 8.10. Μηκοτομή σημείων της ίδιας γραμμής για το δεύτερο παράδειγμα

Τέλος, στην προοπτική απεικόνιση της περιοχής (εικόνες 8.11, 8.12), όπου μπορούμε να δούμε με περισσότερη λεπτομέρεια το ανάγλυφό της, παρατηρούμε ότι αυτό σχηματίζει κοιλάτητα μεγαλύτερης έκτασης.



Εικόνα 8.11. προοπτική απεικόνιση για το δεύτερο παράδειγμα

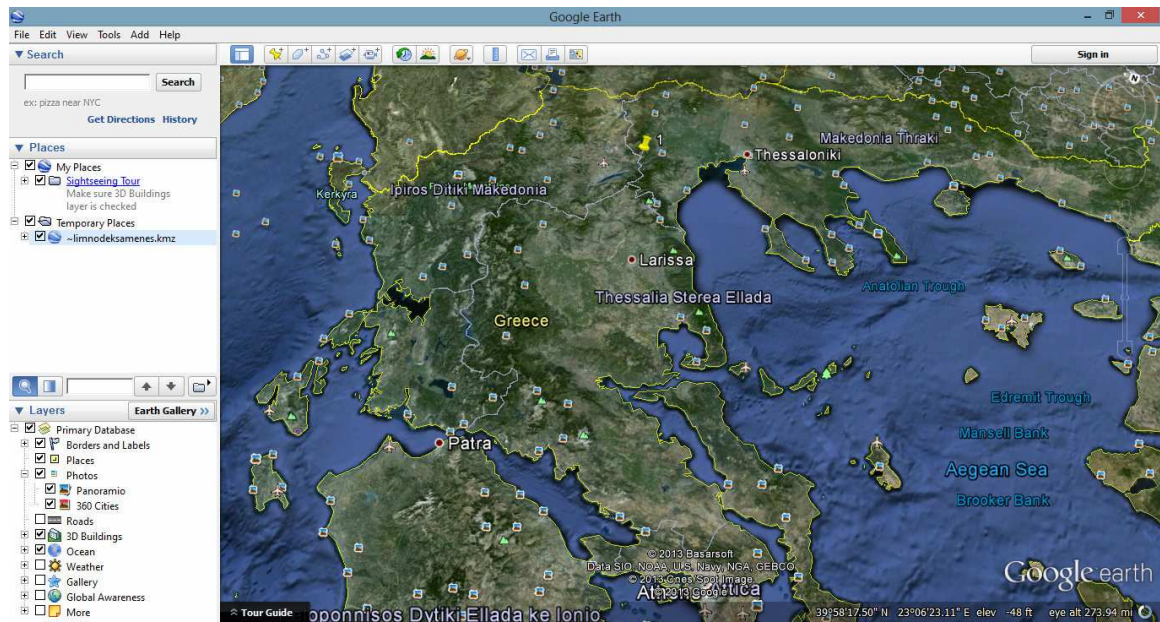


Εικόνα 8.12. προοπτική απεικόνιση για το δεύτερο παράδειγμα

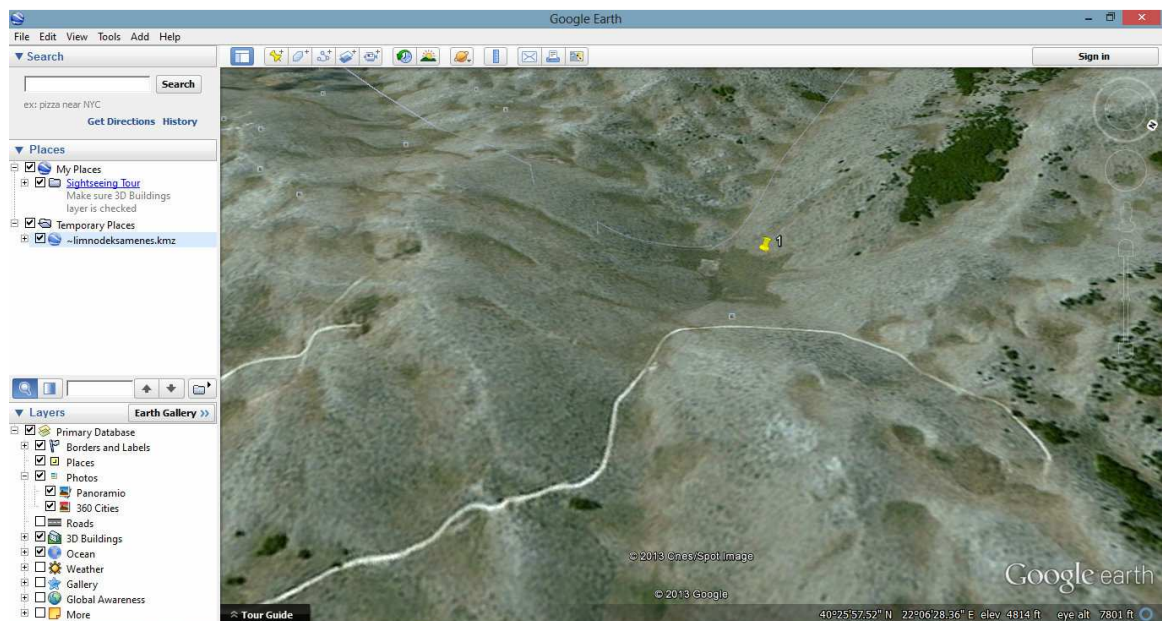
8.4 ΠΑΡΑΔΕΙΓΜΑ3

Στο τρίτο παράδειγμα που παρουσιάζεται φαίνεται πιο καθαρά η μορφολογία την οποία αναζητάμε, αφού έχει εντοπιστεί μια φυσική κοιλότητα στο έδαφος. Πρόκειται για το σημείο της 2527^{ns} στήλης και 5482^{ns} γραμμής και βρέθηκε να έχει 6 περιμέτρους των οποίων τα σημεία ικανοποιούν τη μεθοδολογία, ενώ αντιστοιχούσε σε όγκο 8.200.000 κυβικών μέτρων.

Το σημείο αυτό έχει γεωγραφικό μήκος $\lambda = 22.10458$ και γεωγραφικό πλάτος $\varphi = 40.43291$ και πρόκειται για σημείο της κεντρικής Μακεδονίας όπως φαίνεται στην εικόνα 8.13 από το πρόγραμμα google earth.



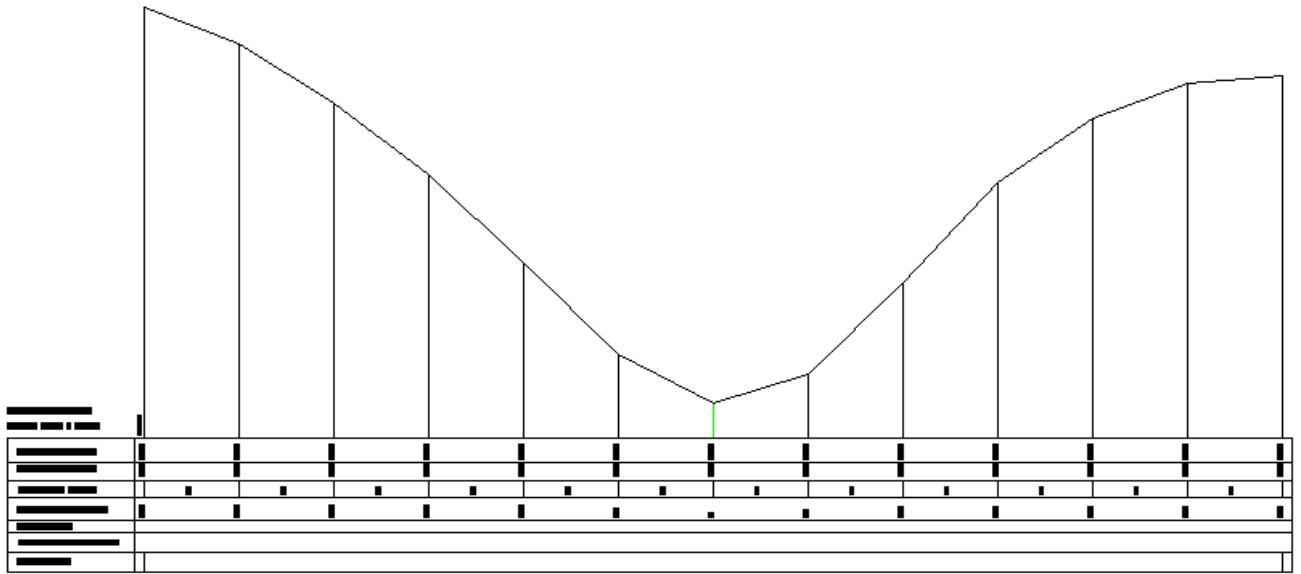
Εικόνα 8.13. Το τρίτο παράδειγμα στο χάρτη από το πρόγραμμα google earth



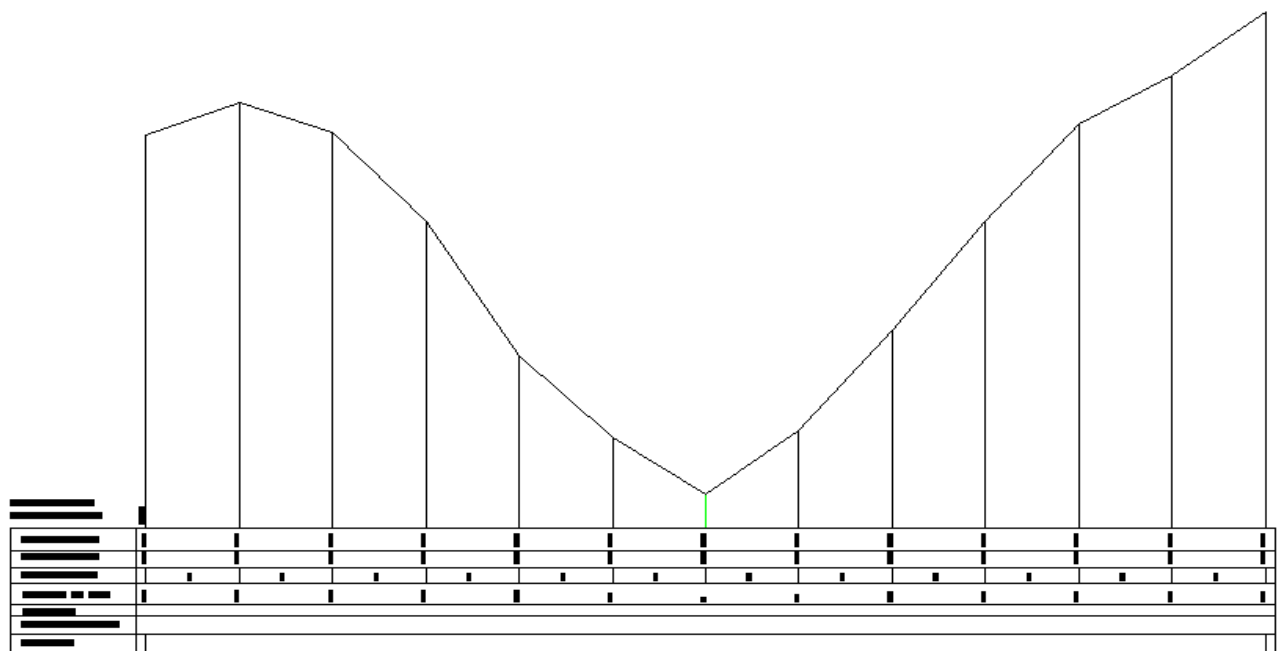
Εικόνα 8.14. Το τρίτο παράδειγμα στο χάρτη από το πρόγραμμα google earth

Από παρατήρηση της εικόνας 8.14 από το πρόγραμμα google earth μπορούμε να συμπεράνουμε ότι τα υψόμετρα γύρω από το σημείο και καθώς απομακρυνόμαστε από αυτό αυξάνουν, αποτελεί επομένως κλασσικό παράδειγμα κοιλάτητας που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή.

Οι μηκοτομές του σημείου φαίνονται στα σχήματα 8.15 και 8.16

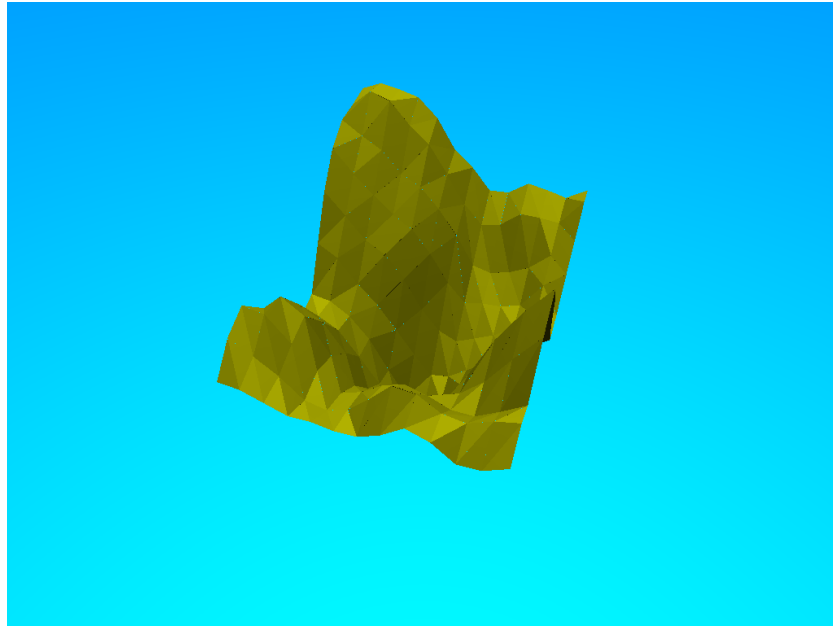


Σχήμα 8.15. Μηκοτομή σημείων της ίδιας γραμμής για το τρίτο παράδειγμα

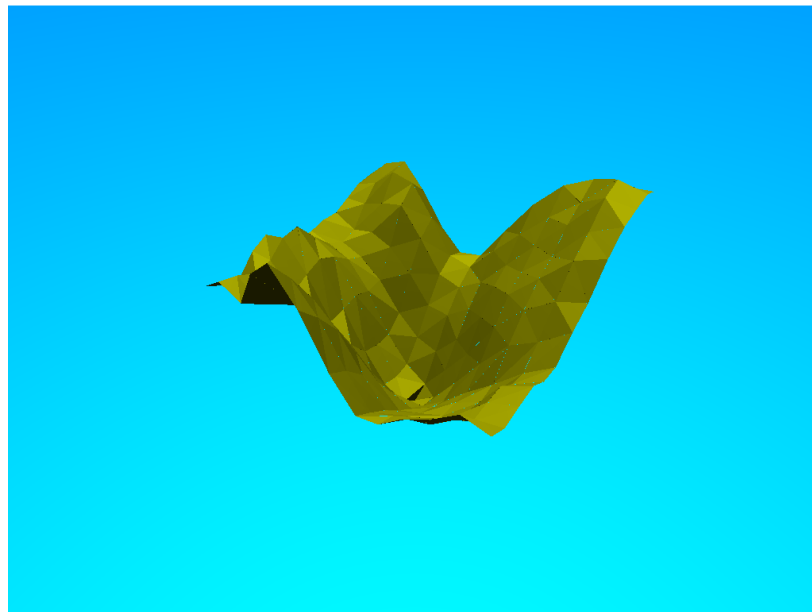


Σχήμα 8.16. Μηκοτομή σημείων της ίδιας στήλης για το τρίτο παράδειγμα

Στην προοπτική απεικόνιση του σημείου επαληθεύεται ξανά η μορφή που είχε εντοπιστεί προηγουμένως στο πρόγραμμα google earth και στις μηκοτομές (εικόνες 8.17 και 8.18).



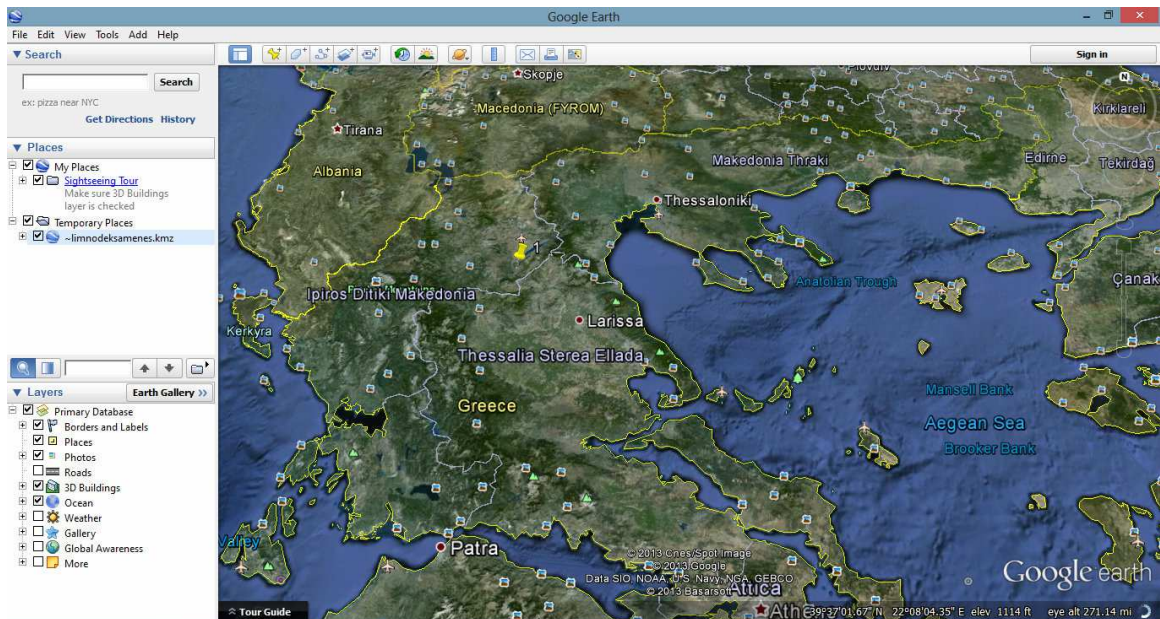
Εικόνα 8.17. προοπτική απεικόνιση για το τρίτο παράδειγμα



Εικόνα 8.18. προοπτική απεικόνιση για το τρίτο παράδειγμα

8.5 ΠΑΡΑΔΕΙΓΜΑ4

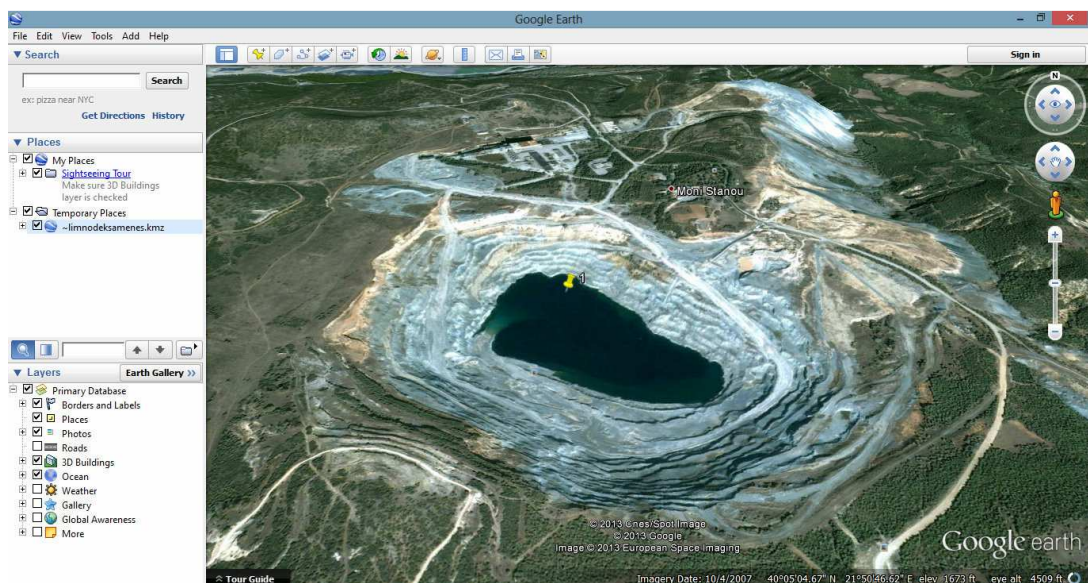
Με το παράδειγμα αυτό επαληθεύουμε την ορθότητα των αποτελεσμάτων, καθώς έχει εντοπιστεί μια θέση λιμνοδεξαμενής ήδη γεμάτης με νερό. Πρόκειται για το σημείο της 2217^{ης} στήλης και 5896^{ης} γραμμής και βρέθηκε να έχει 7 περιμέτρους των οποίων τα σημεία ικανοποιούν τη μεθοδολογία, ενώ αντιστοιχούσε σε όγκο 13.100.000 κυβικών μέτρων.



Εικόνα 8.19. Το τέταρτο παράδειγμα στο χάρτη από το πρόγραμμα google earth

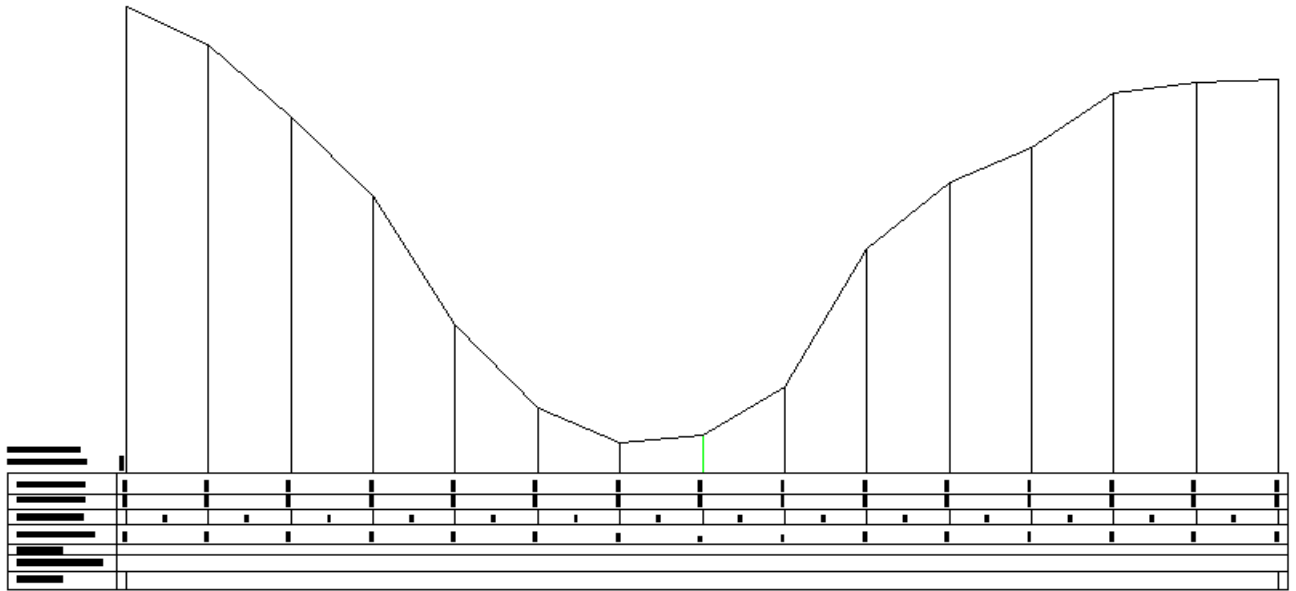
Το σημείο αυτό έχει γεωγραφικό μήκος $\lambda = 21.84625$ και γεωγραφικό πλάτος $\varphi = 40.08792$ και πρόκειται για σημείο της κεντρικής Μακεδονίας όπως φαίνεται στην εικόνα 8.19 από το πρόγραμμα google earth.

Στη εικόνα 8.20 βλέπουμε ότι πρόκειται για μια λιμνοδεξαμενή στην οποία υπάρχει νερό. Φαίνεται καθαρά έτσι ότι τα αποτελέσματα του προγράμματος αντιστοιχούν σε περιοχές που θα μπορούσαμε να εκμεταλλευτούμε ως λιμνοδεξαμενές.

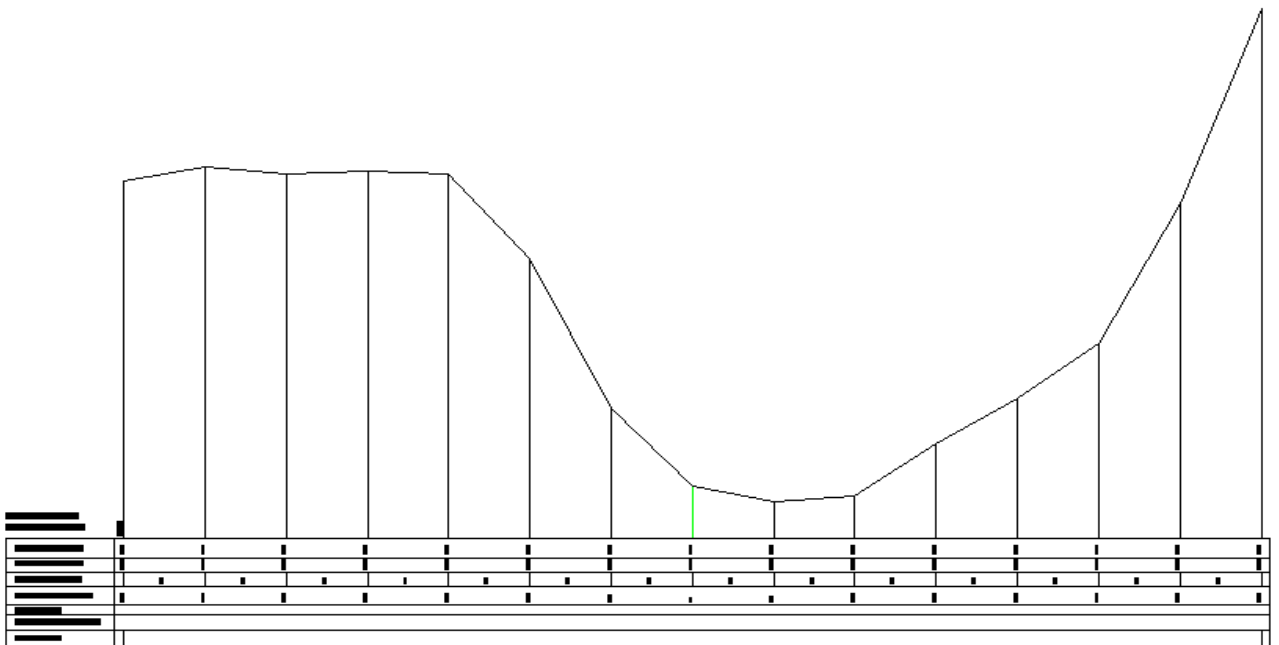


Εικόνα 8.20. Το τέταρτο παράδειγμα στο χάρτη από το πρόγραμμα google earth

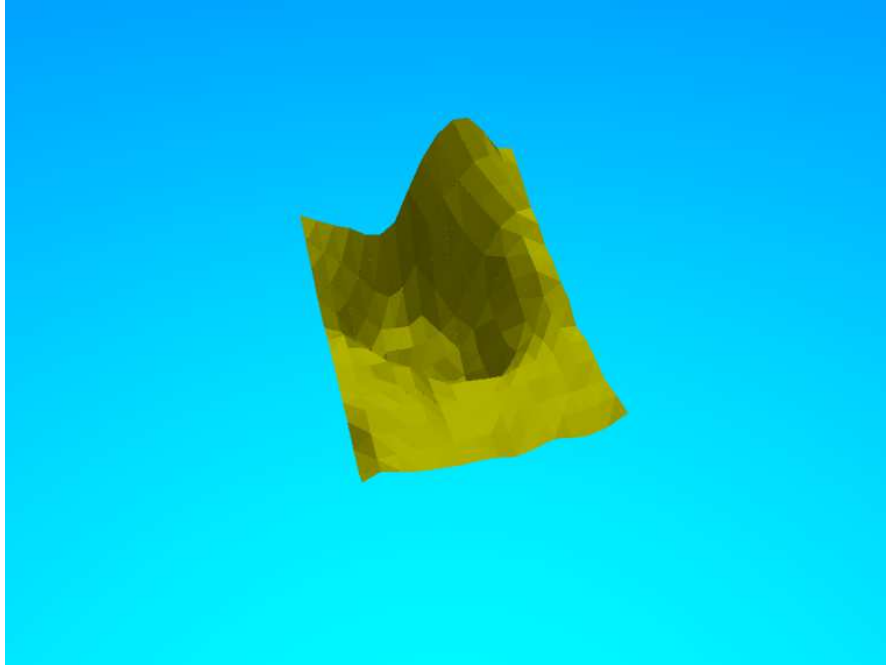
Παρακάτω παρουσιάζονται οι μηκοτομές του σημείου (σχήματα 8.21 και 8.22), καθώς και μια εικόνα της περιοχής από το πρόγραμμα POV-RAY (εικόνα 8.23).



Σχήμα 8.21. Μηκοτομή σημείων της ίδιας γραμμής για το τέταρτο παράδειγμα



Σχήμα 8.22. Μηκοτομή σημείων της ίδιας στήλης για το τέταρτο παράδειγμα



Εικόνα 8.23. προοπτική απεικόνιση για το τέταρτο παράδειγμα

9. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο κεφάλαιο αυτό παρουσιάζονται αρχικά τα σημαντικότερα συμπεράσματα της παρούσας διπλωματικής εργασίας και στη συνέχεια αναφέρονται προτάσεις για μελλοντική μελέτη και επέκταση του θέματος που πραγματεύεται.

Το ζητούμενο ήταν να μελετηθεί κατά πόσο είναι δυνατός ο υπολογισμός θέσεων λιμνοδεξαμενών και η προοπτική αποτύπωσή τους, χρησιμοποιώντας παγκόσμια ψηφιακά μοντέλα εδάφους. Συντάχθηκε πρόγραμμα H/Y σε γλώσσα FORTRAN, το οποίο επεξεργάστηκε τα δεδομένα του ψηφιακού μοντέλου SRTM. Με το πρόγραμμα αυτό εντοπίστηκαν αρκετές θέσεις με μορφολογία που θα μπορούσε να αντιστοιχεί σε λιμνοδεξαμενή στην περιοχή της Ελλάδας. Συμπεραίνουμε επομένως ότι τα δεδομένα του προγράμματος SRTM μπορούν να χρησιμοποιηθούν για τον εντοπισμό τέτοιων θέσεων.

Χρησιμοποιώντας το πρόγραμμα που συντάχθηκε και τα προγράμματα *ron2syn* και *ronray* δημιουργήθηκαν εικόνες όπου απεικονίζονται προοπτικά τέτοιες θέσεις χρησιμοποιώντας τα δεδομένα του προγράμματος SRTM. Συγκρίνοντας τις εικόνες αυτές με εικόνες από το πρόγραμμα *google earth* συμπεραίνουμε ακόμα ότι τα δεδομένα του SRTM αντιστοιχούν με τα πραγματικά.

Τέλος παρατίθενται κάποιες προτάσεις για μελλοντική έρευνα. Αρχικά θα μπορούσε να εφαρμοστεί η μεθοδολογία, και το πρόγραμμα που παρουσιάζεται στην παρούσα διπλωματική εργασία, χρησιμοποιώντας παγκόσμια ψηφιακά μοντέλα εδάφους διαφορετικά του SRTM, με σκοπό να επαληθευτούν τα αποτελέσματα και να εντοπιστούν επιπλέον θέσεις αν είναι δυνατόν λόγω διαφοράς στην ακρίβεια μεταξύ των μοντέλων. Επιπλέον ίσως θα μπορούσαν να αναπτυχθούν μεθοδολογίες για τον εντοπισμό πιο σύνθετων μορφών εδάφους που θα αντιστοιχούν σε λιμνοδεξαμενές. Ακόμη, θα μπορούσε να ελεγχθεί κατά πόσο είναι δυνατόν να εκμεταλλευτούμε τις θέσεις λιμνοδεξαμενών που έχουν υπολογιστεί. Για να γίνει κάτι τέτοιο απαιτείται αρχικά να εξεταστούν οι θέσεις αυτές και να διαπιστωθεί ότι η περιοχή της λιμνοδεξαμενής δεν έχει σημαντική οικονομική αξία (κατοικημένες περιοχές, τουριστικές περιοχές, εύφορες περιοχές κλπ), οικολογική αξία (προστατευόμενες περιοχές) και αρχαιολογική αξία. Ακόμα απαιτείται να ερευνηθούν βροχομετρικά στοιχεία των περιοχών για να διαπιστωθεί αν υπάρχει η απαιτούμενη μέση βροχόπτωση για να γεμίσει η λιμνοδεξαμενή και τέλος να γίνουν γεωλογικές έρευνες στις θέσεις που εντοπίστηκαν καθώς

χρειάζεται το έδαφος να μπορεί να συγκρατήσει το νερό, δηλαδή να έχει κάποιο αδιαπέρατο από το νερό υπόστρωμα.

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α

ΠΑΡΑΡΤΗΜΑ Α.1.1

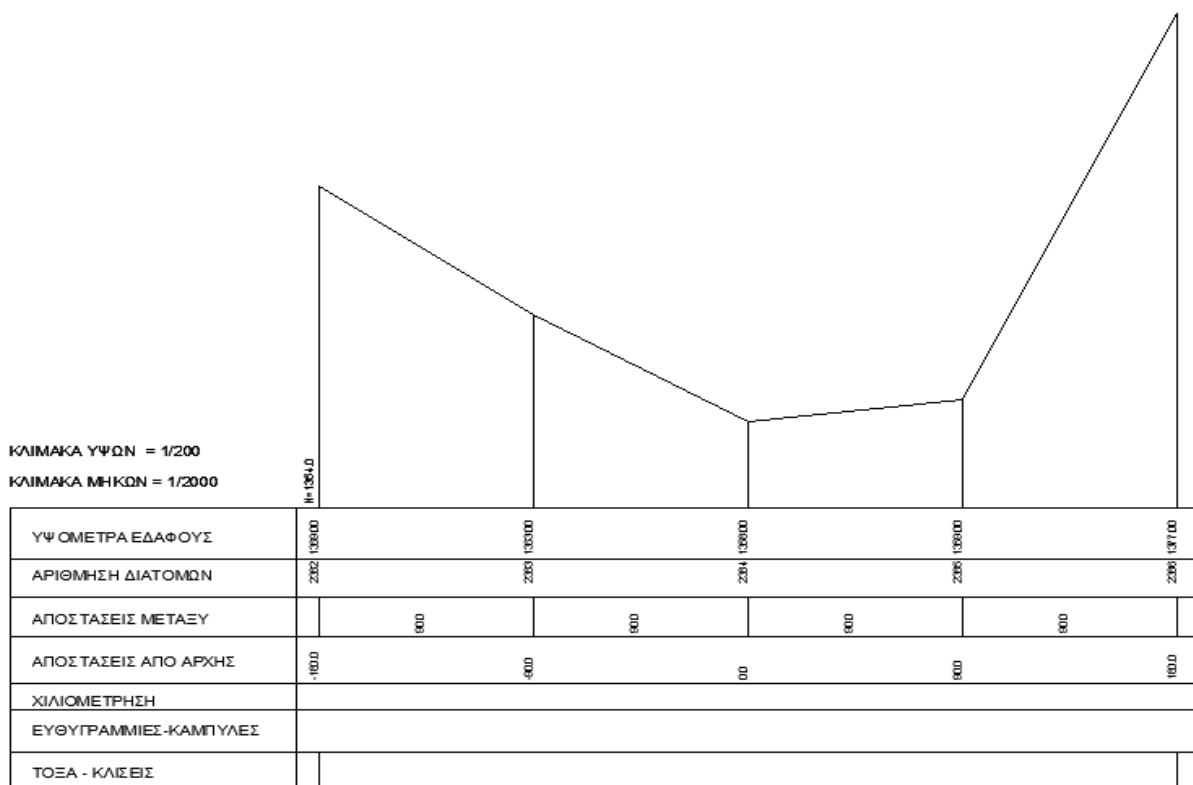
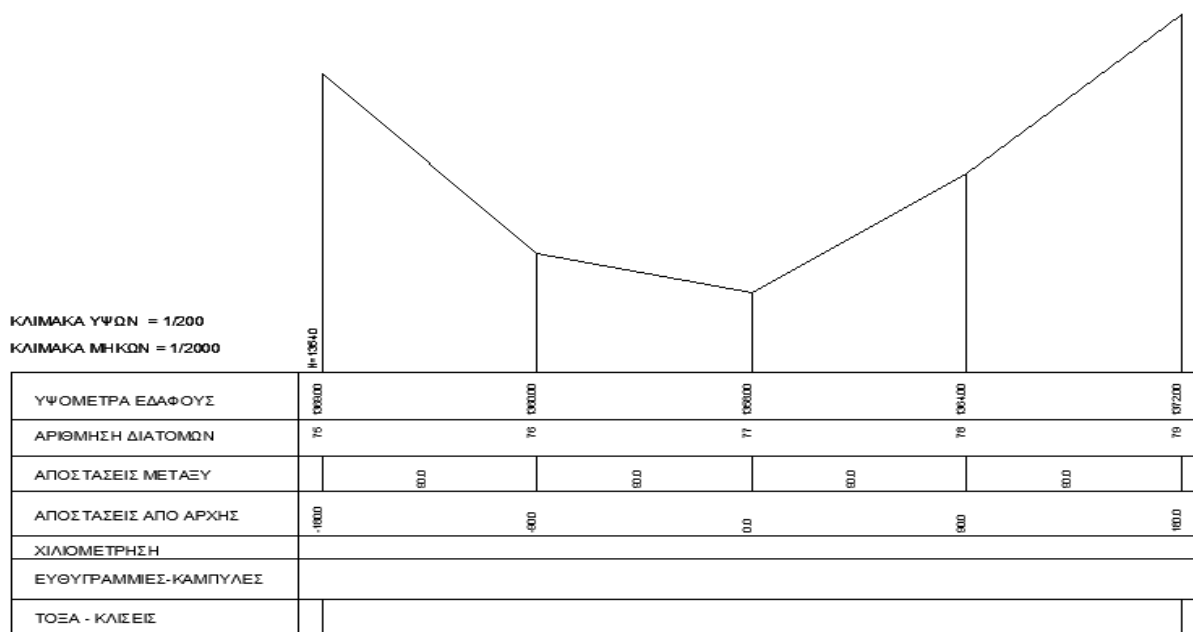
ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΠΡΩΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ limnodeksameni1

Στήλη πίνακα	Γραμμή πίνακα	Περίμετροι που ικανοποιούν
77	2384	2
236	15716	2
884	6078	2
1667	3973	2
1910	14763	2
1956	4964	2
2724	7836	2
3046	2331	2
3070	16648	2
3384	17894	2
3444	17672	2
5596	11624	2
8373	906	2
8439	847	2
8534	1211	2
8616	1669	3
8994	8479	2
9499	9457	2
9769	9423	2
10542	4821	2
10620	17013	2
10896	8212	2
11126	6502	2
11369	6427	2
11515	10193	2
11540	10133	2
11674	10210	2

ΠΑΡΑΡΤΗΜΑ Α.1.2

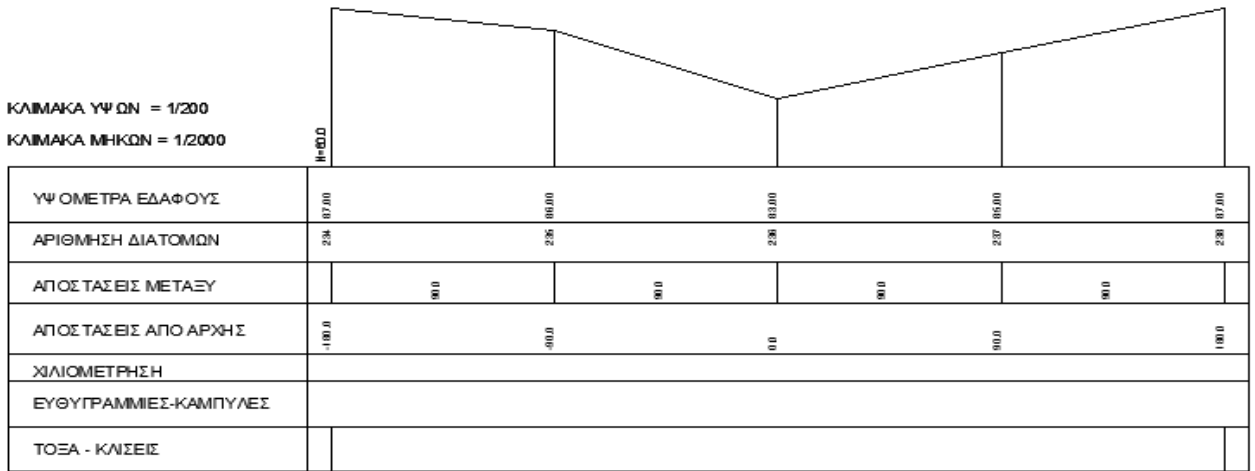
ΜΗΚΟΤΟΜΕΣ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΠΡΩΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

Μηκοτομές σημείου στήλης 77, γραμμής 2384

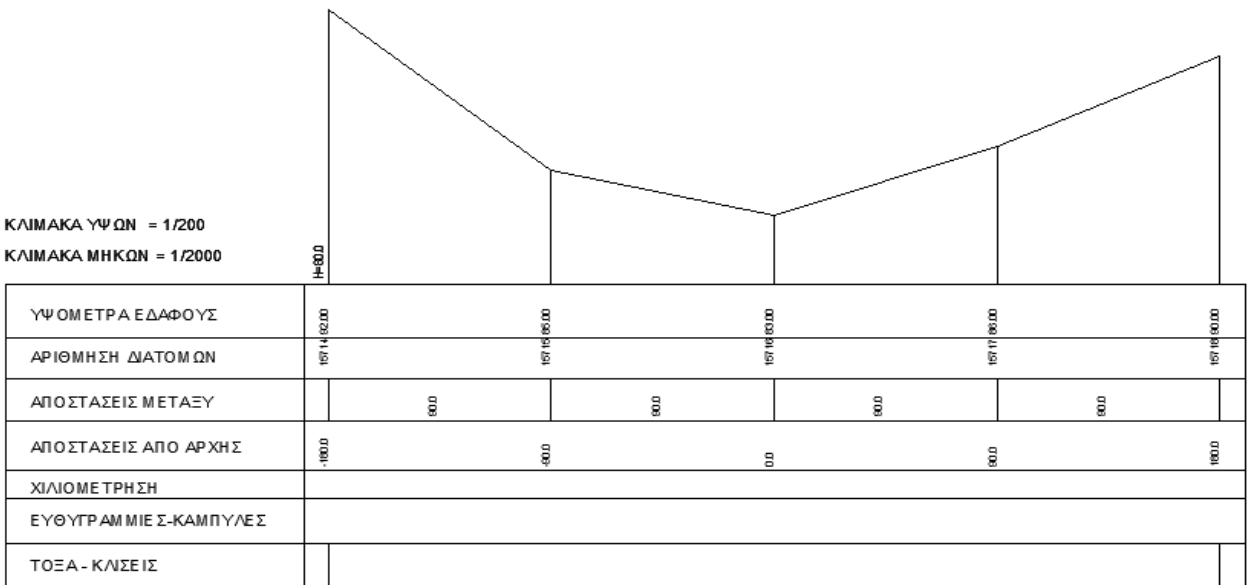


Μηκοτομές σημείου στήλης 236, γραμμής 15716

ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/200
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000

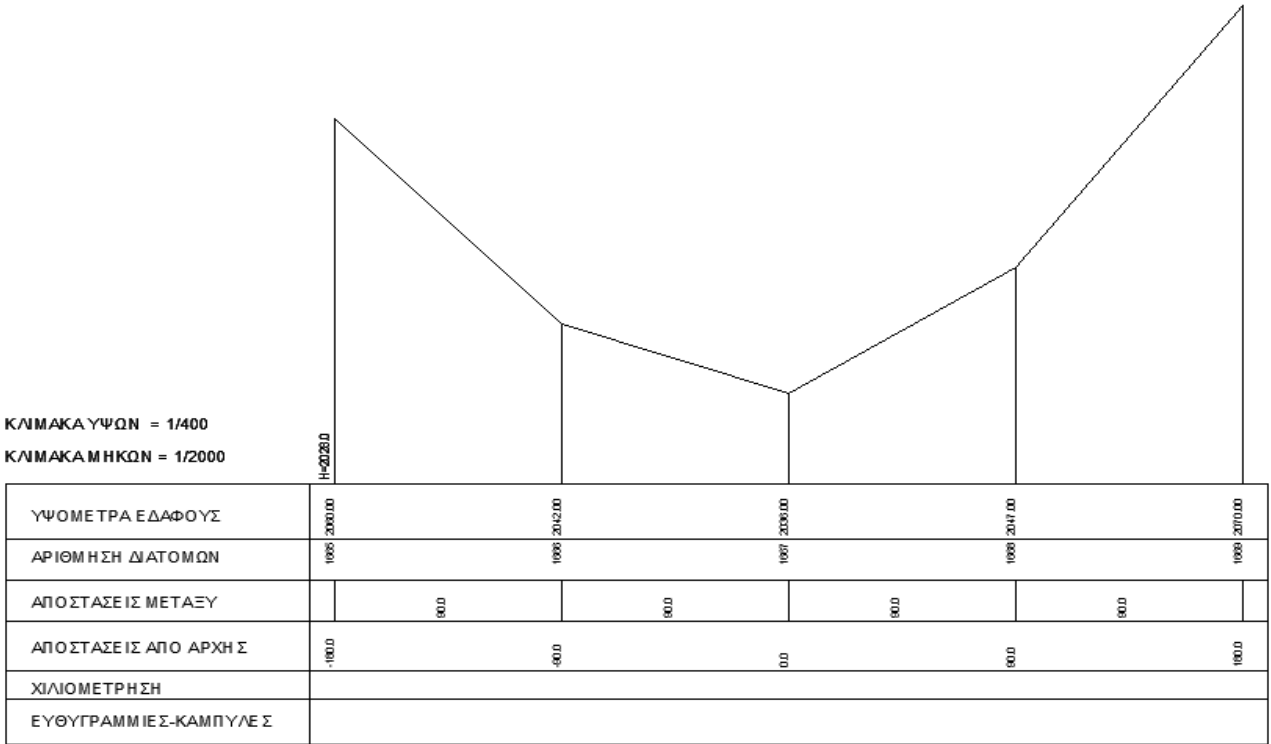


ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/200
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000

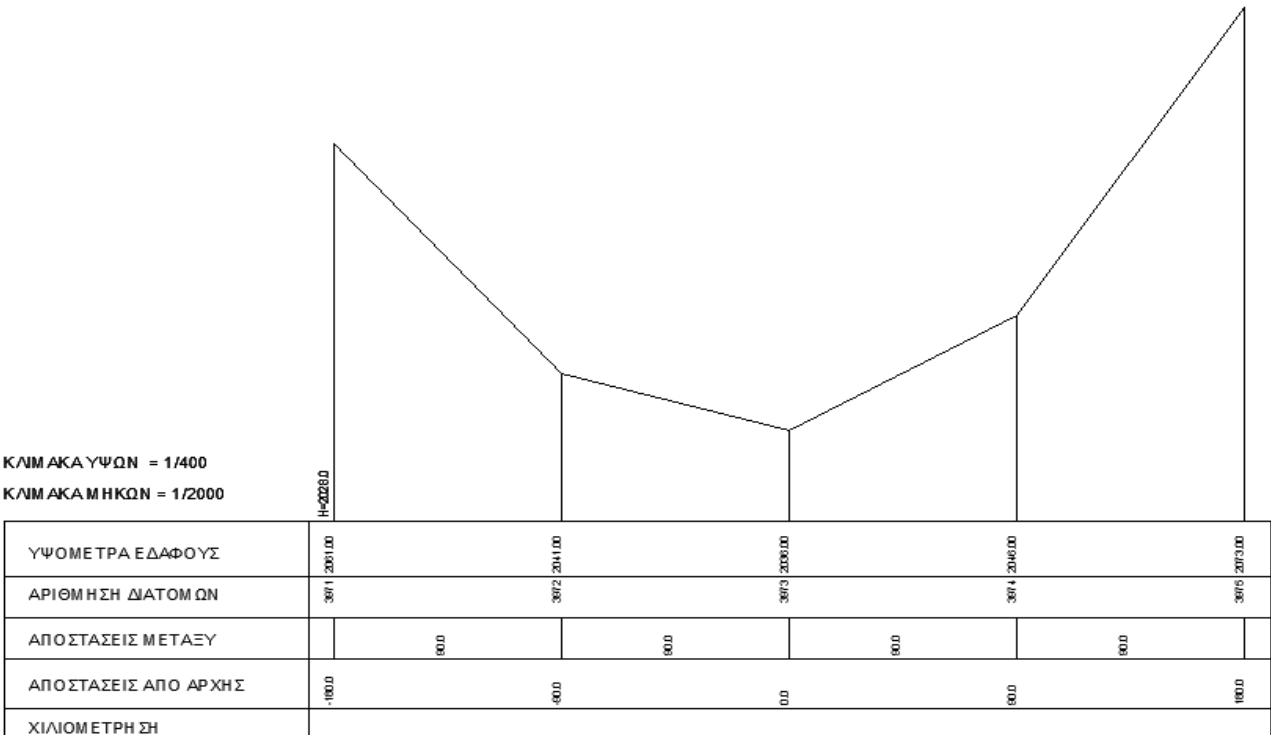


Μηκοτομές σημείου στήλης 1667, γραμμής 3973

ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/400
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000

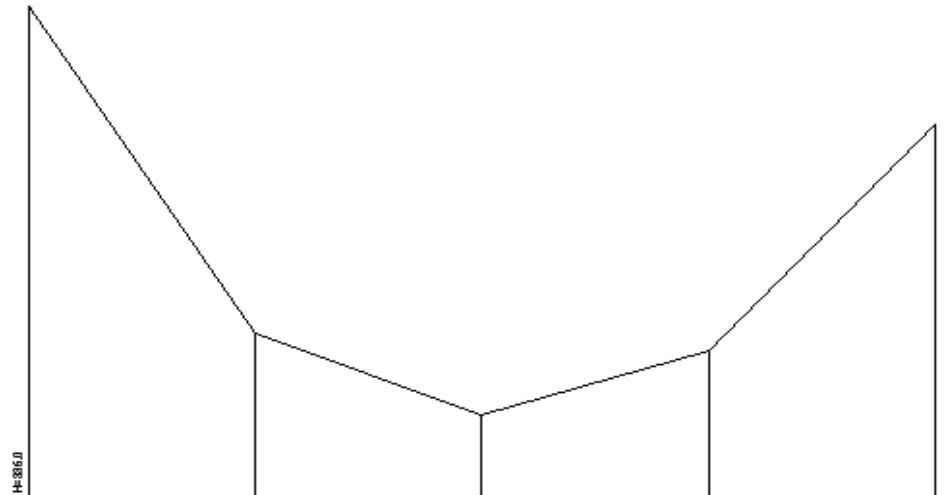


ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/400
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000



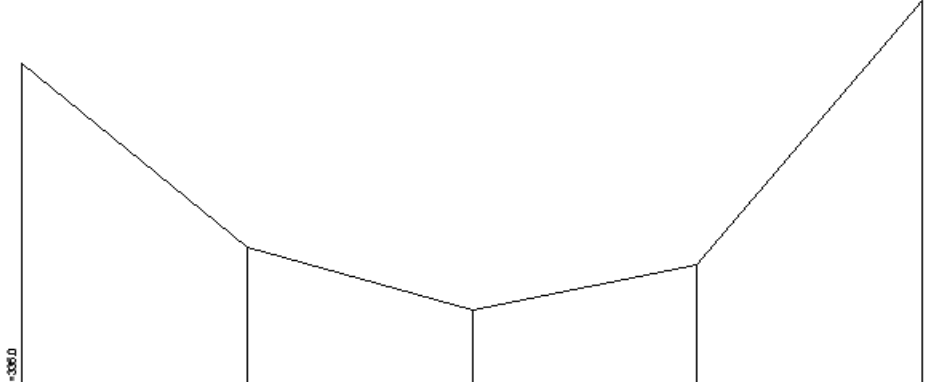
Μηκοτομές σημείου στήλης 1910, γραμμής 14763

ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/500
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000



ΥΨΟΜΕΤΡΑ ΕΔΑΦΟΥΣ	1100	1080	1060	1040	1060	1080	1100
ΑΡΙΘΜΗΣΗ ΔΙΑΤΟΜΩΝ	1000	1000	1000	1000	1000	1000	1000
ΑΠΟΣΤΑΣΕΙΣ ΜΕΤΑΞΥ		900	900	900	900	900	
ΑΠΟΣΤΑΣΕΙΣ ΑΠΟ ΑΡΧΗΣ	0.0	900.0	1800.0	2700.0	3600.0	4500.0	5400.0
ΧΙΛΙΟΜΕΤΡΗΣΗ							
ΕΥΘΥΓΡΑΜΜΙΕΣ-ΚΑΜΠΥΛΕΣ							
ΤΟΞΑ - ΚΛΙΣΕΙΣ							

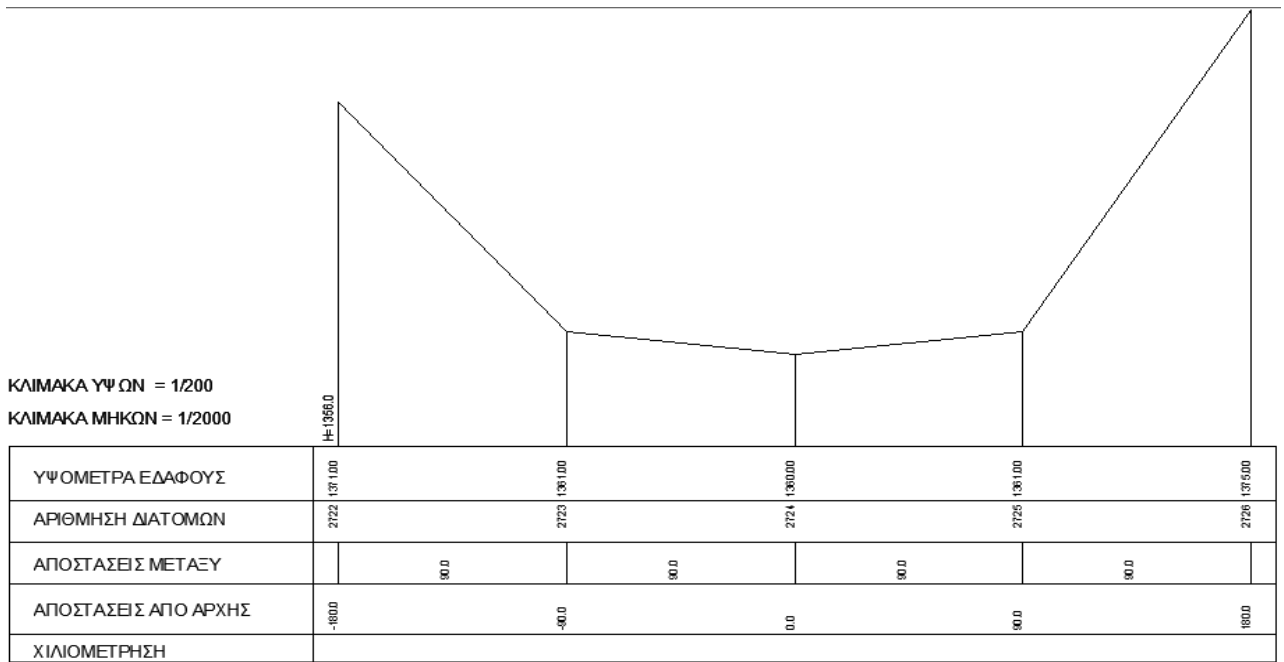
ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/500
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000



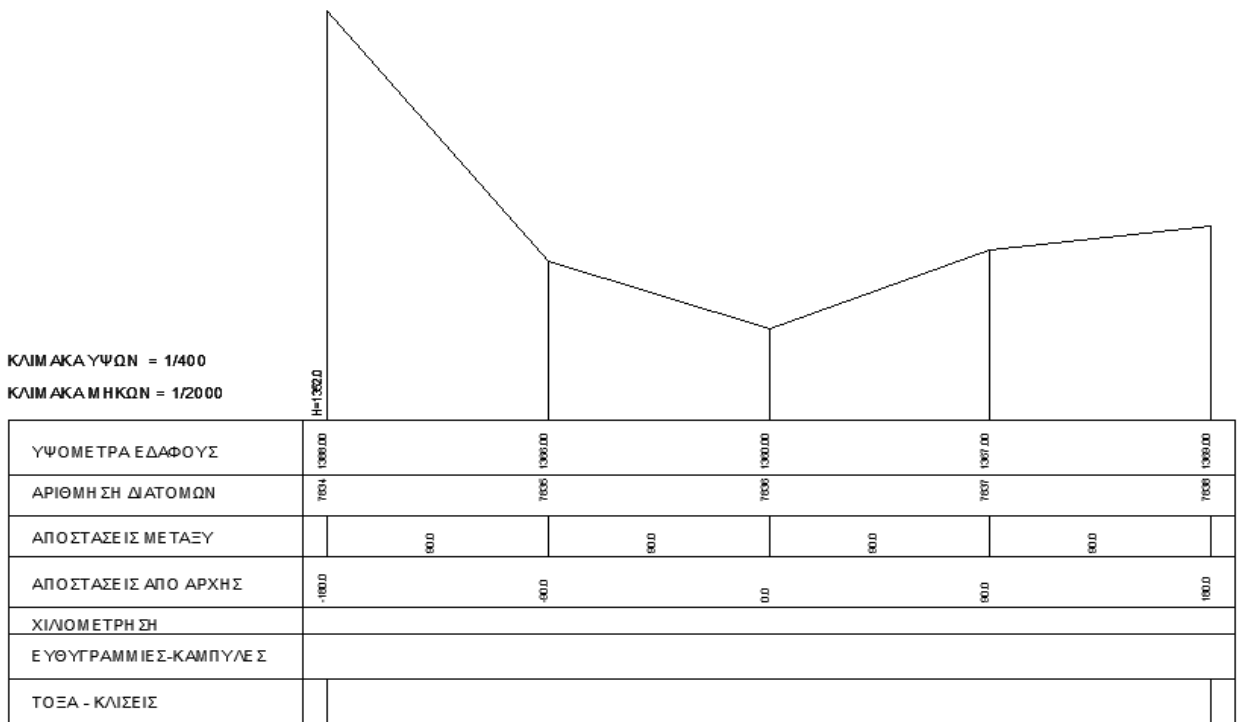
ΥΨΟΜΕΤΡΑ ΕΔΑΦΟΥΣ	1100	1080	1060	1040	1060	1080	1100
ΑΡΙΘΜΗΣΗ ΔΙΑΤΟΜΩΝ	1000	1000	1000	1000	1000	1000	1000
ΑΠΟΣΤΑΣΕΙΣ ΜΕΤΑΞΥ		900	900	900	900	900	
ΑΠΟΣΤΑΣΕΙΣ ΑΠΟ ΑΡΧΗΣ	0.0	900.0	1800.0	2700.0	3600.0	4500.0	5400.0
ΧΙΛΙΟΜΕΤΡΗΣΗ							
ΕΥΘΥΓΡΑΜΜΙΕΣ-ΚΑΜΠΥΛΕΣ							
ΤΟΞΑ - ΚΛΙΣΕΙΣ							

Μηκοτομές σημείου στήλης 2724, γραμμής 7836

ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/200
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000



ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/400
 ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000



ΠΑΡΑΡΤΗΜΑ Α.2.1

ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΔΕΥΤΕΡΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ limnodeksameni2

Στήλη πίνακα Γραμμή πίνακα Περίμετροι που ικανοποιούν

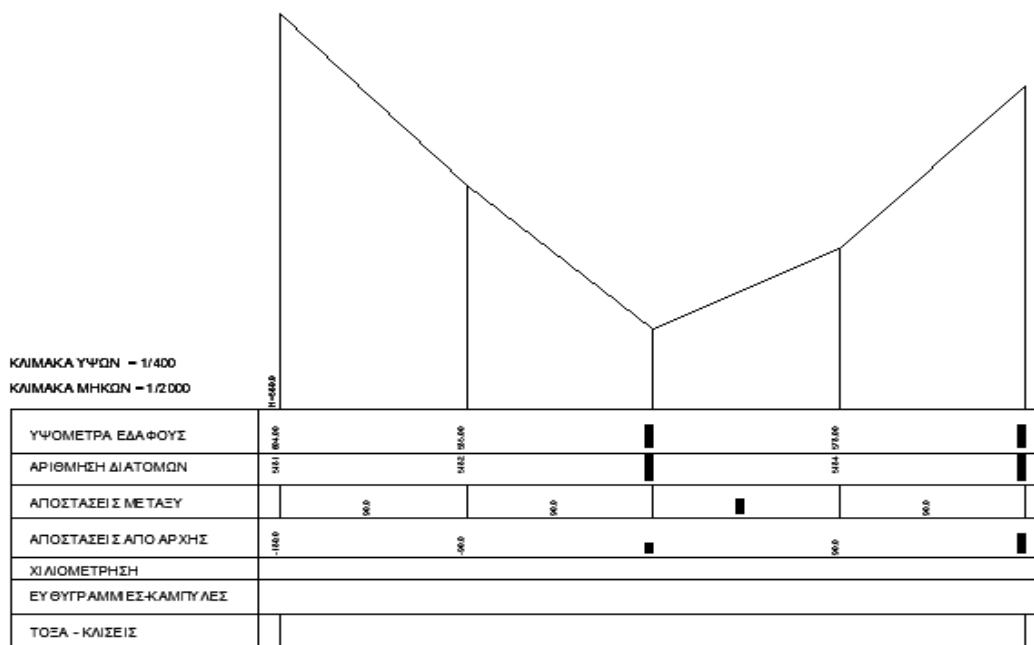
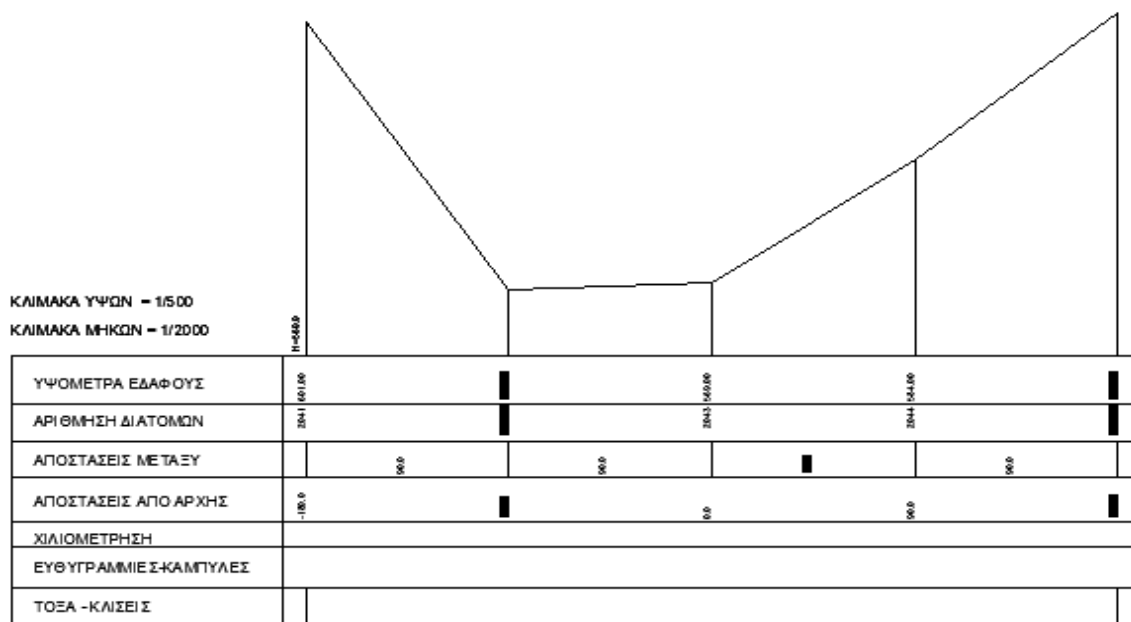
77	2384	2
197	3753	2
236	15716	2
301	16257	2
683	17058	2
687	3321	2
884	6078	2
1001	264	2
1470	17120	2
1602	17540	2
1667	3973	2
1860	16152	2
1910	14763	2
1956	4964	2
2016	17839	2
2043	5183	2
2225	6642	2
2266	8319	2
2553	14645	2
2724	7836	2
2820	3999	2
2935	7298	2
3046	2331	2
3070	16648	2
3340	2310	2
3353	9662	2
3384	17894	2
3444	17672	2
3546	17206	2
3813	90	2
4581	17403	2
4614	1275	2
4679	2298	2
4771	11686	2
4968	11622	2
5142	16109	2
5276	2245	2
5596	11624	2
5753	16032	2
5945	4761	2
6055	987	2
6145	3343	2
6553	16425	2
6665	2946	2
6670	16402	2
6853	858	2
7975	17643	2
8110	609	2
8373	906	2

8381	2773	2
8439	847	2
8446	357	2
8534	1211	2
8561	17372	2
8597	17636	2
8616	1669	3
8672	17311	2
8994	8479	2
9297	7764	2
9316	10601	2
9337	1561	2
9499	9457	2
9769	9423	2
10045	1036	2
10191	1650	2
10267	1384	2
10542	4821	2
10620	17013	2
10896	8212	2
10954	17872	2
11030	17898	2
11126	6502	2
11369	6427	2
11515	10193	2
11540	10133	2
11674	10210	2

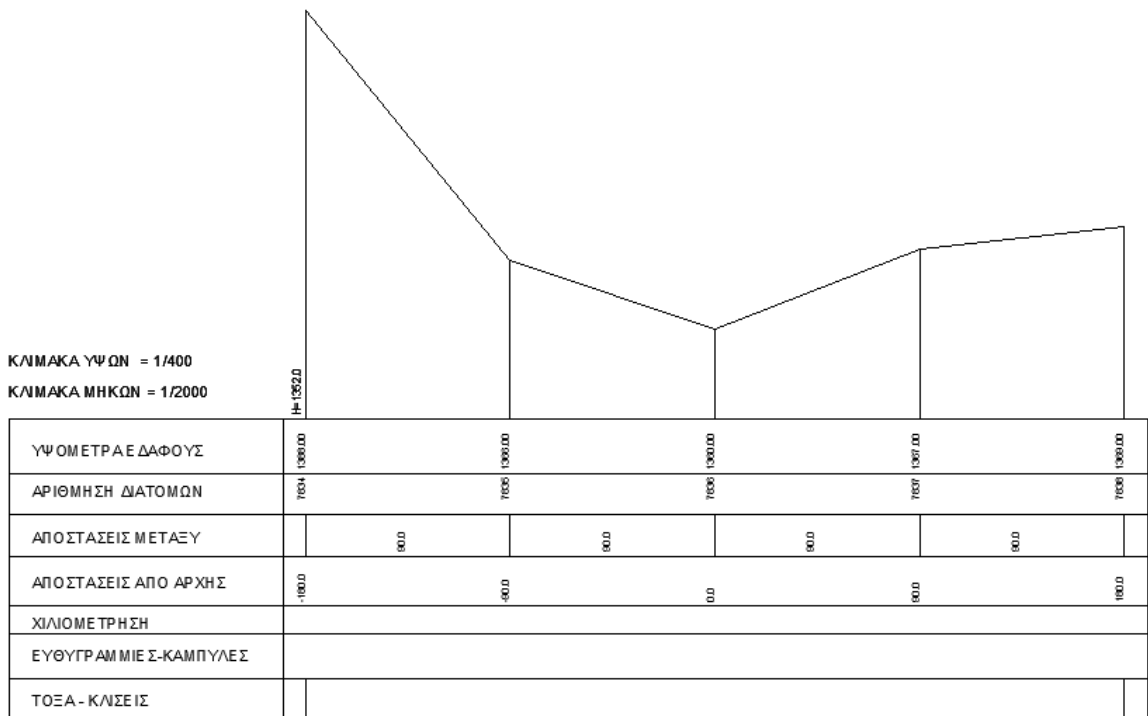
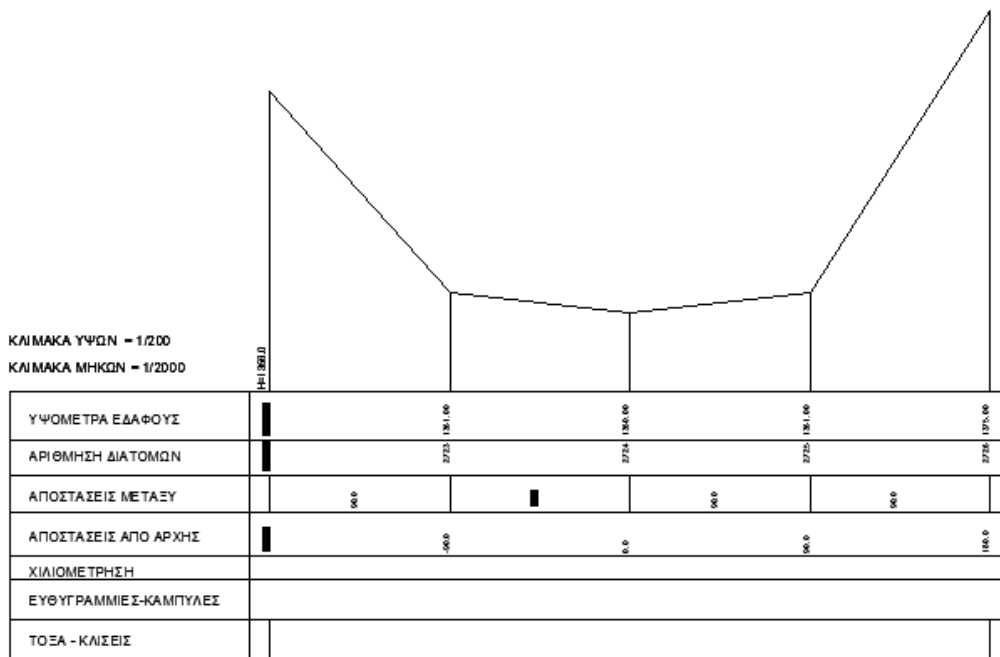
ΠΑΡΑΡΤΗΜΑ Α.2.2

ΜΗΚΟΤΟΜΕΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΔΕΥΤΕΡΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

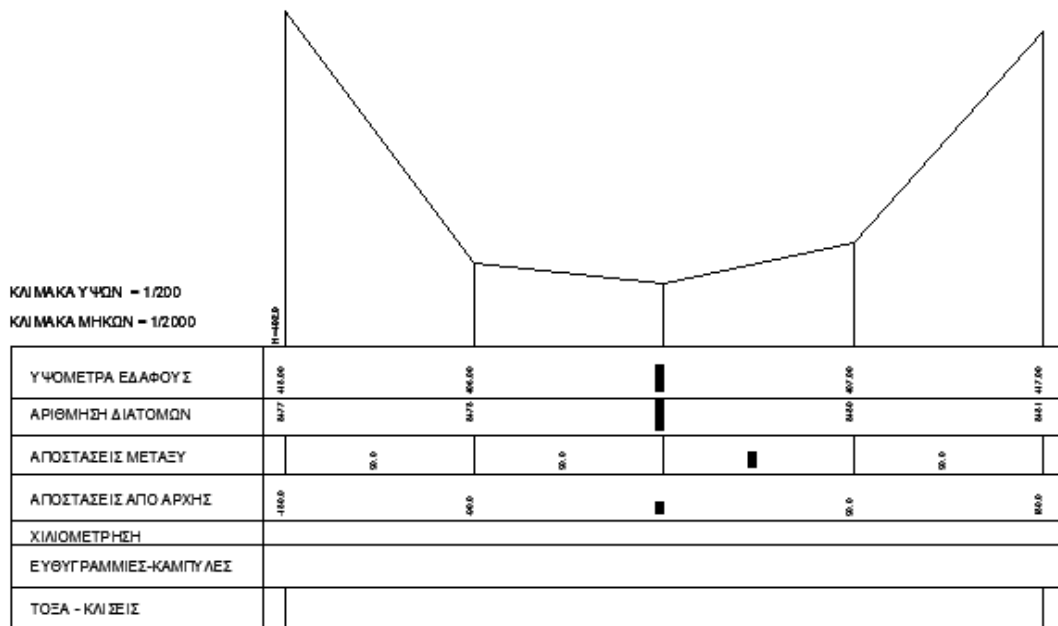
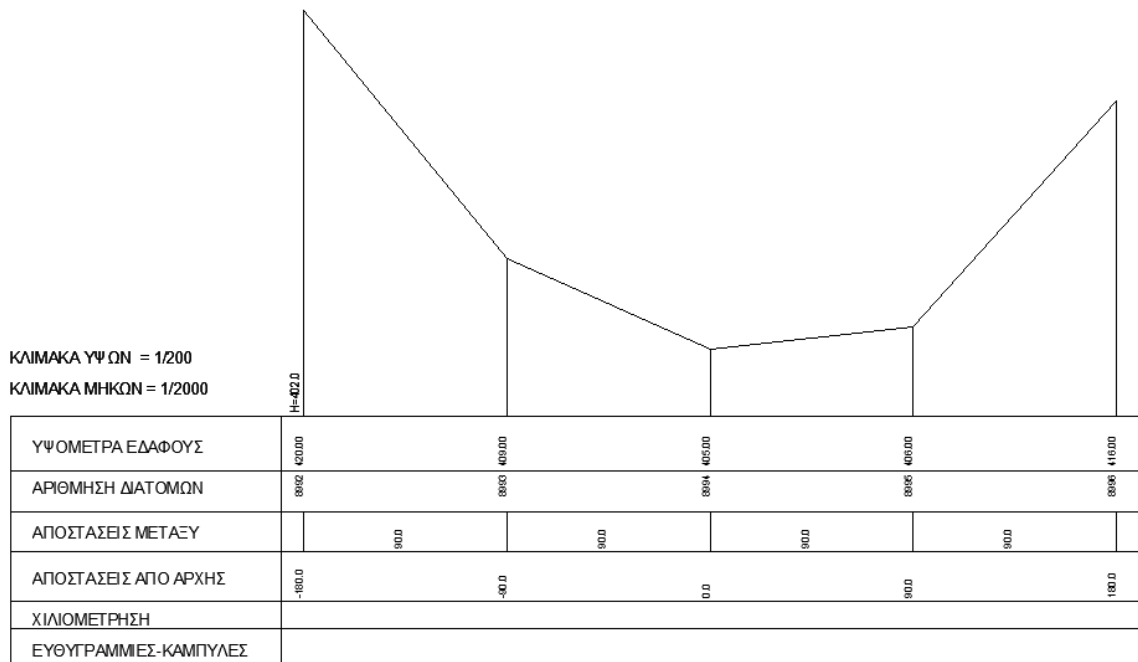
Μηκοτομές σημείου στήλης 2043, γραμμής 5183



Μηκοτομές σημείου στήλης 2724, γραμμής 7836



Μηκοτομές σημείου στήλης 8994, γραμμής 8479



ΠΑΡΑΡΤΗΜΑ Α.3.1

ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΤΡΙΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ limnodeksameni2

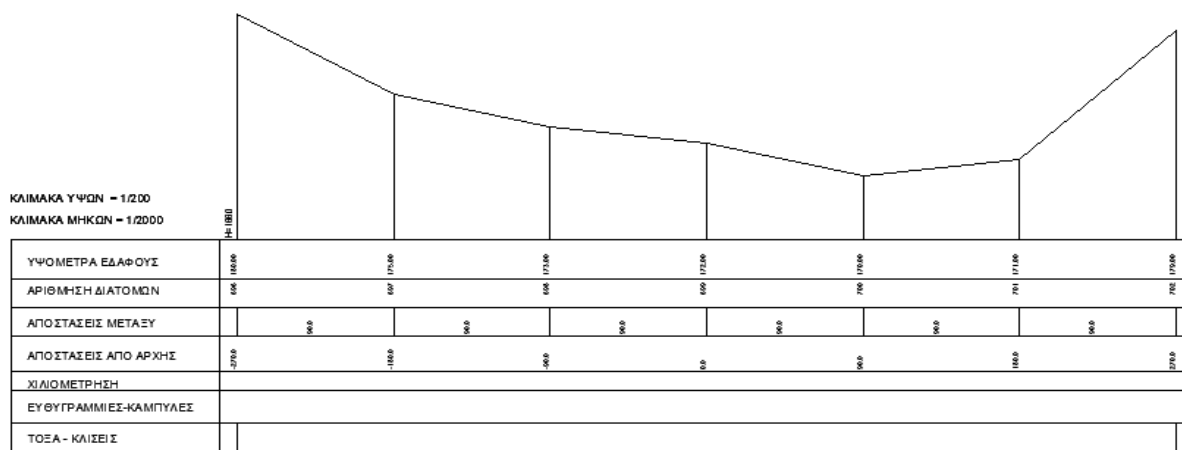
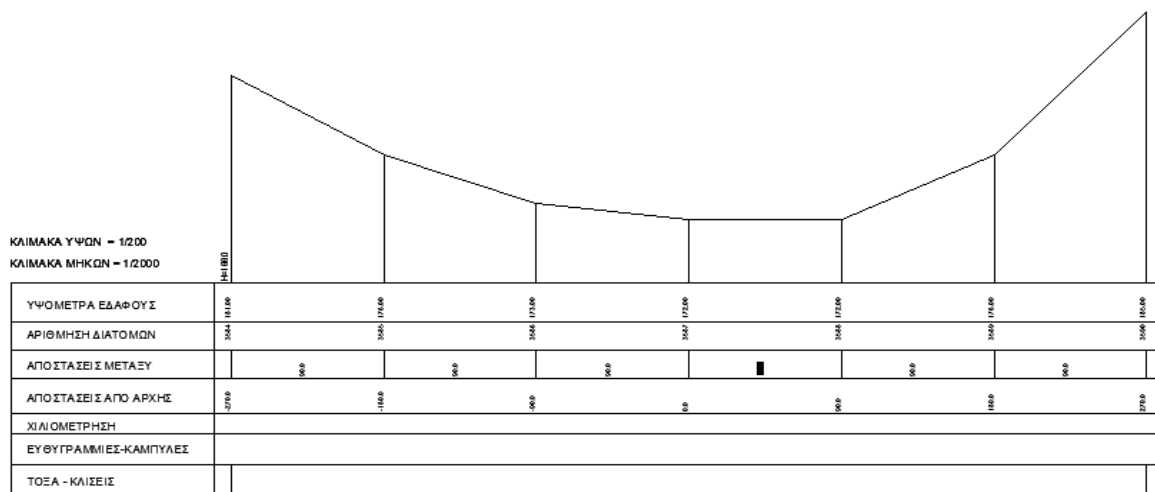
Στήλη πίνακα Γραμμή πίνακα Περίμετροι που ικανοποιούν

3587	699	3
3710	10129	3
3712	2231	3
5907	11656	2
5908	147	2
5909	3022	2
5909	4156	2
5909	17909	2
5910	738	2
5910	3022	2
5914	145	2
5914	1002	2
5914	4589	2
5915	4648	2
5916	1617	2
5916	11856	2
5916	12064	2
5917	1617	2
5918	1617	2
5918	16752	2
5919	1956	2
5921	17334	2
5922	11527	2
5922	11536	2
5923	2052	2
5923	2365	2
5923	4116	2
5924	1599	2
5924	2591	2
5924	4474	2
5924	4617	2
5927	16924	2
5928	156	2
5929	2593	2
5929	2689	2
5929	3016	2
5929	11809	2
5930	17647	2
5938	4547	2
5941	1205	2
5942	3753	2
5942	3997	2
5944	2684	2
5944	2936	2
5945	2659	2
5945	3729	2

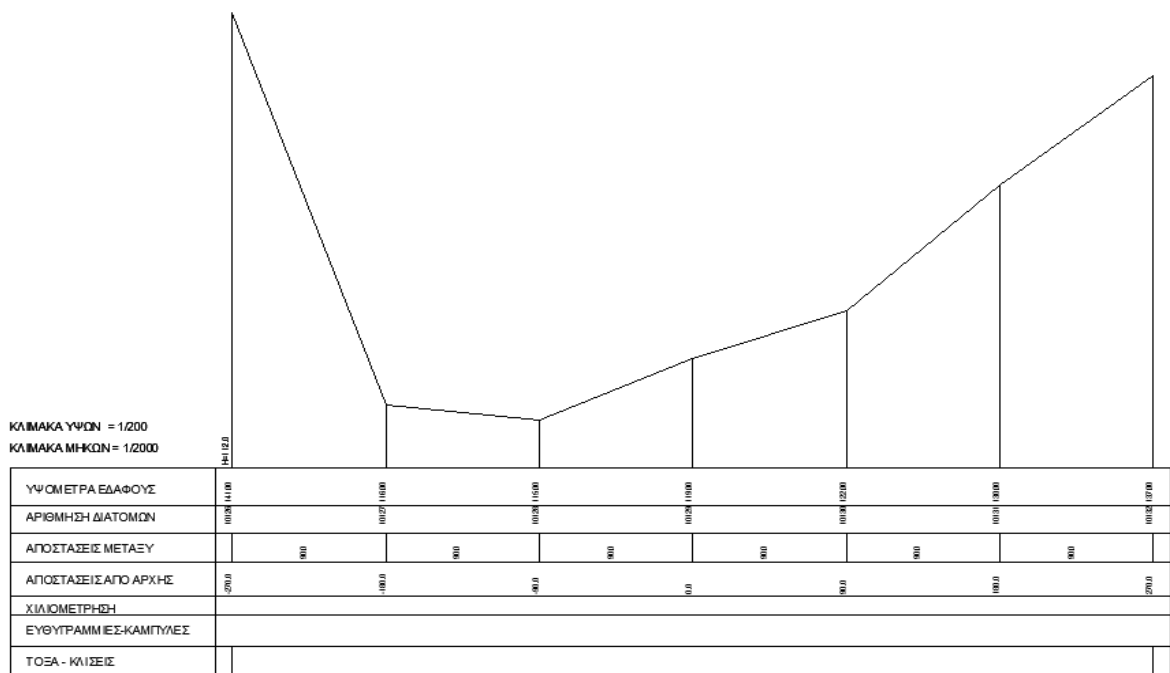
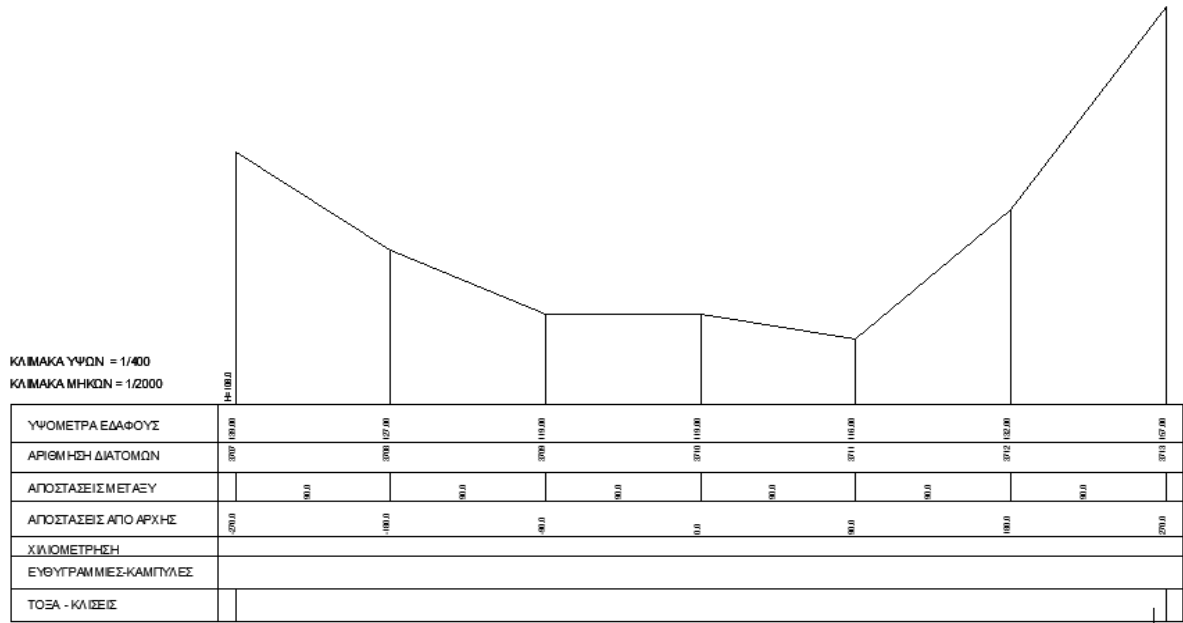
ΠΑΡΑΡΤΗΜΑ Α.3.2

ΜΗΚΟΤΟΜΕΣ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΤΡΙΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

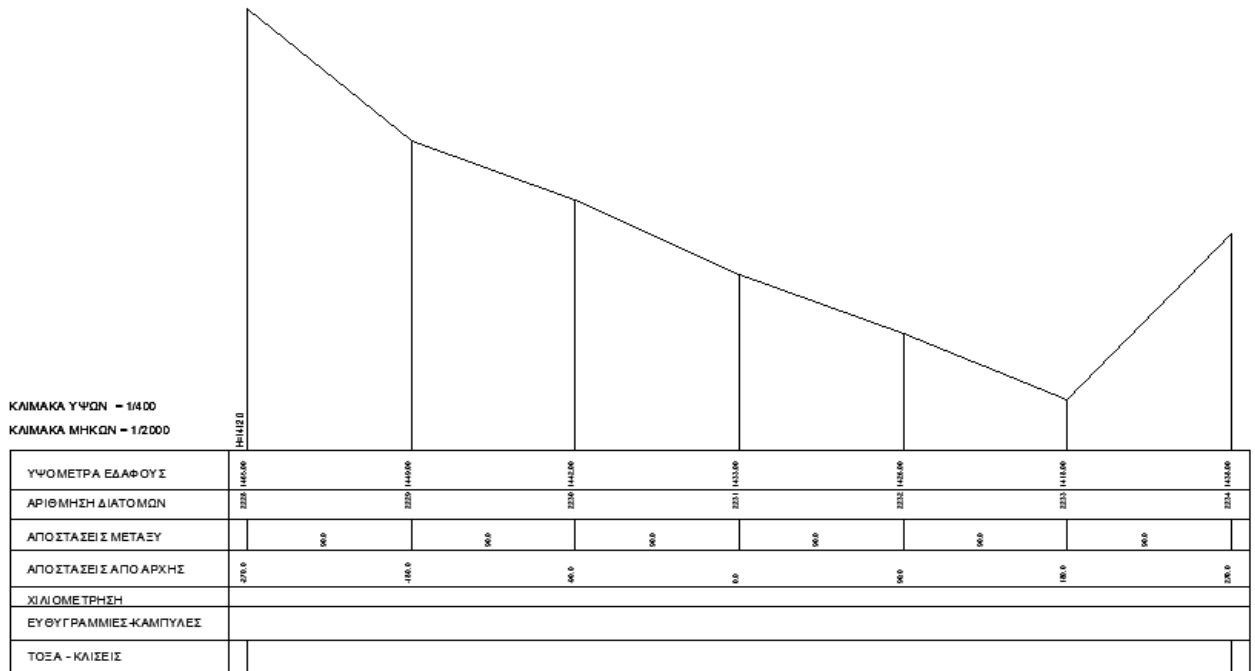
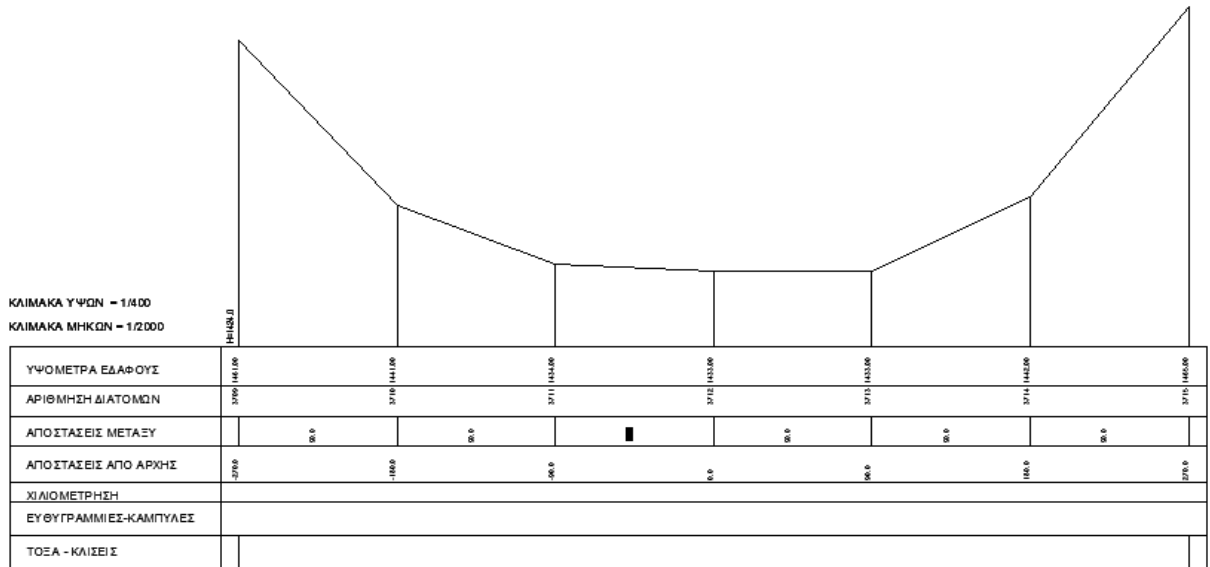
Μηκοτομές σημείου στήλης 3587, γραμμής 699



Μηκοτομές σημείου στήλης 3710, γραμμής 10129



Μηκοτομές σημείου στήλης 3712, γραμμής 2231



ΠΑΡΑΡΤΗΜΑ Α.4.1

ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΤΕΤΑΡΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ ΑΠΟ ΤΟ ΠΡΟΓΡΑΜΜΑ LIMNODEKSAMENI3

Στήλη	Γραμμή	Περ	Ελάχ υψ	Μεγ υψ	Όγκος
62	4351	15	1456.0	1487.0	8.7
149	4348	19	1271.0	1325.0	17.7
150	4347	20	1274.0	1322.0	10.9
150	4348	19	1273.0	1322.0	15.1
150	4349	18	1273.0	1325.0	19.9
150	4350	18	1273.0	1326.0	27.9
151	4347	20	1274.0	1318.0	8.9
151	4348	19	1273.0	1322.0	10.0
151	4349	18	1272.0	1322.0	16.5
151	4350	17	1267.0	1325.0	27.8
152	4348	19	1268.0	1318.0	11.5
152	4349	18	1263.0	1322.0	15.8
152	4350	17	1263.0	1325.0	30.7
152	4351	18	1270.0	1326.0	22.1
153	4348	19	1259.0	1313.0	9.3
153	4349	18	1249.0	1318.0	17.4
153	4350	17	1263.0	1325.0	19.2
153	4351	18	1267.0	1326.0	26.0
154	4348	19	1271.0	1312.0	6.6
154	4349	18	1255.0	1315.0	12.5
154	4350	17	1269.0	1325.0	14.6
154	4351	18	1268.0	1326.0	11.4
225	5772	12	1216.0	1247.0	9.2
225	5774	14	1216.0	1246.0	10.0
225	5775	14	1213.0	1245.0	5.2
225	5776	14	1215.0	1239.0	6.8
226	5772	12	1217.0	1245.0	9.2
226	5775	14	1216.0	1245.0	9.1
226	5776	14	1217.0	1239.0	6.1
328	5463	15	329.0	373.0	5.1
329	5463	15	325.0	373.0	6.5
330	5464	16	326.0	379.0	7.2
331	5464	16	330.0	379.0	9.8
331	5465	17	330.0	381.0	10.1
332	5465	17	328.0	381.0	9.3
332	5466	18	325.0	372.0	6.5
332	5467	17	332.0	371.0	6.3
333	5465	17	333.0	381.0	6.5
333	5467	17	326.0	371.0	7.8
334	5467	17	329.0	371.0	5.8
334	5468	18	325.0	371.0	7.5
335	5468	18	326.0	371.0	6.0
335	5469	13	327.0	367.0	7.1
354	4688	10	1384.0	1429.0	8.2
354	4689	9	1392.0	1422.0	6.4
355	4688	10	1383.0	1429.0	7.4
355	4689	10	1385.0	1440.0	15.6
356	4689	10	1377.0	1440.0	6.4
356	4690	10	1383.0	1449.0	23.3
357	4688	11	1389.0	1440.0	12.2

357	4689	11	1380.0	1449.0	23.7
357	4690	11	1381.0	1453.0	25.5
357	4691	11	1382.0	1449.0	19.4
358	4689	12	1390.0	1453.0	5.4
358	4692	12	1387.0	1449.0	16.8
359	4693	12	1385.0	1440.0	11.8
421	4235	11	1155.0	1187.0	5.6
421	4237	10	1158.0	1192.0	6.4
547	6777	11	239.0	294.0	7.0
547	6778	10	237.0	294.0	7.6
547	6779	9	243.0	294.0	5.8
547	6780	8	257.0	294.0	5.5
548	6777	11	236.0	294.0	8.8
548	6778	10	231.0	294.0	9.7
548	6779	9	236.0	294.0	7.9
549	6712	19	69.0	108.0	12.5
549	6777	11	237.0	294.0	10.7
549	6778	10	236.0	294.0	10.5
549	6779	9	235.0	294.0	6.9
549	6780	8	236.0	294.0	9.7
549	6781	9	236.0	287.0	6.7
550	6712	18	67.0	108.0	6.8
550	6777	11	239.0	294.0	9.7
550	6778	10	238.0	294.0	7.3
550	6779	9	237.0	294.0	8.9
550	6780	8	238.0	294.0	9.2
550	6781	9	239.0	287.0	5.4
551	6713	17	71.0	108.0	8.0
551	6714	17	68.0	108.0	12.0
551	6715	17	69.0	108.0	14.4
551	6716	17	71.0	108.0	11.5
551	6777	11	243.0	294.0	5.5
551	6778	10	242.0	294.0	7.0
551	6779	9	240.0	294.0	8.8
551	6780	8	240.0	294.0	8.8
552	6713	16	71.0	108.0	7.4
552	6714	16	68.0	108.0	11.8
552	6716	16	67.0	108.0	13.9
552	6779	9	250.0	294.0	5.2
553	6716	15	68.0	108.0	12.7
553	6717	15	71.0	108.0	9.5
599	4936	15	957.0	995.0	5.7
739	6098	14	463.0	481.0	6.3
824	6192	18	569.0	589.0	5.1
824	6193	18	570.0	591.0	5.7
825	6194	18	570.0	592.0	6.7
826	6194	18	570.0	593.0	5.9
898	6102	20	579.0	662.0	8.3
899	6100	20	567.0	651.0	5.1
899	6101	20	566.0	661.0	12.1
899	6102	20	573.0	662.0	9.8
899	6103	20	575.0	667.0	9.5
900	6104	20	576.0	668.0	10.4
920	6449	19	612.0	632.0	9.7
921	6449	20	612.0	632.0	6.4
922	6449	20	610.0	630.0	5.6
923	6449	20	611.0	628.0	7.0
954	6554	17	620.0	637.0	6.3

957	6553	18	620.0	637.0	5.8
957	6554	17	619.0	637.0	8.0
958	6553	18	619.0	637.0	7.4
958	6554	17	618.0	637.0	6.2
958	6555	16	619.0	637.0	7.7
959	6552	19	620.0	637.0	7.0
959	6554	17	618.0	637.0	7.5
1045	4789	16	1427.0	1464.0	16.6
1045	4790	17	1424.0	1464.0	16.6
1045	4792	16	1428.0	1463.0	9.9
1045	4793	17	1429.0	1463.0	15.2
1046	4790	17	1425.0	1464.0	17.3
1046	4792	16	1427.0	1463.0	14.7
1046	4793	17	1423.0	1463.0	14.0
1047	4790	17	1427.0	1464.0	16.3
1047	4792	16	1425.0	1463.0	16.1
1047	4793	17	1424.0	1463.0	13.1
1048	4791	15	1425.0	1463.0	11.0
1048	4792	16	1426.0	1463.0	9.1
1048	4793	15	1424.0	1460.0	14.3
1048	4796	16	1427.0	1454.0	5.8
1049	4792	14	1428.0	1460.0	7.1
1154	5102	19	1073.0	1095.0	5.1
1155	5103	20	1077.0	1097.0	5.4
1155	5104	20	1075.0	1100.0	7.4
1156	5103	19	1076.0	1099.0	6.3
1156	5104	20	1074.0	1101.0	20.1
1157	5103	19	1077.0	1098.0	12.5
1157	5105	19	1077.0	1098.0	12.4
1158	5104	20	1073.0	1098.0	11.9
1159	5104	20	1074.0	1096.0	13.4
1370	7506	9	187.0	240.0	5.0
1371	7504	7	164.0	239.0	9.3
1371	7505	8	152.0	240.0	10.8
1371	7506	8	147.0	240.0	11.3
1371	7507	8	160.0	240.0	10.1
1372	7504	7	149.0	240.0	13.4
1372	7505	7	137.0	240.0	14.5
1372	7506	7	133.0	240.0	15.0
1372	7507	7	142.0	240.0	13.4
1373	7504	6	149.0	240.0	13.3
1373	7505	6	140.0	240.0	14.8
1373	7506	6	136.0	240.0	14.4
1373	7507	6	139.0	235.0	12.1
1374	7504	6	159.0	231.0	7.8
1374	7505	6	156.0	231.0	8.6
1374	7506	6	156.0	231.0	8.6
1374	7507	6	164.0	231.0	6.3
1556	7005	12	1090.0	1142.0	6.9
1556	7006	11	1089.0	1142.0	8.0
1556	7007	11	1088.0	1140.0	7.9
1557	7005	10	1090.0	1142.0	8.3
1557	7006	11	1088.0	1142.0	8.3
1557	7007	12	1091.0	1142.0	7.2
1559	7026	7	1154.0	1213.0	5.7
1559	7027	8	1159.0	1214.0	5.3
1560	7026	7	1154.0	1214.0	6.4
1561	7026	7	1150.0	1214.0	6.7

1602	7094	20	891.0	923.0	5.5
1603	7094	20	891.0	923.0	6.8
1603	7095	20	890.0	933.0	12.5
1603	7096	19	892.0	933.0	10.3
1604	7093	20	889.0	916.0	6.0
1605	7093	20	891.0	916.0	6.3
1605	7094	20	888.0	923.0	10.0
1605	7095	20	890.0	933.0	11.5
1605	7096	19	890.0	933.0	11.0
1606	7094	20	891.0	923.0	7.1
1606	7095	20	889.0	933.0	14.5
1606	7096	19	889.0	933.0	13.4
1607	7094	20	888.0	923.0	6.9
1619	7129	9	710.0	762.0	6.3
1619	7130	8	710.0	762.0	6.6
1620	7129	9	713.0	762.0	6.2
1620	7130	8	710.0	762.0	6.6
1620	7131	7	713.0	762.0	5.4
1621	7129	9	717.0	762.0	5.8
1621	7130	8	712.0	762.0	5.9
1621	7131	7	715.0	762.0	5.6
1622	7130	8	711.0	762.0	5.9
1622	7131	7	713.0	762.0	5.3
1623	7132	7	709.0	753.0	5.4
1720	7084	17	269.0	303.0	10.5
1819	4726	20	535.0	594.0	42.0
1819	4727	20	535.0	594.0	44.6
1820	4725	20	541.0	601.0	8.1
1820	4726	20	537.0	601.0	14.5
1898	5096	11	619.0	666.0	5.9
1899	5097	10	622.0	669.0	6.0
1900	5098	9	633.0	672.0	5.9
1984	5244	20	501.0	544.0	19.9
1985	5242	19	500.0	531.0	11.5
1985	5243	20	498.0	534.0	7.1
1985	5244	20	496.0	544.0	18.9
1985	5245	20	496.0	550.0	61.3
1985	5246	20	497.0	550.0	22.5
1986	5242	19	497.0	531.0	12.0
1986	5243	20	496.0	534.0	17.0
1986	5244	20	497.0	544.0	18.2
1986	5245	20	493.0	545.0	41.6
1986	5246	20	493.0	545.0	28.0
1986	5247	20	496.0	545.0	30.8
1987	5242	19	497.0	531.0	13.7
1987	5243	20	502.0	534.0	13.4
1987	5245	20	497.0	544.0	34.5
1987	5246	20	496.0	544.0	39.3
2083	5394	9	565.0	598.0	7.0
2083	5398	9	570.0	600.0	6.9
2084	5394	9	566.0	601.0	5.9
2084	5398	9	567.0	601.0	5.0
2084	5399	9	573.0	601.0	6.2
2085	5393	10	571.0	599.0	5.8
2085	5394	9	569.0	601.0	7.8
2085	5396	10	576.0	603.0	6.5
2085	5397	10	569.0	603.0	6.1
2085	5398	10	567.0	603.0	6.6

2086	5394	10	572.0	602.0	5.3
2086	5397	10	573.0	606.0	10.5
2086	5398	10	567.0	606.0	7.6
2149	5469	19	523.0	560.0	9.3
2150	5468	19	522.0	567.0	16.0
2150	5469	19	521.0	564.0	13.2
2150	5470	18	521.0	560.0	10.4
2151	5468	18	521.0	567.0	11.7
2151	5469	18	517.0	567.0	15.8
2151	5470	18	521.0	564.0	13.4
2152	5469	17	520.0	567.0	12.4
2152	5470	17	515.0	567.0	15.1
2153	5468	16	524.0	567.0	13.4
2153	5469	16	524.0	567.0	6.9
2153	5470	16	518.0	567.0	16.5
2153	5471	16	520.0	567.0	17.4
2154	5471	15	524.0	567.0	6.8
2155	5469	15	522.0	568.0	12.1
2164	7557	12	618.0	714.0	12.6
2164	7558	10	620.0	705.0	12.0
2164	7559	11	613.0	697.0	8.5
2165	7557	12	619.0	714.0	12.4
2165	7558	10	629.0	705.0	6.1
2165	7560	9	607.0	689.0	7.2
2165	7561	9	601.0	683.0	6.7
2166	7561	9	613.0	683.0	6.2
2172	5492	16	488.0	543.0	26.1
2172	5493	17	489.0	550.0	33.7
2173	5496	19	489.0	564.0	53.8
2174	5495	18	491.0	564.0	31.4
2174	5496	18	489.0	567.0	62.6
2175	5494	17	494.0	564.0	33.8
2175	5495	17	487.0	567.0	50.6
2175	5496	17	491.0	567.0	48.8
2176	5494	16	492.0	567.0	41.4
2176	5495	16	489.0	567.0	38.3
2215	5895	9	506.0	575.0	9.4
2215	5896	9	494.0	576.0	11.8
2215	5897	9	490.0	576.0	14.1
2215	5898	9	496.0	576.0	13.5
2216	5895	8	499.0	576.0	9.8
2216	5896	8	484.0	576.0	14.7
2216	5897	8	482.0	576.0	14.3
2216	5898	8	488.0	576.0	14.3
2216	5899	8	507.0	576.0	5.6
2216	5900	8	521.0	576.0	8.6
2217	5896	7	486.0	576.0	13.1
2217	5897	7	481.0	576.0	14.8
2217	5898	7	483.0	576.0	10.0
2217	5899	7	499.0	576.0	12.8
2217	5900	7	513.0	576.0	10.6
2218	5897	6	481.0	576.0	12.6
2218	5898	6	480.0	576.0	17.0
2218	5899	6	491.0	576.0	14.9
2218	5900	7	510.0	577.0	10.9
2219	5898	6	481.0	554.0	8.2
2219	5899	6	488.0	554.0	6.8
2421	5600	12	888.0	956.0	29.5

2421	5601	13	886.0	964.0	35.7
2421	5602	13	887.0	964.0	36.5
2422	5598	11	885.0	946.0	26.8
2422	5599	12	886.0	949.0	16.0
2422	5600	13	885.0	954.0	32.5
2422	5601	13	885.0	956.0	26.4
2422	5602	13	886.0	956.0	28.9
2422	5603	12	887.0	949.0	19.7
2423	5598	12	885.0	944.0	17.6
2423	5599	12	885.0	946.0	25.5
2423	5600	13	885.0	949.0	26.7
2423	5601	13	886.0	949.0	20.4
2423	5602	12	886.0	946.0	16.5
2423	5603	12	887.0	946.0	18.5
2424	5599	12	883.0	942.0	20.0
2424	5600	12	884.0	942.0	12.9
2424	5601	12	884.0	942.0	16.1
2424	5602	12	885.0	942.0	15.2
2424	5603	12	887.0	938.0	14.6
2425	5599	12	886.0	935.0	10.7
2425	5602	11	886.0	928.0	6.2
2426	5600	11	888.0	920.0	5.3
2495	9066	12	302.0	336.0	9.2
2495	9067	13	296.0	336.0	10.2
2495	9068	14	298.0	336.0	9.8
2495	9069	15	302.0	336.0	7.9
2497	9065	11	297.0	336.0	12.1
2497	9066	12	293.0	336.0	12.4
2497	9067	13	292.0	336.0	11.9
2497	9068	14	293.0	336.0	12.5
2497	9069	15	292.0	336.0	15.0
2497	9070	16	298.0	336.0	13.6
2498	9065	11	299.0	336.0	12.9
2498	9066	12	296.0	336.0	12.5
2498	9067	13	292.0	336.0	11.7
2498	9068	14	292.0	336.0	15.1
2498	9069	15	294.0	336.0	14.2
2498	9070	16	299.0	336.0	12.9
2499	9066	12	301.0	336.0	8.1
2499	9067	13	296.0	336.0	13.0
2499	9068	14	294.0	336.0	13.6
2499	9069	15	297.0	336.0	12.2
2499	9070	16	302.0	336.0	10.8
2525	5482	7	1439.0	1484.0	5.1
2525	5483	7	1435.0	1484.0	5.1
2526	5481	6	1423.0	1484.0	7.1
2526	5482	6	1416.0	1484.0	7.9
2526	5483	6	1424.0	1484.0	6.6
2527	5481	6	1419.0	1480.0	5.7
2527	5482	6	1404.0	1480.0	8.2
2527	5483	6	1421.0	1480.0	5.9
2528	9135	20	280.0	322.0	32.5
2528	9136	16	277.0	320.0	31.8
2529	9135	20	278.0	322.0	31.6
2529	9136	15	277.0	322.0	30.6
2530	9135	20	276.0	322.0	23.1
2530	9136	16	276.0	320.0	23.5
2531	9134	20	281.0	323.0	30.2

2531	9135	20	274.0	322.0	19.1
2531	9136	20	276.0	320.0	27.9
2532	9135	20	273.0	322.0	31.6
2532	9136	20	276.0	320.0	5.0
2576	8917	11	1264.0	1311.0	10.0
2576	8918	11	1259.0	1311.0	11.0
2576	8919	11	1260.0	1311.0	12.2
2576	8920	10	1258.0	1308.0	10.1
2577	8918	10	1259.0	1310.0	10.8
2577	8919	10	1258.0	1310.0	12.2
2577	8920	10	1254.0	1310.0	12.2
2578	8920	10	1263.0	1308.0	10.0
2579	8919	11	1266.0	1308.0	8.8
2601	8253	17	486.0	581.0	13.1
2601	8254	18	479.0	565.0	9.7
2716	7726	10	1503.0	1539.0	8.0
2717	7726	10	1499.0	1536.0	8.5
2717	7728	9	1504.0	1531.0	5.1
2718	7726	10	1502.0	1531.0	5.5
2718	7727	10	1499.0	1531.0	5.8
2719	7726	11	1501.0	1531.0	6.5
2719	7729	10	1500.0	1527.0	5.4
2720	7727	11	1498.0	1527.0	5.4
2720	7730	11	1502.0	1527.0	5.2
2721	7728	12	1493.0	1527.0	5.7
2738	7709	8	1546.0	1587.0	6.5
2750	7771	20	1086.0	1115.0	9.5
2751	7769	15	1088.0	1115.0	11.0
2751	7770	15	1086.0	1115.0	11.3
2751	7771	15	1087.0	1115.0	8.1
2751	7773	15	1088.0	1115.0	7.7
2751	7774	15	1087.0	1115.0	8.3
2752	7769	16	1088.0	1126.0	19.7
2752	7770	16	1085.0	1126.0	20.8
2752	7771	16	1086.0	1126.0	17.5
2753	7771	17	1088.0	1140.0	31.3
2753	7774	17	1088.0	1140.0	28.7
2754	7770	17	1087.0	1145.0	38.7
2755	7770	18	1087.0	1145.0	12.5
2755	7771	18	1086.0	1145.0	17.0
2755	7774	18	1087.0	1145.0	24.3
3026	7888	20	540.0	562.0	9.0
3026	7889	20	540.0	560.0	5.6
3027	7889	19	540.0	560.0	6.6
3411	7954	15	797.0	841.0	17.4
3412	7953	14	796.0	829.0	7.9
3412	7954	15	796.0	841.0	18.3
3412	7955	15	798.0	850.0	26.8
3413	7954	15	796.0	841.0	21.2
3413	7955	15	797.0	853.0	26.3
3414	7952	14	793.0	823.0	7.0
3414	7953	14	794.0	829.0	11.8
3414	7954	15	798.0	841.0	10.9
3414	7955	15	798.0	852.0	22.0
3415	7952	14	791.0	823.0	6.1
3415	7953	14	796.0	829.0	11.2
3415	7954	15	798.0	841.0	17.0
3415	7956	15	797.0	856.0	25.3

3416	7954	15	800.0	841.0	16.2
3491	9788	14	458.0	488.0	5.1
3491	9789	14	451.0	487.0	6.0
3491	9791	11	462.0	487.0	5.2
3492	9789	15	452.0	488.0	7.2
3492	9790	10	453.0	487.0	6.6
3493	9788	16	461.0	490.0	5.5
3493	9789	16	454.0	489.0	6.1
3494	9787	17	467.0	494.0	5.5
3602	9732	17	409.0	438.0	11.0
3603	9732	17	410.0	436.0	8.1
3603	9734	15	411.0	432.0	5.2
3603	9735	15	412.0	431.0	5.6
3603	9736	13	410.0	430.0	6.1
3604	9735	14	408.0	430.0	7.1
3604	9736	14	405.0	430.0	8.0
3605	9733	15	409.0	430.0	5.1
3605	9734	15	409.0	430.0	5.8
3605	9736	14	407.0	428.0	5.9
3833	7052	13	14.0	44.0	6.0
3834	7054	12	13.0	47.0	5.2
3834	7055	12	15.0	47.0	6.3
3835	7052	11	15.0	47.0	6.5
3835	7053	11	17.0	47.0	5.4
3835	7054	11	13.0	47.0	5.4
3835	7055	11	13.0	47.0	6.6
3836	7053	10	15.0	47.0	7.8
3836	7054	10	14.0	47.0	6.0
4102	7523	8	122.0	190.0	9.8
4103	7522	7	125.0	183.0	5.4
4103	7523	8	129.0	184.0	5.2
4752	11659	19	321.0	466.0	15.8
4753	11658	15	328.0	484.0	22.3
4753	11659	19	315.0	466.0	17.0
4753	11660	18	311.0	466.0	16.7
4753	11661	18	305.0	445.0	11.8
4753	11662	18	296.0	419.0	7.8
4754	11660	18	331.0	466.0	14.1
4754	11661	18	311.0	445.0	12.5
4754	11663	19	299.0	411.0	6.4
4755	11661	18	323.0	445.0	8.6
4755	11662	18	310.0	419.0	5.5
4819	11583	7	1668.0	1734.0	6.8
4820	11583	7	1676.0	1744.0	11.0
4820	11584	7	1686.0	1744.0	5.1
4820	11585	7	1697.0	1744.0	5.4
4851	11627	5	1965.0	2037.0	5.8
4851	11628	5	1953.0	2035.0	7.1
4851	11629	5	1957.0	2037.0	6.8
4852	11597	11	1762.0	1818.0	9.3
4852	11598	11	1757.0	1824.0	13.7
4852	11599	11	1753.0	1825.0	12.9
4852	11600	11	1761.0	1825.0	12.6
4852	11601	11	1774.0	1825.0	5.8
4852	11627	6	1966.0	2035.0	5.8
4852	11628	6	1954.0	2037.0	6.6
4852	11629	6	1964.0	2037.0	6.1
4853	11597	12	1764.0	1824.0	9.7

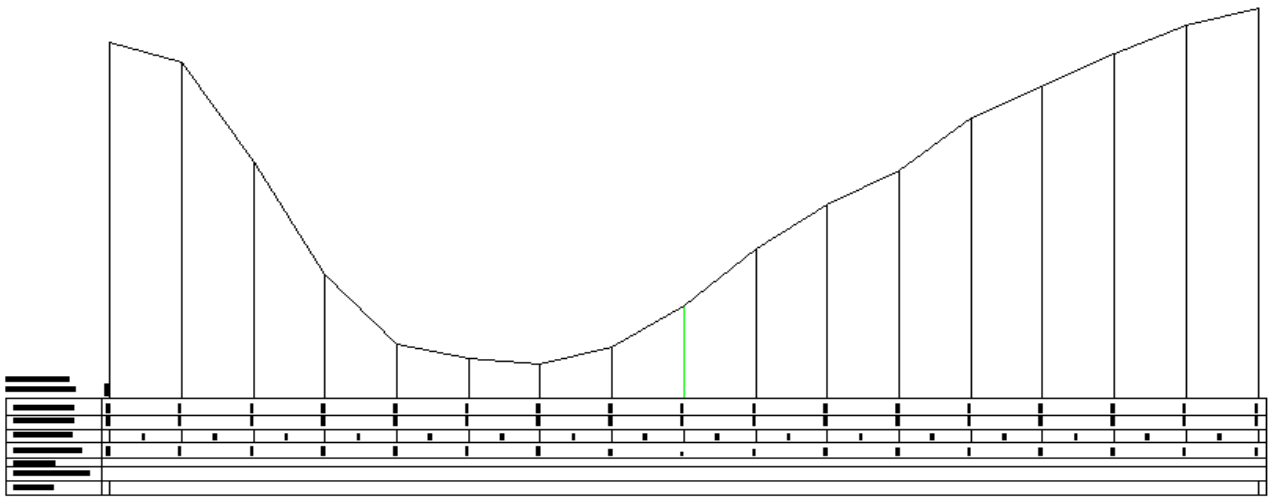
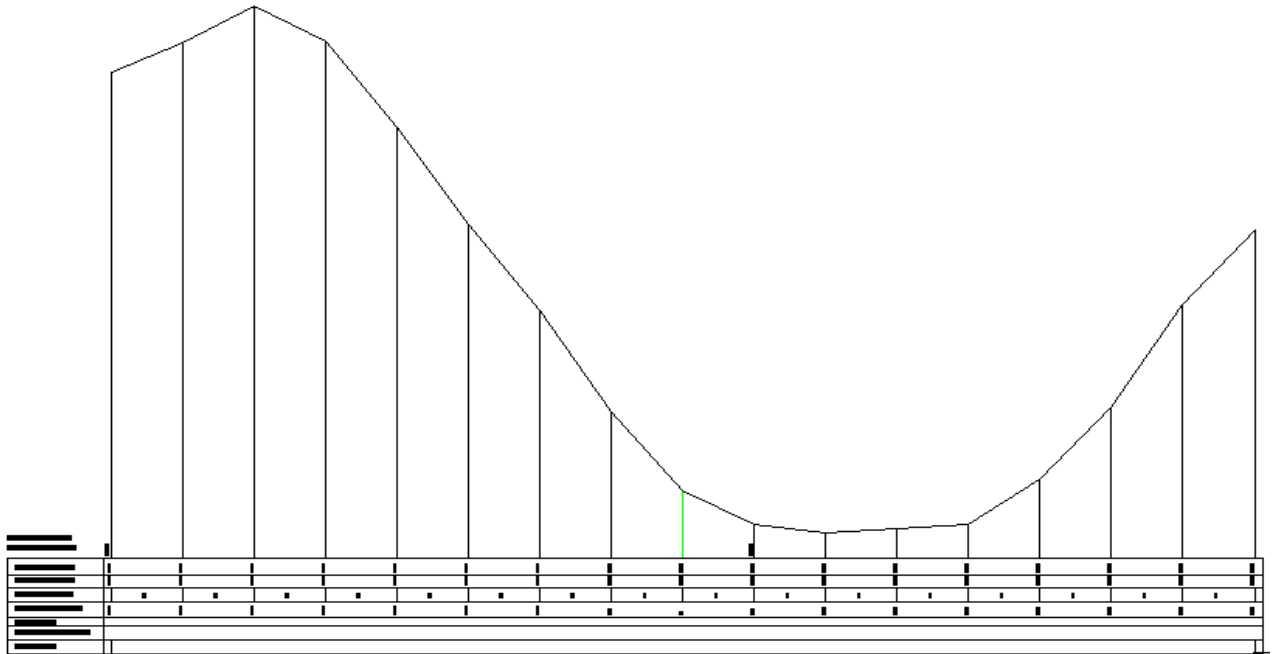
4853	11598	12	1757.0	1825.0	13.0
4853	11599	12	1754.0	1825.0	14.1
4853	11600	10	1758.0	1833.0	12.7
4853	11601	10	1763.0	1833.0	12.3
4853	11628	7	1968.0	2037.0	5.0
4854	11597	13	1772.0	1825.0	6.2
4854	11598	13	1763.0	1825.0	5.8
4854	11599	11	1764.0	1833.0	12.1
4854	11600	11	1762.0	1833.0	11.3
4854	11601	11	1763.0	1833.0	15.7
4855	11598	12	1775.0	1833.0	9.3
4855	11599	12	1767.0	1833.0	11.6
4855	11600	12	1770.0	1833.0	15.4
4855	11601	12	1762.0	1833.0	14.7
4855	11602	12	1775.0	1833.0	11.3
4856	11597	13	1778.0	1833.0	8.4
4856	11598	13	1773.0	1833.0	11.7
4856	11599	13	1767.0	1833.0	18.7
4856	11600	13	1765.0	1833.0	15.7
4856	11601	13	1772.0	1833.0	13.8
4909	11366	12	20.0	71.0	9.9
4909	11367	11	13.0	70.0	12.4
4909	11368	12	10.0	70.0	11.8
4911	11366	13	10.0	73.0	14.5
4913	11366	13	15.0	73.0	7.7
4913	11367	14	12.0	73.0	9.1
4913	11368	15	10.0	73.0	14.7
4913	11369	15	10.0	72.0	11.6
4914	11368	15	18.0	73.0	5.1
4914	11369	16	11.0	73.0	11.9
4928	11630	7	1732.0	1801.0	7.3
4928	11631	7	1737.0	1801.0	6.8
4929	11630	6	1729.0	1801.0	6.9
4929	11631	6	1739.0	1801.0	5.9
4930	11630	6	1728.0	1799.0	5.7
4980	11624	7	1118.0	1170.0	5.3
4980	11625	7	1118.0	1170.0	6.0
4981	11623	7	1124.0	1177.0	5.5
4981	11624	8	1116.0	1185.0	8.5
4981	11625	8	1115.0	1185.0	9.7
4981	11626	8	1119.0	1185.0	7.6
4982	11624	8	1112.0	1177.0	8.5
4982	11625	8	1112.0	1177.0	7.0
4982	11626	9	1119.0	1185.0	7.8
5026	11652	15	672.0	695.0	5.6
5027	11653	14	671.0	695.0	6.7
5028	11653	14	672.0	695.0	6.7
5465	4621	20	716.0	731.0	6.6
5466	4619	20	717.0	730.0	6.0
5466	4621	20	716.0	730.0	7.5
5811	11545	9	270.0	328.0	6.2
5811	11546	9	279.0	334.0	6.1
5811	11758	20	1337.0	1393.0	49.6
5812	11545	9	271.0	334.0	7.7
5812	11546	9	275.0	342.0	7.8
5812	11547	10	296.0	350.0	6.8
5812	11756	20	1338.0	1389.0	43.3
5812	11757	20	1338.0	1392.0	36.7

5813	11546	10	274.0	334.0	6.5
5813	11547	10	281.0	342.0	7.8
5815	11760	20	1339.0	1403.0	77.6
5815	11761	20	1339.0	1403.0	71.1
5816	11761	19	1340.0	1403.0	58.2
6766	11690	8	282.0	319.0	5.1
6767	11691	7	281.0	319.0	5.7

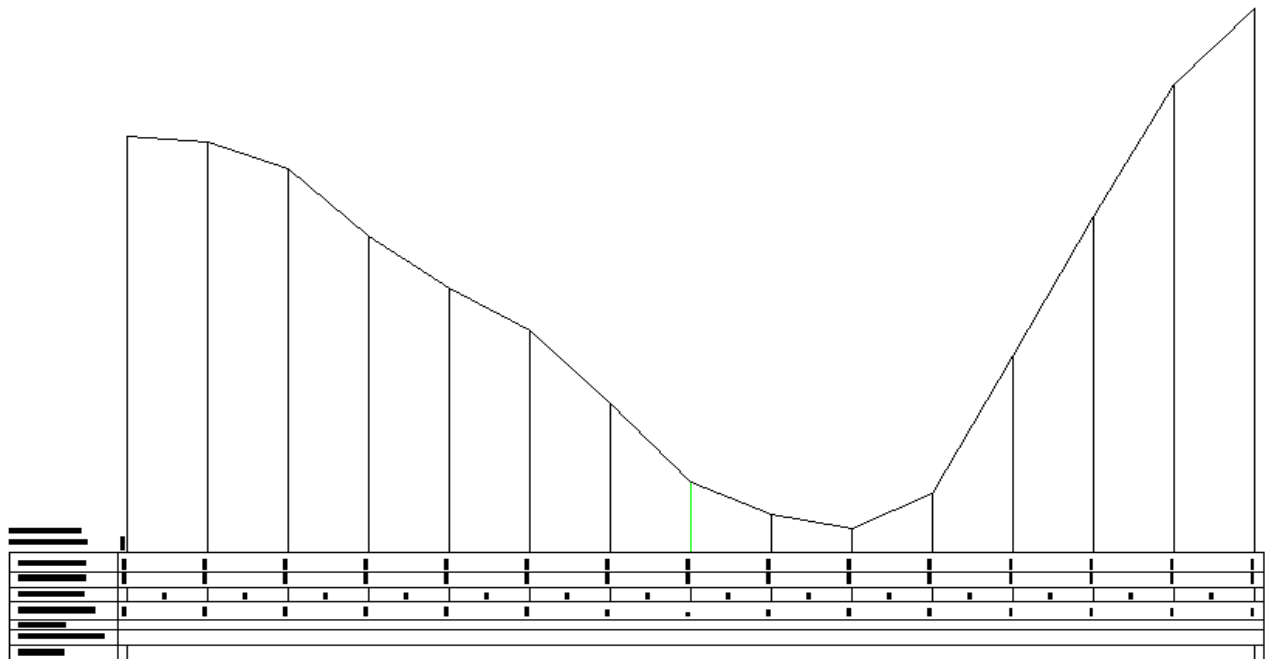
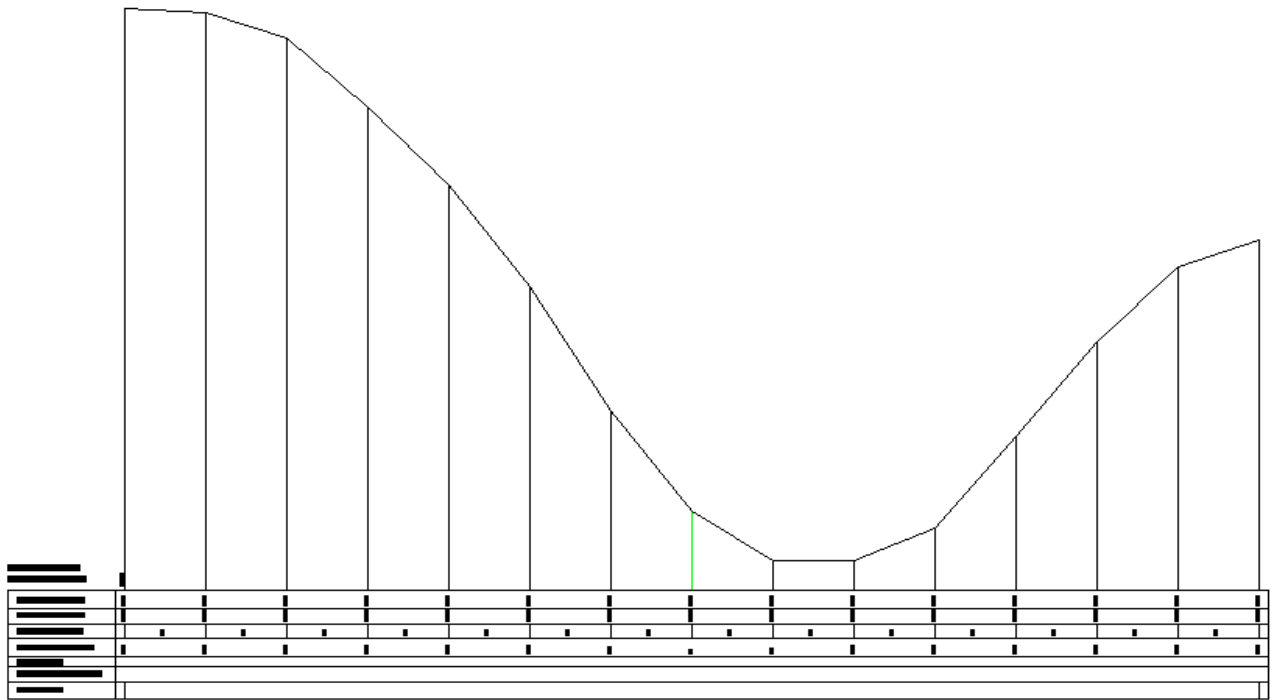
ΠΑΡΑΡΤΗΜΑ Α.4.2

ΜΗΚΟΤΟΜΕΣ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΤΕΤΑΡΤΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

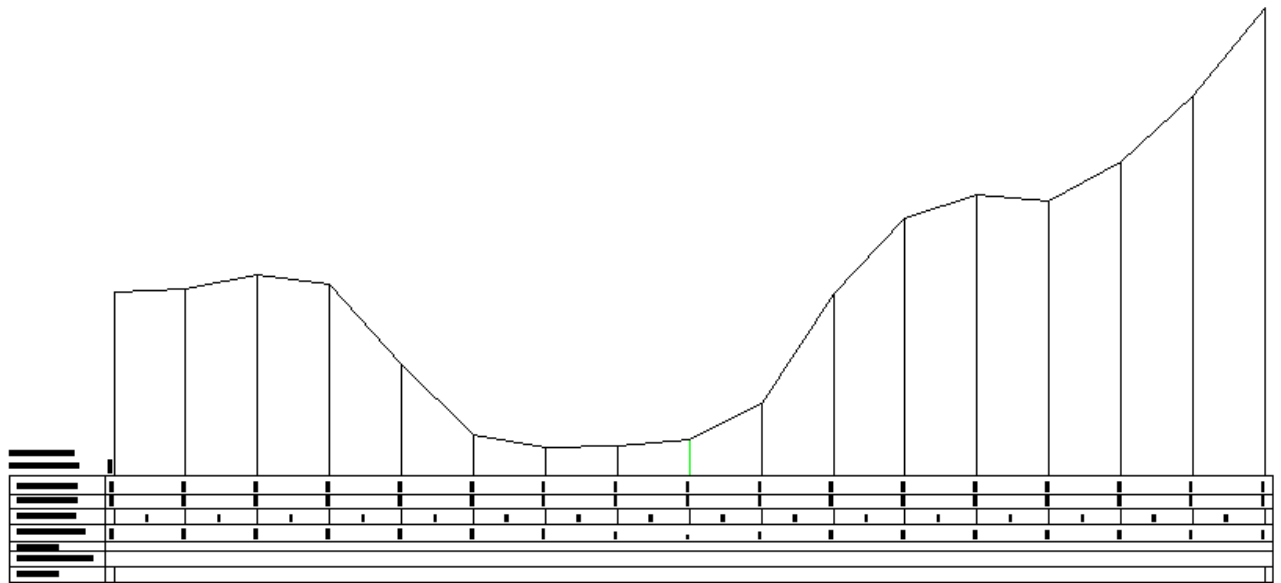
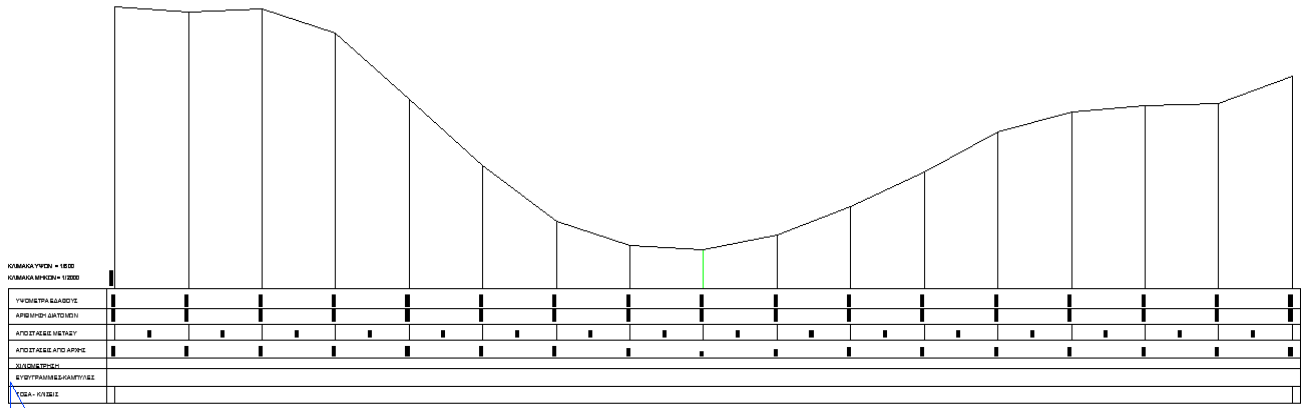
Μηκοτομές σημείου στήλης 547, γραμμής 6780



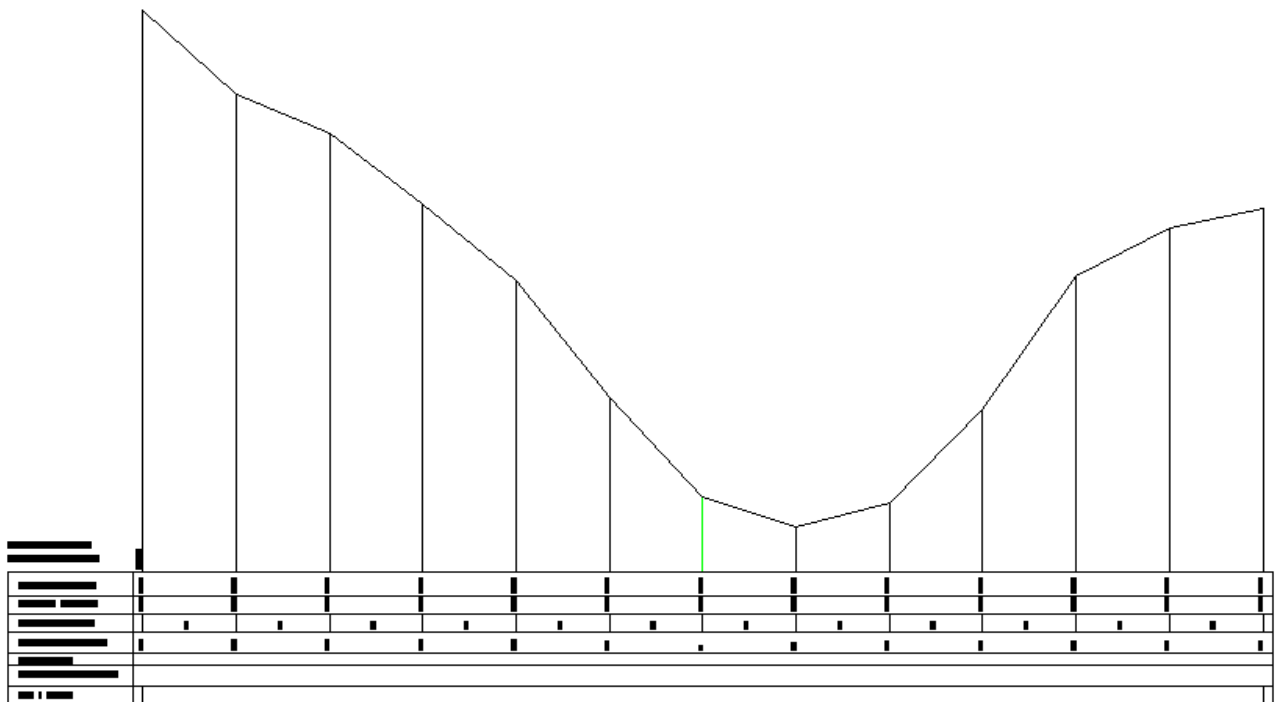
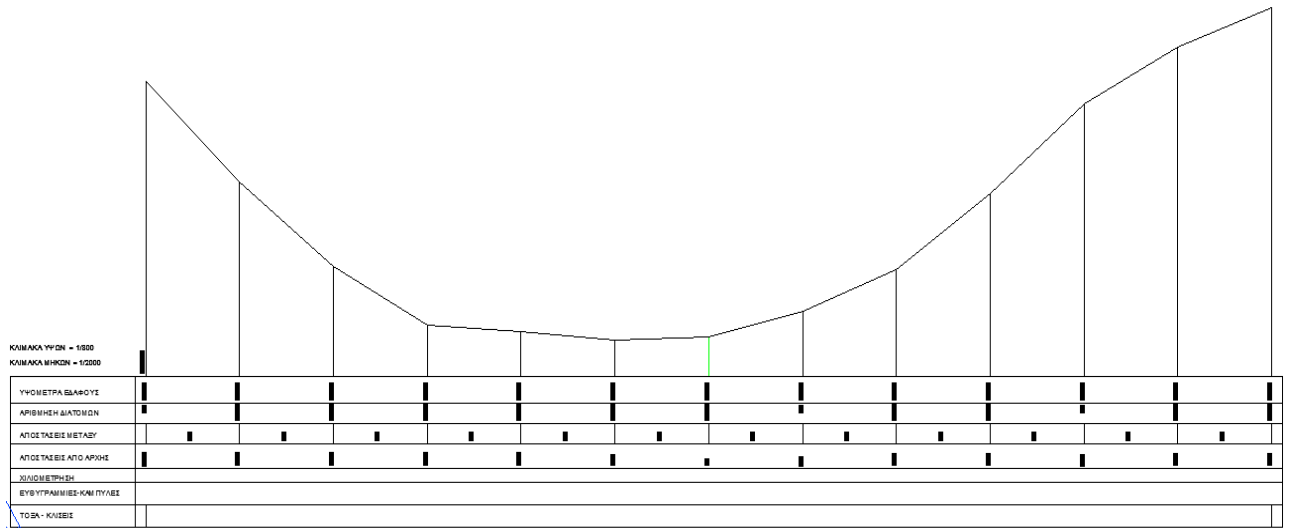
Μηκοτομές σημείου στήλης 1371, γραμμής 7504



Μηκοτομές σημείου στήλης 4102, γραμμής 7523



Μηκοτομές σημείου στήλης 4852, γραμμής 11627



ΠΑΡΑΡΤΗΜΑ Β

ΠΑΡΑΡΤΗΜΑ Β.1

ΠΡΟΓΡΑΜΜΑ limnodeksameni1

```
module moddemusgs

type DEMUSGS
  integer(4) :: nxcols, nyrows
  real(8) :: GDAL_NODATA, X0, Y0, DX, DY
  integer(2), allocatable :: him(:, :)
  character(len=1) :: form
end type DEMUSGS

type(DEMUSGS) :: a(6) !to orizw edw gia na einai global

contains

subroutine init
! kaleitai stin arxi tou programmatos gia na orisoume tous pinakes a(6)
! pairnume tous arxikous pinakes stis katalliles theseis. o 1 panw
aristera, o 2 panw deksia ... o 6 katw deksia
  implicit none
  integer(4) :: ierr
  call thanSet (a(1), 'srtm_41_04.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
  call thanSet (a(2), 'srtm_42_04.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
  call thanSet (a(3), 'srtm_41_05.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
  call thanSet (a(4), 'srtm_42_05.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
  call thanSet (a(5), 'srtm_41_06.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
  call thanSet (a(6), 'srtm_42_06.bin', ierr)
  if(ierr /= 0) stop 'kati pige strava'
end subroutine init

subroutine thanSet (self, filnam, ierr)
! "Read dem from .bin file."
  type(DEMUSGS), intent(inout) :: self
  character(len=*), intent(in) :: filnam
  integer(4), intent(out) :: ierr
  integer(4) uBin, irec, jx, iy
  character(len=1) form
  uBin = 31
  call openBmp (uBin, filnam, 'old', ierr)
  if (ierr /= 0) return !Could not open file
  do
    irec = 1
```

```

        read (uBin, rec=irec, iostat=ierr) self%form, self%GDAL_NODATA
        if (ierr /= 0) exit                !Syntax error
        ierr = 1
        if (self%form /= 'h') exit        !This object handles
only type 'h' (int16)
        irec = irec + 1 + 8
        read (uBin, rec=irec, iostat=ierr) self%nxcols, self%nyrows
        if (ierr /= 0) exit                !Syntax error
        ierr = 2
        if (self%nxcols == 0 .or. self%nyrows == 0) exit
!Invalid image
        irec = irec + 2*4
        read (uBin, rec=irec, iostat=ierr) self%X0, self%Y0, self%DX,
self%DY
        if (ierr /= 0) exit                !Syntax error
        if (self%DX == 0.0_8 .or. self%DY == 0.0_8) exit    !Invalid
image
        irec = irec + 4*8
!        print *, self%form, self%GDAL_NODATA
        print *, self%nxcols, self%nyrows
!        print *, self%X0, self%Y0, self%DX, self%DY
        if (self%form == 'h') then
            allocate (self%him(1:self%nxcols, 1:self%nyrows), stat=ierr)
            if (ierr /= 0) exit                !Out of memory
            do iy=1, self%nyrows
                read (uBin, rec=irec, iostat=ierr) self%him(:, iy)
                if (ierr /= 0) exit                !Syntax error
                irec = irec + self%nxcols*2
            end do
        else
            stop 'Unknown form'
        end if
        exit
    end do
    call closeBmp(uBin)
end subroutine thanSet

```

```

subroutine ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
! pairnei tis sunetetagmenes i kai j ston megalo pinaka kai
epistrefei se poion pinaka anikei to simeio auto (num_pinaka)
! kai poies einai oi suntetagmenes tou ston pinaka auton
(i_pinaka, j_pinaka)
    implicit none
    integer(4), intent(in) :: i, j
    integer(4), intent(out) :: num_pinaka, i_pinaka, j_pinaka
    integer(4) :: icol, jrow

    icol = 1                !deixnei se poia stili vrisketai o
pinakas
    jrow = 1                !deixnei se poia grammi vrisketai o
pinakas

    i_pinaka = i

```

```

        j_pinaka = j

        !kathe pinakas exei diastaseis 6001 x 6001
        IF (i_pinaka > 6001) THEN
            icol = icol + 1

            i_pinaka = i_pinaka - 6001
        END IF

        DO WHILE (j_pinaka > 6001)

            jrow = jrow + 1

            j_pinaka = j_pinaka - 6001
        END DO

        ! vriskw apo poio pinaka pairnw tin timi num_pinaka

        if ( icol == 2 ) then
            num_pinaka = icol * jrow
        else
            num_pinaka = (jrow - 1) * 2 + icol
        end if

    end subroutine ijpinaka

    real(8) function him(i, j)
        ! pairnei tis suntetagmenes (i,j) toy simeiou ston megalo pinaka
        ! epistrefei to upsometro aytou tou simeiou i -1 an to simeio den
        einai mesa ston pinaka
        implicit none
        integer(4) :: i,j, num_pinaka, i_pinaka, j_pinaka

        !epistrefei -1 an kapoio apo ta i, j den einai mesa ston
        megalo pinaka
        IF (i<1 .OR. i>12002 .OR. j<1 .OR. j>18003) THEN
            him = -1.0
            return
        END IF

        ! kalw to ijpinaka gia na parw ta stoixeia toy mikrou pinaka
        p anikei to simeio
        call ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
        him = a(num_pinaka)%him(i_pinaka, j_pinaka)
    end function him

    subroutine perimetroi(i_start,j_start,i_end,j_end)

```

```

! elegxei ta simeia apo tis 8esi (i_start, j_start) ews (i_end,
j_end) gia to poses perimetroi gyrw tous ikanopoion tis proupo8eseis p
exoume 8esei
        implicit none
        integer(4), intent(in) :: i_start, j_start, i_end, j_end
        integer(4) :: i, j, iper, itemp, jtemp
        real(8) :: a_max_next, a_min_next, a_min_prev,
a_max_prev
        logical :: log1, log2, log

        write(*,*) "start"
        open(unit=11, file='out.txt')
        write(11,*) "i          ", "j          ", "perimetroi"

        DO i=i_start, i_end
            if (mod(i, 100) == 0) write (*, *) i
            DO j=j_start, j_end

                iper = 0 ! (arxika den ikanopoieitai gia kai perimetro (to
idio apotelesma kai an to simeio einai <10m)

                IF (him(i,j) >= 10) THEN !elegxw mono an to simeio m exei
upsometro >=10

                    a_min_prev= him(i,j) !otan 8a elegksw me ta max kai min
tis lis perimetrou
                    a_max_prev= him(i,j) !elegxw me to upsometro tou simeiou
mou
                    DO
                        iper = iper + 1
                        a_min_next=100000.0 ! arxiki timi gia na vrw to min
tis epomenis perimetrou
                        a_max_next=0.0 !omoia gia max
                        DO itemp=(i-iper), (i+iper)
                            DO jtemp=(j-iper), (j+iper)
                                log1= ( (itemp==i-iper).or.(itemp==i+iper) ) !
8elw mono ta simeia tis perimetrou kai oxi auta p einai mesa sauti
                                log2= ( (jtemp==j-iper).or.(jtemp==j+iper) ) !
ara prp ena toulaxiston apo ta itemp k jtemp na einai stin perimetro
                                log= log1.or.log2 ! to
ekana etsi giati de xwrouse se mia grammi
                                IF(log) THEN
                                    !an kapoio simeio tis perimetrou <10m
stamataw na psaxnw
                                    IF ( him(itemp,jtemp) < 10.0 ) go to 100

                                    !vriskw max kai min tis perimetrou
                                    IF ( him(itemp,jtemp) < a_min_next) THEN
                                        a_min_next = him(itemp,jtemp)
                                    END IF
                                    IF ( him(itemp,jtemp) > a_max_next) THEN
                                        a_max_next = him(itemp,jtemp)
                                    END IF
                                END IF
                            END DO
                        END DO
                    END DO
                END DO
            END DO
        END DO

```

```

                END IF
            END DO
        END DO
        ! sugrinoyme min_next me max_prev tis proigoumenis
perimetrou
        !an den ikanopoieitai to kritirio stamataw na psaxnw
(exit)
        !kai meiw nw to iper kata 1 giati i perimetros p
elegksame teleutaia
        !den ikanopoioiouse ta kritiria
        IF ( a_min_next <= a_max_prev) THEN
100            iper = iper - 1

                exit
            END IF
        !An ikanopoieitai sunexizw ton elegxo stin epomeni
perimetro
        !twra to min_prev kai max_prev einai isa me to min_next
kai max_next
        !afou i perimetros p elegksame prin ginetai twra i
proigoumeni gia auti p elegxw twra
        a_min_prev = a_min_next
        a_max_prev = a_max_next
        END DO
        !grafw to apotelesma giauto to simeio
        if (iper >= 2) write(11,*) i, j, iper
    END IF
    END DO
    END DO
    write(*,*) "finished writing"

    end subroutine perimetroi

```

```

subroutine mikotomi(i, j, ipl, idieuth)
    ! dimiourgei ena arxeio txt me ta upsometra to opoio sti sunexeia
    xrisimopoieitai apo to programma gia na dimiourgisei mia mikotomi
    ! i,j einai i thesi tou simeiou ston pinaka kai ipl to plithos tw n
    perimetrwn ekso apo to simeio, tw n opoiwn simeia 8a sxediastoun
    ! idieuth orizei an i mikotomi 8a ginei me ta simeia tis idias
    grammis me to kentro mas i stin idia stili.
        implicit none
        integer(4), intent(in) :: i, j, ipl, idieuth
        integer(4) :: itemp
        real(8) :: xth

        write(12, 130) "mikotomi", i, j

        xth = ipl * (-90) ! xarakteristiki thesi
        if ( idieuth == 1) then !mikotomi kata ton x ara allazei mono
to x enw to y menei stathero
            do itemp= (i-ipl), (i+ipl)

```

```

        write(12,120) xth, him(itemp, j), itemp
        xth = xth + 90.0
    end do
    else      !mikotomi kata ton y ara allazei mono to y enw to
x stathero
        do itemp= (j-ipl), (j+ipl)
            write(12,120) xth, him(i, itemp), itemp
            xth = xth + 90.0
        end do
    end if

120     format(F10.2, F10.2, 1x, I8)

        write(12, 130) "$"
130     format (a, i6, i6)

        end subroutine mikotomi

end module moddemusgs

```

```

!=====
====
!====GNU G95 SPECIFIC ROUTINES
!=====
====

```

```

    subroutine openBmp (uBmpx, filnam, stat1, ierr)
    implicit none
    integer*4 uBmpx, ierr           ! ARGUMENTS
    character*(*) filnam, stat1    ! ARGUMENTS
    character*10 stat

    stat = stat1
    if (stat .eq. ' ') stat = 'unknown'

!-----GNU g77 compiler

    open (uBmpx, file=filnam, status=stat, iostat=ierr, &
&form='unformatted', access='stream')      ! It behaves like direct
file of recl=1

    return
    end subroutine openBmp

```

```

!=====
====

```

```

    subroutine closeBmp (uBmpx)
    implicit none
    integer*4 uBmpx           ! ARGUMENTS

    close (uBmpx)
    return
    end subroutine closeBmp

```

```
program limnodeksamen1
use moddemusgs

call init

call perimetro(1,1,12002,18003)

end program limnodeksamen1
```

ΠΑΡΑΡΤΗΜΑ Β.2

ΠΡΟΓΡΑΜΜΑ limnodeksameni2

```
module moddemusgs
```

```
type DEMUSGS
```

```
integer(4) :: nxcols, nyrows
```

```
real(8) :: GDAL_NODATA, X0, Y0, DX, DY
```

```
integer(2), allocatable :: him(:, :)
```

```
character(len=1) :: form
```

```
end type DEMUSGS
```

```
type(DEMUSGS) :: a(6) !to orizw edw gia na einai global
```

```
contains
```

```
subroutine init
```

```
!to kalw stin arxi tou programmatos gia na orisw tous pinakes a(6)
```

```
! pairnume tous arxikous pinakes stis katalliles theseis. o 1 panw
```

```
aristera, o 2 panw deksia ... o 6 katw deksia
```

```
implicit none
```

```
integer(4) :: ierr
```

```
call thanSet (a(1), 'srtm_41_04.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
call thanSet (a(2), 'srtm_42_04.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
call thanSet (a(3), 'srtm_41_05.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
call thanSet (a(4), 'srtm_42_05.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
call thanSet (a(5), 'srtm_41_06.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
call thanSet (a(6), 'srtm_42_06.bin', ierr)
```

```
if(ierr /= 0) stop 'kati pige strava'
```

```
end subroutine init
```

```
subroutine thanSet (self, filnam, ierr)
```

```
! "Read dem from .bin file."
```

```
type(DEMUSGS), intent(inout) :: self
```

```
character(len=*), intent(in) :: filnam
```

```
integer(4), intent(out) :: ierr
```

```
integer(4) uBin, irec, jx, iy
```

```
character(len=1) form
```

```
uBin = 31
```

```
call openBmp (uBin, filnam, 'old', ierr)
```

```
if (ierr /= 0) return !Could not open file
```

```
do
```

```
irec = 1
```

```
read (uBin, rec=irec, iostat=ierr) self%form, self%GDAL_NODATA
```

```
if (ierr /= 0) exit !Syntax error
```

```
irec = 1
```



```

        if (self%form /= 'h') exit                !This object handles
only type 'h' (int16)
        irec = irec + 1 + 8
        read (uBin, rec=irec, iostat=ierr) self%nxcols, self%nyrows
        if (ierr /= 0) exit                        !Syntax error
        ierr = 2
        if (self%nxcols == 0 .or. self%nyrows == 0) exit
!Invalid image
        irec = irec + 2*4
        read (uBin, rec=irec, iostat=ierr) self%X0, self%Y0, self%DX,
self%DY
        if (ierr /= 0) exit                        !Syntax error
        if (self%DX == 0.0_8 .or. self%DY == 0.0_8) exit    !Invalid
image
        irec = irec + 4*8
!        print *, self%form, self%GDAL_NODATA
print *, self%nxcols, self%nyrows
!        print *, self%X0, self%Y0, self%DX, self%DY
        if (self%form == 'h') then
            allocate (self%him(1:self%nxcols, 1:self%nyrows), stat=ierr)
            if (ierr /= 0) exit                    !Out of memory
            do iy=1, self%nyrows
                read (uBin, rec=irec, iostat=ierr) self%him(:, iy)
                if (ierr /= 0) exit                !Syntax error
                irec = irec + self%nxcols*2
            end do
        else
            stop 'Unknown form'
        end if
        exit
    end do
    call closeBmp(uBin)
end subroutine thanSet

```

```

subroutine ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
! dinw to i,j sto megalο pinaka kai epistrefei se poion pinaka
anikei to simeio (num_pinaka)
! kai to topiko i,j tou simeiou ston pinaka auton (i_pinaka,
j_pinaka)
    implicit none
    integer(4), intent(in) :: i, j
    integer(4), intent(out) :: num_pinaka, i_pinaka, j_pinaka
    integer(4) :: icol, jrow

    icol = 1
    jrow = 1

    i_pinaka = i
    j_pinaka = j

    IF (i_pinaka > 6001) THEN
        icol = icol + 1
    END IF

```

```

        i_pinaka = i_pinaka - 6001
    END IF

    DO WHILE (j_pinaka > 6001)

        jrow = jrow + 1

        j_pinaka = j_pinaka - 6001
    END DO

    ! vriskw apo poio pinaka pairnw tin timi num_pinaka

    if ( icol == 2 ) then
        num_pinaka = icol * jrow
    else
        num_pinaka = (jrow - 1) * 2 + icol
    end if

end subroutine ijpinaka

real(8) function him(i, j)
    ! dinw to i kai to j tou megalou pinaka kai epistrefei to
    upsometro apo tous mikrous pinakes
    !epistrefei -1 an kapoio apo ta i,j den einai mesa ston megalou
    pinaka
    implicit none
    integer(4) :: i,j, num_pinaka, i_pinaka, j_pinaka

    !epistrefei -1 an kapoio apo ta i, j den einai mesa ston
    megalou pinaka
    IF (i<1 .OR. i>12002 .OR. j<1 .OR. j>18003) THEN
        him = -1.0
        return
    END IF

    ! kalw to ijpinaka gia na parw ta stoixeia toy mikrou pinaka
    p anikei to simeio
    call ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
    him = a(num_pinaka)%him(i_pinaka, j_pinaka)
end function him

subroutine perimetro(i_start,j_start,i_end,j_end, istop, idam)
    !elegxei ta simeia apo ti 8esi (i_start,j_start) ews
    (i_end,j_end)
    ! istop dilwnei poses diadoxikes perimetroi prp na min
    ikanopoioun, gia na stamatisw na psaxnw
    ! idam dilwnei posa simeia tis perimetrou dexomai na exoun
    mikrotero upsometro
    implicit none

```

```

integer(4), intent(in) :: i_start, j_start, i_end, j_end,
istop, idam
integer(4) :: i, j, iper, icount, ipl
logical, dimension(160) :: fail
! elegxei mexri 20 perimetrous alliws allazw tis diastaseis (20*8)
real(8) :: a_max_next, a_max_prev

write(*,*) "start"
write(*,*) "i          ", "j          ", "perimetroi"

DO i=i_start, i_end
  if (mod(i, 100) == 0) write (*, *) i
  DO j=j_start, j_end

    iper = 0 ! (arxika den ikanopoieitai gia kamia perimetro (to
idio apotelesma kai an to simeio einai <10m)

    IF (him(i,j) >= 10) THEN !elegxw mono an to simeio m exei
upsometro >=10

      a_max_prev = him(i,j) !elegxw me to upsometro tou
simeiou mou
      icount = 0           ! midenizw tis perimetrous pou den
ikanopoion (elegxos gia termatismo me istop)
      DO

        iper = iper + 1
        !pairnw to plithos kai ti thesi tw'n mikroterwn
simeiw'n tis perimetrou kai to a_max tis perimetrou apti subroutine
        call find_pl(i, j, iper, idam, a_max_prev,
a_max_next, ipl, fail)

        ! xrisimopoiw ti function CHECK_PER gia na dw an i
perimetros me ipl k ipos ikanopoiei
        IF ( check_per(iper, idam, ipl, fail) ) THEN           !
i perimetros ikanopoiei sunexizw ton elegxw
          icount = 0           ! midenizw to icount afou
twra pleon metraw apo tin arxi mexri to istop gia na stamatisw
          a_max_prev = a_max_next ! max_prev einai iso me
to max_next afou i perimetros p elegksame prin ginetai twra i
proigoumeni gia auti p elegxw twra
        ELSE
          !i perimetros den
ikanopoiei
          icount = icount + 1 !auksanw ton arithmo tw'n
perimetrwn p den ikanopoion
          IF (icount > istop) THEN !otan o arithmos
perimetrwn > arithmo pou dexomai (istop) (px an istop=1 kai icount=1 exw
elegkseis 1 perimetro k prp na min ikanopoiei oute i 2i gia na
stamatisw)
            iper = iper - icount ! meiw'nw ton arithmo
tw'n perimetrwn p ikanopoion afou oi teleutaies icount den ikanopoiouan
            exit ! stamataw na elegxw
epomenes perimetrous

```

```

                                END IF

                                if (a_max_next > a_max_prev) a_max_prev =
a_max_next ! thelw to max apo tis perimetrous pou eixa kai tin nea pou
den ikanopoiise

                                END IF
                                END DO
                                !grafw to apotelesma giauto to simeio
                                if (iper>=3) write(11,*) i, j, iper
                                END IF
                                END DO
                                END DO
                                write(*,*) "finished writing"

                                end subroutine perimetri

subroutine find_pl(i, j, iper, idam, a_max_prev, a_max_next, ipl,
fail)
! pairnei tis suntetagmenes simeiou(i,j) tin perimetro
(iper) to pli8os twn simeiwv pou dexomai (idam) kai to max upsometro tis
proigoumenis perimetrou (a_max_prev)
! epistrefei to max upsometro tis perimetrou (a_max_next) to
plithos twn simeiwv p den ikanopoioun (ipl)
! fail einai LOGIKOS PINAKAS pou exei TRUE otan to simeio
den ikanopoiiei. i 8esi 1 antistoixei panw aristera--> katw aristera -->
katw deksia--> panw aristera (to apnw aristera simeio periexetai mono
mia fora)

    implicit none
    integer(4), intent(in) :: iper, i, j, idam
    real(8), intent(in) :: a_max_prev
    real(8), intent(out) :: a_max_next
    integer(4), intent(out) :: ipl
    logical, dimension(160), intent(out) :: fail
!elegxei mexri 20 perimetrous (20*8)
    integer(4), dimension(160,2) :: ipos
!omoia
    integer (4) :: itemp, jtemp

    call points_pos(i, j, iper, ipos) !pairnw tis 8eseis
twn simeiwv tis perimetrou

    do itemp = 1, 160
        fail(itemp) = .false.
    end do

    a_max_next=a_max_prev !8a vrw to max apo tis 2
perimetrous
    ipl = 0 !arxiki timi gia ti plithos twn
simeiwv me mikro upsometro
    DO itemp = 1, iper*8 !ka8e perimetros exei iper *
8 simeia

```

```

                IF (ipl <= idam) THEN                                ! otan exw
perasei to plithos idam den xreiazetai na elegxw allo gia epibleon
simeia
                IF ( him(ipos(itemp,1), ipos(itemp,2)) <=
a_max_prev) THEN
                    ipl = ipl + 1
                    fail(itemp) = .TRUE.                                !to
simeio auto den ikanopoiei sto pinaka 8a exei .true.
                    END IF
                END IF
                !vriskw to max_next (prokuptei apo to max tw 2
perimetrwn)
                if (him(ipos(itemp,1), ipos(itemp,2)) > a_max_next)
then
                    a_max_next = him(ipos(itemp,1), ipos(itemp,2))
                end if

                END DO

                end subroutine find_pl

                subroutine points_pos(itemp, jtemp, iper, ipos)
                ! pairnei ti thesi (itemp, jtemp) tou simeiou pou eksetazw kai
iper tin perimetro. kai tupwnei ta simeia tis perimetrou arxizontas apo
to
                ! panw aristera --> katw aristera --> katw deksia --> panw deksia
--> panw aristera (xwris na epistrefei to lo simeio 2 fores)

                implicit none
                integer(4), intent(in) :: itemp, jtemp, iper
                integer(4) :: i, j, ichange, icount, iplevra, k
                integer(4), intent(inout) :: ipos(160,2)
                !80 giati upo8etw mexri 10 perimetrous 8a ftasw ton elegxw ( alliws
prepei na allaksw kai sto find pl)

                ichange = iper * 2    !posa simeia exei ka8e perimetros (i li
perimetros exei ichange + 1)

                i = itemp - iper
                j = jtemp - iper
                iplevra = 1            !metraei se poia plevra vriskomai
                icount = 0            !metraw poses fores exw allaksei simeio
                k = 0                !metraei se poia 8esi tou pinaka grafw
                DO

                    k = k + 1
                    ipos(k,1) =i
                    ipos(k,2) =j

                    if (iplevra == 1) then                                !elegxw se poia plevra
vriskomai
                        j = j + 1
                    else if (iplevra == 2) then
                        i = i + 1
                    else if (iplevra == 3) then
                        j = j - 1
                    else

```

```

        i = i - 1
    end if

    icount = icount + 1

    if (icount >= ichange) then
        iplevra = iplevra + 1
        icount = 0
    end if

    if (iplevra >= 5) exit
end do

end subroutine points_pos

logical function check_per(iper, idam, ipl, fail)
! pairnei plithos(ipl) kai theseis (fail) tw'n simeiw'n tis
perimetrou iper p exoun mikrotero upsometro kai posa tetoia simeia
dexomai (idam)
! epistrefei TRUE an i perimetros ikanopoiei kai FALSE an oxi
    implicit none
    integer(4), intent(in) :: iper, idam, ipl
    logical, dimension(160), intent(in) :: fail
    integer(4) :: i, j, itemp, icount
    logical :: check

    IF ( ipl == 0) THEN
        check_per = .TRUE. !ola ta simeia exoun megalutero
upsometro
    ELSE IF ( ipl <= idam) THEN
        !elegxw an ta simeia mou einai sunexomena
        do i = 1, 8 * iper
            check = .FALSE.
            IF ( fail(i) ) then
                check = .true.
                icount = 1
                DO j=(i+1), iper*8 !prepei kai ta epomena ipl
simeia na min ikanopoioun (na einai TRUE) etc wste ola ta simeia pou den
ikanopoioun na einai sunexomena
                    if ( fail(j) .eqv. .TRUE.) then
                        icount = icount + 1
                    else
                        exit
                    end if
                END DO

                if (icount == ipl) then !ola ta simeia sunexomena
                    check_per = .true.

                else if (i==1) then !elegxw to lo simeio ws
gwniako
                    check_per = .true.
                    do j= 1, (ipl - icount) !elegxw tosa teleutaia
simeia

```

```

                                if ( fail(8 * iper - j + 1) .eqv. .false.
) then    ! an kapoio apo auta ta simeia ikanopoioun
                                check_per = .false.
                                exit
                                end if
                                end do
                                end if

                                END IF

                                IF ( check .eqv. .true. ) exit
                                END DO

                                ELSE ! uparxoun perissotera simeia me mikrotero upsometro
apoti dexomai (ipl > idam)
                                check_per = .FALSE.
                                END IF

                                end function check_per

```

```

subroutine l_f(i, j, al, f)
!pairnei ti thesi (i,j) tou simeiou st megalo pinaka kai
epistrefei to l kai to f me vasi ta x0,y0
    implicit none
    integer(4), intent(in) :: i, j
    real(8), intent(out) :: al, f
    integer(4) :: num_pinaka, i_pinaka, j_pinaka

    !painrume ton pinaka p anikei to simeio mas kai ta i,j st
topiko sustima tou pinaka
    call ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
    al = a(num_pinaka)%X0 + (i_pinaka - 1) * a(num_pinaka)%DX
! oso pigainume pros ta "deksia" to l auksanei
    f = a(num_pinaka)%Y0 - (j_pinaka - 1) * a(num_pinaka)%DY
! oso pigainume pros ta "katw" to f meiwnetai

    end subroutine l_f

```

```

subroutine mikotomi(i, j, ipl, idieuth)
! tupwnei se ena arxeio ta upsometra. to arxeio auto
xrisimopoietai apo to programma c_mhk gia na dimiourgisei mia mikotomi
! i,j einai i thesi tou simeiou ston pinaka kai ipl to plithos twn
perimetrwn ekso apo to simeio, twn opoiwn simeia 8a sxediastoun
! idieuth orizei an i mikotomi 8a ginei me ta simeia tis idias
grammis me to kentro mas i stin idia stili.
    implicit none
    integer(4), intent(in) :: i, j, ipl, idieuth
    integer(4) :: itemp, jtemp
    integer(4), dimension(160,2) :: ipos
    real(8) :: xth

    xth = ipl * (-90) ! xarakteristiki thesi

```

```

        if ( idieuth == 1) then !mikotomi kata ton x ara allazei mono
to x enw to y menei stathero
        write(12, 130) "mikotomi kata x", i, j
        do itemp= (i-ipl), (i+ipl)

                write(12,120) xth, him(itemp, j), itemp

                xth = xth + 90.0
        end do
        else if ( idieuth == 2) then      !mikotomi kata ton y ara
allazei mono to y enw to x stathero
        write(12, 130) "mikotomi kata y", i, j
        do itemp= (j-ipl), (j+ipl)
                write(12,120) xth, him(i, itemp), itemp
                xth = xth + 90.0
        end do
        else
        write(12,130) "mikotomi perimetrou", i, j
        xth = 0.0
        do jtemp = ipl, ipl+1
                call points_pos(i, j, jtemp, ipos)

                do itemp = 1, jtemp * 8
                        write(12,120) xth, him(ipos(itemp,1),
ipos(itemp,2)), itemp
                        xth = xth + 90.0
                end do
                write(12,120) xth, him(ipos(1,1), ipos(1,2)), itemp +
1
                xth = xth + 90.0
        end do
        end if

120         format(F10.2, F10.2, 1x, I8)

        write(12, 130) "$"
130         format (a, i6, i6)

        end subroutine mikotomi

end module moddemusgs

!=====
====
!====GNU G95 SPECIFIC ROUTINES
!=====
====

subroutine openBmp (uBmpx, filnam, stat1, ierr)
implicit none
integer*4 uBmpx, ierr                ! ARGUMENTS
character*(*) filnam, stat1         ! ARGUMENTS
character*10 stat

```



```

stat = stat1
if (stat .eq. ' ') stat = 'unknown'

!-----GNU g77 compiler

      open (uBmpx, file=filnam, status=stat, iostat=ierr, &
&form='unformatted', access='stream')      ! It behaves like direct
file of recl=1

      return
      end subroutine openBmp

!=====
=====

      subroutine closeBmp (uBmpx)
      implicit none
      integer*4 uBmpx                                ! ARGUMENTS

      close (uBmpx)
      return
      end subroutine closeBmp

program limnodeksameni2
use moddemusgs
implicit none
integer(4) :: istop, idam

call init

open(unit=11, file='out.txt')
istop = 5
idam = 3
call perimetro(1, 1, 12002,18003, istop, idam)
close (11)

end program limnodeksameni2

```

ΠΑΡΑΡΤΗΜΑ Β.3

ΠΡΟΓΡΑΜΜΑ limnodeksameni3

```
module moddemusgs
```

```
type DEMUSGS
```

```
integer(4) :: nxcols, nyrows  
real(8) :: GDAL_NODATA, X0, Y0, DX, DY  
integer(2), allocatable :: him(:, :)  
character(len=1) :: form
```

```
end type DEMUSGS
```

```
type(DEMUSGS) :: a(6) !to orizw edw gia na einai global
```

```
contains
```

```
subroutine init
```

```
! to kalw stin arxi tou programmatos gia na orisw tous pinakes a(6)  
! pairnume tous arxikous pinakes stis katalliles theseis. o 1 panw  
aristera, o 2 panw deksia ... o 6 katw deksia
```

```
implicit none  
integer(4) :: ierr  
call thanSet (a(1), 'srtm_41_04.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'  
call thanSet (a(2), 'srtm_42_04.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'  
call thanSet (a(3), 'srtm_41_05.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'  
call thanSet (a(4), 'srtm_42_05.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'  
call thanSet (a(5), 'srtm_41_06.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'  
call thanSet (a(6), 'srtm_42_06.bin', ierr)  
if(ierr /= 0) stop 'kati pige strava'
```

```
write(*,*) ""
```

```
end subroutine init
```

```
subroutine thanSet (self, filnam, ierr)
```

```
! "Read dem from .bin file."
```

```
type(DEMUSGS), intent(inout) :: self  
character(len=*), intent(in) :: filnam  
integer(4), intent(out) :: ierr  
integer(4) uBin, irec, jx, iy  
character(len=1) form  
uBin = 31  
call openBmp (uBin, filnam, 'old', ierr)  
if (ierr /= 0) return !Could not open file  
do  
irec = 1  
read (uBin, rec=irec, iostat=ierr) self%form, self%GDAL_NODATA  
if (ierr /= 0) exit !Syntax error
```

```

        ierr = 1
        if (self%form /= 'h') exit !This object handles
only type 'h' (int16)
        irec = irec + 1 + 8
        read (uBin, rec=irec, iostat=ierr) self%nxcols, self%nyrows
        if (ierr /= 0) exit !Syntax error
        ierr = 2
        if (self%nxcols == 0 .or. self%nyrows == 0) exit
!Invalid image
        irec = irec + 2*4
        read (uBin, rec=irec, iostat=ierr) self%X0, self%Y0, self%DX,
self%DY
        if (ierr /= 0) exit !Syntax error
        if (self%DX == 0.0_8 .or. self%DY == 0.0_8) exit !Invalid
image
        irec = irec + 4*8
!        print *, self%form, self%GDAL_NODATA
        print *, self%nxcols, self%nyrows
!        print *, self%X0, self%Y0, self%DX, self%DY
        if (self%form == 'h') then
            allocate (self%him(1:self%nxcols, 1:self%nyrows), stat=ierr)
            if (ierr /= 0) exit !Out of memory
            do iy=1, self%nyrows
                read (uBin, rec=irec, iostat=ierr) self%him(:, iy)
                if (ierr /= 0) exit !Syntax error
                irec = irec + self%nxcols*2
            end do
        else
            stop 'Unknown form'
        end if
        exit
    end do
    call closeBmp(uBin)
end subroutine thanSet

```

```

subroutine ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
! dinw ti thesi toy simeiou sto megalo pinaka kai epistrefei se
poion pinaka anikei to simeio (num_pinaka)
! kai to topiko i,j tou simeiou ston pinaka auton (i_pinaka,
j_pinaka)

```

```

    implicit none
    integer(4), intent(in) :: i, j
    integer(4), intent(out) :: num_pinaka, i_pinaka, j_pinaka
    integer(4) :: icol, jrow

```

```

        icol = 1
        jrow = 1

```

```

        i_pinaka = i
        j_pinaka = j

```

```

        IF (i_pinaka > 6001) THEN
            icol = icol + 1

```

```

        i_pinaka = i_pinaka - 6001
    END IF

    DO WHILE (j_pinaka > 6001)

        jrow = jrow + 1

        j_pinaka = j_pinaka - 6001
    END DO

    ! vriskw apo poio pinaka pairnw tin timi num_pinaka

    if ( icol == 2 ) then
        num_pinaka = icol * jrow
    else
        num_pinaka = (jrow - 1) * 2 + icol
    end if

end subroutine ijpinaka

real(8) function him(i, j)
    ! dinw ti thesi tou simeiou sto megalo pinaka kai epistrefei to
    upsometro apo tous mikrous pinakes
    !epistrefei -1 an kapoio apo ta i,j den einai mesa ston megalo
    pinaka
        implicit none
        integer(4) :: i,j, num_pinaka, i_pinaka, j_pinaka

        !epistrefei -1 an kapoio apo ta i, j den einai mesa ston
    megalo pinaka
        IF (i<1 .OR. i>12002 .OR. j<1 .OR. j>18003) THEN
            him = -1.0
            return
        END IF

        ! kalw to ijpinaka gia na parw ta stoixeia toy mikrou pinaka
    p anikei to simeio
        call ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
        him = a(num_pinaka)%him(i_pinaka, j_pinaka)
        if (him == a(1)%GDAL_NODATA) him = -1.0
    end function him

subroutine perimetroi(i_start,j_start,i_end,j_end, istop, idam)
    !elegxei ta simeia apo ti 8esi (i_start,j_start) ews
    (i_end,j_end)

```

```

! i metavliti istop dilwnei poses diadoxikes perimetroi prp na min
ikanopoioun, gia na stamatisw na elegxw
! i metavliti idam dilwnei posa simeia tis perimetrou dexomai na
min exoun to apaitoumeno upsometro

implicit none
integer(4), intent(in) :: i_start, j_start, i_end, j_end,
istop, idam
integer(4) :: i, j, iper, icount, ipl
logical, dimension(160) :: fail
! elegxei mexri 20 perimetrous alliw allazw tis diastaseis (20*8)
real(8) :: a_min_next, a_min_prev, temp

write(*,*) "start"

open(unit=11, file='out.txt')
write(11,*) "i ", "j ", "iper ", "min ", "max ",
"volume "

DO i=i_start, i_end
if (mod(i, 100) == 0) write (*, *) i
DO j=j_start, j_end

iper = 0 ! (arxika den ikanopoietai gia kamia perimetro (to
idio apotelesma kai an to simeio einai <10m)

IF (him(i,j) >= 10) THEN !elegxw mono an to simeio m exei
upsometro >=10

a_min_prev = him(i,j) !elegxw me to upsometro tou
simeiou mou
icount = 0 ! midenizw tis perimetrous pou den
ikanopoioun (elegxos gia termatismo me istop)
DO
iper = iper + 1
if (iper > 20) then ! ama elegxsw 20 perimetrous
stamataw ton elegxo
iper = 20 - icount ! afairw tis teleutaies
icount perimetrous pou den ikanopoioun
exit
end if
!pairnw to plithos kai ti thesi tw n mikroterwn
simeiw n tis perimetrou kai to a_max tis perimetrou apti subroutine
call find_pl(i, j, iper, idam, a_min_prev,
a_min_next, ipl, fail)

! xrisimopoiw ti function CHECK_PER gia na dw an i
perimetros me ipl k ipos ikanopoiiei
IF ( check_per(iper, idam, ipl, fail) ) THEN !
i perimetros ikanopoiiei sunexizw ton elegxw
icount = 0 ! midenizw to icount afou
twra pleon metraw apo tin arxi mexri to istop gia na stamatisw
a_min_prev = a_min_next ! max_prev einai iso me
to max_next afou i perimetros p elegksame prin ginetai twra i
proigoumeni gia auti p elegxw twra
ELSE !i perimetros den
ikanopoiiei

```

```

                icount = icount + 1    !auksanw ton arithmo twm
perimetrwn p den ikanopoioun
                IF (icount > istop) THEN    !otan o arithmos
perimetrwn > arithmo pou dexomai (istop) (px an istop=1 kai icount=1 exw
elegkseis 1 perimetro k prp na min ikanopoiiei oute i 2i gia na
stamatisw)
                iper = iper - icount    ! meiwntw ton arithmo
twm perimetrwn p ikanopoioun afou oi teleutaies icount den ikanopoiouan
                exit                    ! stamataw na elegxw
epomenes perimetrous
                END IF
                END IF
            END DO
            !grafw to apotelesma giauto to simeio

            if (iper >= 1) then
                temp = volume(i, j, him(i,j), a_min_prev)/1000000.0d0
                if (temp >= 5) write(11,999) i, j, iper, him(i,j),
a_min_prev, temp
            end if
999            format(I5, 2x, I5, 2x, I2, 2x, F6.1, 2x, F6.1, 2x, F4.1)
                END IF
            END DO
        END DO
        write(*,*) "finished writing"
        close(11)

        end subroutine perimetroi

        subroutine find_pl(i, j, iper, idam, a_min_prev, a_min_next, ipl,
fail)
            ! pairnei tis suntetagmenes simeiou(i,j) tin perimetro
(iper) to pli8os twm simeiwntw pou dexomai (idam) kai to min upsometro tis
proigoumenis perimetrou (a_min_prev)
            ! epistrefei to min upsometro tis perimetrou
(a_min_next)ektos apo ta upsometra pou einai sunexomena, to plithos twm
simeiwntw p den ikanopoioun (ipl)
            ! fail einai LOGIKOS PINAKAS pou exei TRUE otan to simeio
den ikanopoiiei. i 8esi 1 antistoixei panw aristera--> katw aristera -->
katw dekcia--> panw aristera (to apnw aristera simeio periexetai mono
mia fora)

            implicit none
            integer(4), intent(in) :: iper, i, j, idam
            real(8), intent(in) :: a_min_prev
            real(8), intent(out) :: a_min_next
            integer(4), intent(out) :: ipl
            logical, dimension(160), intent(out) :: fail
!elegxei mexri 20 perimetrous (20*8)
            integer(4), dimension(160,2) :: ipos
!omoia
            integer (4) :: itemp, jtemp, iout
            real(8) :: amin

```

```

                call points_pos(i, j, iper, ipos)           !pairnw tis 8eseis
tw n simeiw n tis perimetrou

                do itemp = 1, 160                          !arxikopoiisi
pinaka
                fail(itemp) = .false.
                end do

                ipl = 0                                     !arxiki timi gia ti plithos tw n
simeiw n me mikro upsometro
                a_min_next = -1000000.0                   ! gia na vrw to lo min
arkei na einai megalutero apo 0
                DO WHILE(ipl <= idam .and. check_per(iper, idam, ipl, fail)
.eqv. .true.)      !ka8e perimetros exei iper * 8 simeia
                call find_min(ipos, iper, a_min_next, amin, iout)
                a_min_next = amin                         ! ananewnw to cur_min gia otan
psaksw epomeno simeio, episis an vgw meta apo auti tin klisi tis
subroutine auto 8a einai to min autis tis perimetrou, afou ta
proigoumena ikanopoiou san
                IF ( amin <= a_min_prev) THEN
                ipl = ipl + 1
                fail(iout) = .TRUE.                       !to simeio
auto den ikanopoi ei sto pinaka 8a exei .true.
                if (a_min_next < 0) then                  ! exw
simeio me gdal_nodata i ektos pinaka mesa stin perimtro ara i perimetros
aporiptetai
                ipl = idam + 1
                exit
                end if
                ELSE
                exit      !stamataw na psaxnw molis ena amin >
a_min_prev (tote i perimetros ikanopoi ei an ikanopoiountai kai auta mesa
sto while)
                END IF

                END DO

        end subroutine find_pl

        subroutine points_pos(itemp, jtemp, iper, ipos)
        ! pairnei ti thesi (itemp, jtemp) tou simeiou pou eksetazw kai tin
perimetro(iper) kai epistrefei ta simeia tis perimetrou arxizontas apo
to
        ! panw aristera --> katw aristera --> katw deksia --> panw deksia
--> panw aristera (xwris na epistrefei to lo simeio 2 fores)

        implicit none
        integer(4), intent(in) :: itemp, jtemp, iper
        integer(4) :: i, j, ichange, icount, iplevra, k
        integer(4), intent(inout) :: ipos(160,2)
!160 giati upo8etw mexri 20 perimetrous 8a ftasw ton elegxw ( alliws
prepei na allaksw kai sto find pl)

        ichange = iper * 2      !posa simeia exei ka8e perimetros (i li
perimetros exei ichange + 1)

        i = itemp - iper

```

```

j = jtemp - iper
iplevra = 1          !metraei se poia plevra vriskomai
icount = 0          !metraw poses fores exw allaksei simeio
k = 0              !metraei se poia 8esi tou pinaka grafw
DO

    k = k + 1
    ipos(k,1) =i
    ipos(k,2) =j

    if (iplevra == 1) then                !elegxw se poia plevra
vriskomai
        j = j + 1
    else if (iplevra == 2) then
        i = i + 1
    else if (iplevra == 3) then
        j = j - 1
    else
        i = i - 1
    end if

    icount = icount + 1

    if (icount >= ichange) then
        iplevra = iplevra + 1
        icount = 0
    end if

    if (iplevra >= 5) exit
end do

end subroutine points_pos

subroutine find_min(ipos, iper, a_cur_min, amin, iout)
    ! vriskei to elaxiston upsometro(amin) gia tin perimetro iper
    me tis 8eseis tw n simeiw n ipos. epistrefei kai ti 8esi panw ston pinaka
    ipos (iout)
    ! to min 8a prepei na einai megalutero apo to a_cur_min pou
    einai to proigoumeno amin pou exw vrei ews twra

    implicit none
    integer(4), intent(in) :: iper
    integer(4), dimension(160,2), intent(in) :: ipos
    real(8), intent(in) :: a_cur_min
    real(8), intent(out) :: amin
    integer(4), intent(out) :: iout
    integer(4) :: i
    real(8) :: temp

    amin = 10000.0d0
    do i=1, 8*iper
        temp = him(ipos(i,1), ipos(i,2))
        if (temp >= a_cur_min .and. temp <= amin) then
            amin = temp
            iout = i
        end if
    end do
end subroutine find_min

```



```

        end if
    end do

end subroutine find_min

logical function check_per(iper, idam, ipl, fail)
! pairnei plithos(ipl) kai tis theseis (fail) tw'n simeiw'n tis
perimetrou iper p exoun mikrotero upsometro kai posa tetoia simeia
dexomai (idam)
! epistrefei TRUE an i perimetros ikanopoiei kai FALSE an oxi
implicit none
integer(4), intent(in) :: iper, idam, ipl
logical, dimension(160), intent(in) :: fail
integer(4) :: i, j, itemp, icount
logical :: check

IF ( ipl == 0 ) THEN
    check_per = .TRUE. !ola ta simeia exoun megalutero
upsometro
ELSE IF ( ipl <= idam ) THEN
    !elegxw an ta simeia mou einai sunexomena
    check = .FALSE.
    do i = 1, 8 * iper
        IF ( fail(i) ) then
            check = .true.
            icount = 1
            DO j=(i+1), iper*8 !prepei kai ta epomena ipl
simeia na min ikanopoioun (na einai TRUE) etc wste ola ta simeia pou den
ikanopoioun na einai sunexomena
                if ( fail(j) .eqv. .TRUE.) then
                    icount = icount + 1
                else
                    exit
                end if
            END DO

            if ( icount == ipl ) then !ola ta simeia sunexomena
                check_per = .true.

gwniako
                else if ( i==1 ) then !elegxw to lo simeio ws
                    check_per = .true.
                    do j= 1, (ipl - icount) !elegxw tosa teleutaia
simeia
                        if ( fail(8 * iper - j + 1) .eqv. .false.
) then ! an kapoio apo auta ta simeia ikanopoioun
                            check_per = .false.
                            exit
                        end if
                    end do
                end if
            END IF

            IF ( check .eqv. .true. ) exit

```

```

                END DO

                ELSE ! uparxoun perissotera simeia me mikrotero upsometro
apoti dexomai (ipl > idam)
                    check_per = .FALSE.
                END IF

end function check_per

subroutine l_f(i, j, al, f)
!pairnei ti thesi (i,j) tou simeiou st megalo pinaka kai
epistrefei to l kai to f me vasi ta x0,y0
    implicit none
    integer(4), intent(in) :: i, j
    real(8), intent(out) :: al, f
    integer(4) :: num_pinaka, i_pinaka, j_pinaka

!painrume ton pinaka p anikei to simeio mas kai ta i,j st
topiko sustima tou pinaka
    call ijpinaka(i, j, num_pinaka, i_pinaka, j_pinaka)
    al = a(num_pinaka)%X0 + (i_pinaka - 1) * a(num_pinaka)%DX
    f = a(num_pinaka)%Y0 - (j_pinaka - 1) * a(num_pinaka)%DY

end subroutine l_f

subroutine mikotomi(i, j, ipl, idieuth)
! gia idieuth=1 kanai mikotomi kata x apo to simeio i,j me ipl
perimetrous deksia k aristera
! gia idieuth=2 mikotomi kata y
! gia idieuth=3 mikotomi tis perimetrou ipl ksekinontas apo to
panw aristera simeio, panw katw aristera... kleinei me to panw aristera
ksana kai sunexizei me tin epomeni perimetro.
    implicit none
    integer(4), intent(in) :: i, j, ipl, idieuth
    integer(4) :: itemp, jtemp
    integer(4), dimension(160,2) :: ipos
    real(8) :: xth

    xth = ipl * (-90) ! xarakteristiki thesi
    if ( idieuth == 1) then !mikotomi kata ton x ara allazei mono
to x enw to y menei stathero
        write(12, 130) "mikotomi kata x", i, j
        do itemp= (i-ipl), (i+ipl)

            write(12,120) xth, him(itemp, j), itemp

            xth = xth + 90.0
        end do
    else if ( idieuth == 2) then !mikotomi kata ton y ara
allazei mono to y enw to x stathero

```

```

        write(12, 130) "mikotomi kata y", i, j
        do itemp= (j-ipl), (j+ipl)
            write(12,120) xth, him(i, itemp), itemp
            xth = xth + 90.0
        end do
    else
        write(12,130) "mikotomi perimetrou", i, j
        xth = 0.0
        do jtemp = ipl, ipl+1
            call points_pos(i, j, jtemp, ipos)

            do itemp = 1, jtemp * 8
                write(12,120) xth, him(ipos(itemp,1),
ipos(itemp,2)), itemp
                xth = xth + 90.0
            end do
            write(12,120) xth, him(ipos(1,1), ipos(1,2)), itemp +
1
            xth = xth + 90.0
        end do
    end if

120     format(F10.2, F10.2, 1x, I8)

        write(12, 130) "$"
130     format (a, i6, i6)

    end subroutine mikotomi

    subroutine isoupsis(i_in, j_in, upsometro, pos)
        ! dimiourgei tin isoupsi gia to upsometro pou dinetai, gurw apo to
simeio me thesi(i_in,j_in)
        ! kai epistrefei ton pinaka pos pou exei tis suntetagmenes se
metra twv simeiwv tis isoupsous (suntetagmenes sto top sustima)
        implicit none
        integer(4), intent(in) :: i_in, j_in
        real(8), intent(in) :: upsometro
        real(8), dimension(8,2), intent(inout) :: pos
        integer(4) :: idir, itemp, jtemp, i, j
        real(8) :: ai_out, aj_out

        write(13, 310) upsometro
310     format(F15.3)
        DO idir=1,8
            itemp = i_in
            jtemp = j_in
            DO !euresi simeiou
                call get_next_point(itemp, jtemp, idir, i, j) !pairnw
to epomeno simeio sti dieu8unsi pou psaxnw
                IF (him(i,j) > upsometro) THEN
                    ai_out = i_in + (i - i_in) / (him(i,j) - him(i_in,
j_in)) * (upsometro - him(i_in, j_in))
                    aj_out = j_in + (j_in - j) / (him(i,j) - him(i_in,
j_in)) * (upsometro - him(i_in, j_in))

```

```

        exit
    ELSE IF (him(i,j) == upsometro) THEN
        ai_out = i
        aj_out = j
        exit
    ELSE
        itemp = i
        jtemp = j
    END IF
END DO
! metatropi suntetagmenwn simeiou se metra kai dimiourgia
.syk file
ai_out = (ai_out - i_in) *90.0
aj_out = (aj_out - j_in) *90.0
pos(idir,1) = ai_out
pos(idir,2) = aj_out
write(13,320) ai_out, aj_out

END DO

write(13,330) "$"
330 format(a)
320 format(F15.3, F15.3)
end subroutine isoupsis

```

```

subroutine get_next_point(i_start, j_start, idir, i, j)
! epistrefei ti thesi(i,j) tou epomenou simeiou apo to simeio
(i_start, i_start) sti dieu8unsi idir
implicit none
integer(4), intent(in) :: i_start, j_start, idir
integer(4), intent(out) :: i, j

i = i_start
j = j_start
if (idir == 1) j = j - 1
if (idir == 2) then
    i = i + 1
    j = j - 1
end if
if (idir == 3) i = i + 1
if (idir == 4) then
    i = i + 1
    j = j + 1
end if
if (idir == 5) j = j + 1
if (idir == 6) then
    i = i - 1
    j = j + 1
end if
if (idir == 7) i = i - 1
if (idir == 8) then
    i = i - 1
    j = j - 1
end if

```

```

end subroutine get_next_point

real(8) function volume(i, j, amin, amax)
! vriskei ton oggo tou nerou pou uparxei sti limnodeksameni me
kentro to simeio me thesi (i,j), megisto upsometro amax kai elaxiston to
amin
    implicit none
    integer(4), intent(in) :: i, j
    real(8), intent(in) :: amax, amin
    real(8) :: b1, b2
    real(8), dimension(8,2) :: pos

    call isoupsis(i, j, amax, pos)
    b1 = emvadon(pos)
    volume = (amax - amin) / 3 * b1

end function volume

real(8) function emvadon(pos)
!vriskei to emvadon tw'n 8 simeiw'n me suntetagmenes apo8ikevmenes
ston pinaka pos
!i seira tw'n simeiw'n einai orologiaki
    implicit none
    real(8), dimension(8,2), intent(inout) :: pos
!
mexri 8 simeia
    integer(4) :: i, ipl
    real(8) :: amin, sum

    ipl=8
! doulevw me 8 simeia, an allaksei o kwdikas
to allazw eukola apo edw
!vriskw to min y
amin = 1000000.0
do i=1,ipl
    if (pos(i,2) < amin) amin = pos(i,2)
end do
! pros8etw se ola ta y to -min wste na einai ola >= 0 (gia na
doulevei o tupos)
do i=1,ipl
    pos(i,2) = pos(i,2) - amin
end do
!efarmozw ton tupo
sum = 0
do i=2,ipl
    sum = sum + (pos(i,1) - pos(i-1,1)) * 0.5 * (pos(i,2) +
pos(i-1,2))
end do
sum = sum + (pos(1,1) - pos(ipl,1)) * 0.5 * (pos(1,2) +
pos(ipl,2))
if (sum < 0 ) sum = -sum
emvadon = sum

```

```

end function emvadon

subroutine entopismos(i,j)
! entopizei sto xarti to simeio pou exei 8esi i,j ston pinaka sto
google earth
  implicit none
  integer(4), intent(in) :: i, j
  real(8) :: al, f

  open(unit=12, file='limnodeksamenes.kml', status='old',
access='append')

  write(12,301) "<Placemark>"
301   format(a)
  write(12,301) "<name>"
  write(12,302) 1
302   format(I2)
  write(12,301) "</name>"
  write(12,301) "<Point>"
  write(12,301) "<coordinates>"
  call l_f(i, j, al, f)
  write(12,303) al, ",", f, ",0"
303   format(F9.6, a, F9.6, a)
  write(12,301) "</coordinates>"
  write(12,301) "</Point>"
  write(12,301) "</Placemark>"

  write(12,301) "</Folder>"
  write(12,301) "</Document>"
  write(12,301) "</kml>"
  close(12)

end subroutine entopismos

subroutine upsometra(i_in, j_in, iper)
! dexetai ti 8esi tou kentrikoou simeiou (i_in, j_in) kai to
plithos twv perimetrwn pou ikanopoioun gurw apo auto (iper)
! tupwnei sto upsometra.syn tis gewgrafikes suntetagmenes kai to
upsometro ka8e simeiou pou aniksei se autas tis perimetous
! gurw apo to kentriko simeio

  implicit none
  integer(4), intent(in) :: i_in, j_in, iper
  integer(4) :: itemp, i, j
  real(8) :: al, f

  open(11,file='upsometra.syn')

  itemp = 0
  do j=(j_in - iper), (j_in + iper)

```

```

        do i=(i_in - iper), (i_in + iper)
            itemp = itemp + 1
            call l_f(i, j, al, f)
            write(11,101) itemp, al, f, him(i,j)
101      format(I10, F15.6, F15.6, F15.3)
        end do
    end do

    close(11)

    end subroutine upsometra

```

```
end module moddemusgs
```

```

!=====
====
!====GNU G95 SPECIFIC ROUTINES
!=====
====

```

```

subroutine openBmp (uBmpx, filnam, stat1, ierr)
implicit none
integer*4 uBmpx, ierr           ! ARGUMENTS
character*(*) filnam, stat1    ! ARGUMENTS
character*10 stat

```

```

stat = stat1
if (stat .eq. ' ') stat = 'unknown'

```

```
!-----GNU g77 compiler
```

```

open (uBmpx, file=filnam, status=stat, iostat=ierr, &
&form='unformatted', access='stream')      ! It behaves like direct
file of recl=1

```

```

return
end subroutine openBmp

```

```

!=====
====

```

```

subroutine closeBmp (uBmpx)
implicit none
integer*4 uBmpx           ! ARGUMENTS

```

```

close (uBmpx)
return
end subroutine closeBmp

```

```
program limnodeksameni3
  use moddemusgs
  implicit none
  integer(4) :: istop, idam

  call init

  istop = 5
  idam = 3
  call perimetro(1, 1, 12002,18003, istop, idam)

end program limnodeksameni3
```


ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Bamler R., 1999. The SRTM Mission: A World-Wide 30 m Resolution DEM from SAR Interferometry in 11 Days. Photogrammetric Week 99, Wichmann Verlag: 145-154.
- [2] Chalmers A., Davis, T. and Reinhard, E., 2002. Practical parallel rendering. AK Peters Publishing, Nantich Massachusetts.
- [3] Ioannidis Ch. and Vassilaki D.I., 2008. Combined use of Spaceborne Optical and SAR Data – Incompatible Data Sources or a Useful Procedure?. FIG Working Week 2008, June 14-18, Stockholm, Sweden.
- [4] Li Z., Zhu Q. and Gold C., 2005. Digital Terrain Modelling: Principles and Methodology, CRC Press, 323p.
- [5] Nikodym T., 2010. Ray Tracing Algorithm For Interactive Applications, Master thesis, Czech Technical University, Faculty of Electrical Engineering, Department of Computer Science and Engineering, Supervisor Ing. Vlastimil Havran Ph.D.
- [6] <http://www.povray.org/>: POV-Ray - The Persistence of Vision Raytracer.
- [7] <http://srtm.csi.cgiar.org/>: SRTM 90m Digital Elevation Data, version4