Τ Ε Χ Ν Ο Λ Ο Γ Ι Κ Ο  Ε Κ Π Α Ι Δ Ε Υ Τ Ι Κ Ο  Ι Δ Ρ Υ Μ Α  Σ Ε Ρ Ρ Ω Ν
Σ Χ Ο Λ Η  Τ Ε Χ Ν Ο Λ Ο Γ Ι Κ Ω Ν  Ε Φ Α Ρ Μ Ο Γ Ω Ν
Τ Μ Η Μ Α  Π Λ Η Ρ Ο Φ Ο Ρ Ι Κ Η Σ  &  Ε Π Ι Κ Ο Ι Ν Ω Ν Ι Ω Ν

# ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ ΟΠΤΙΚΟΠΟΙΗΣΗ ΤΟΥ ΤΡΟΠΟΥ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ ΜΕ ΑΡΙΘΜΗΤΙΚΕΣ ΜΕΘΟΔΟΥΣ ΓΙΑ ΤΟ ΕΡΓΑΣΤΗΡΙΟ ΤΟΥ ΜΑΘΗΜΑΤΟΣ ΑΡΙΘΜΗΤΙΚΕΣ ΜΕΘΟΔΟΙ ΣΕ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

Πτυχιακή Εργασία του

Δριγγόπουλου Ευάγγελου (496)

Επιβλέπων: Βαρσάμης Δημήτριος

Σέρρες, Ιούνιος 2009

## Πίνακας Περιεχομένων

**Εισαγωγή**……4

**Κεφάλαιο 1**

Αριθμητική ανάλυση……..

1.1 Με τι ασχολείται η αριθμητική ανάλυση;……..... 5

1.2 Τι προβλήματα μπορούμε να λύσουμε με τη χρήση αριθμητικών μεθόδων…….. 7

1.3 Οι αλγόριθμοι στην αριθμητική ανάλυση……..... 9

**Κεφάλαιο 2**

Επίλυση μη γραμμικών εξισώσεων……...……..

2.1 Ποιες είναι οι μη γραμμικές εξισώσεις; 10

2.2 Μέθοδος Newton-Raphson…….. 10

2.2.1 Παράδειγμα της μεθόδου Newton-Raphson…… 11

2.3 Μέθοδος Τέμνουσας……..... 15

2.3.1 Παράδειγμα της μεθόδου Τέμνουσας…….. 16

2.4 Μέθοδος Διχοτόμησης……. 19

2.4.1 Παράδειγμα της μεθόδου Διχοτόμησης……... 20

2.5 Μέθοδος Regula falsi……..... 24

2.5.1 Παράδειγμα της μεθόδου Regula falsi……..... 25

**Κεφάλαιο 3**

Αριθμητική Ολοκλήρωση……..... 30

3.1 Κανόνας του τραπεζίου……..... 31

3.1.1 Παράδειγμα του κανόνα τραπεζίου……..... 33

3.2 Μέθοδος Simpson…… 36

3.2.1 Παράδειγμα της μεθόδου Simpson……..... 37

**Κεφάλαιο 4**

Πολυωνυμική Παρεμβολή

4.1 Πολυώνυμο παρεμβολής LaGrange……..... 40

4.1.1 Παράδειγμα πολυώνυμου παρεμβολής τύπου LaGrange……. 41

4.2 Μέθοδος Ελαχίστων Τετραγώνων…… 44

4.2.1 Παράδειγμα υπολογισμού πολυωνύμου με τη χρήση της μεθόδου των Ελαχίστων…… Τετραγώνων…….. 44

Παράρτημα ……..... 48

Μέθοδος Muller…….. 54

Κώδικας……..... 57

Βιβλιογραφία…… 104

# Εισαγωγή

---

**Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη** εκπαιδευτικής εφαρμογής σε γραφικό περιβάλλον Windows, στην οποία θα απεικονίζονται με γραφικό τρόπο τα βήματα που ακολουθούνται για την επίλυση προβλημάτων με αριθμητικές μεθόδους.

Τα προβλήματα που θα αντιμετωπιστούν είναι

- Επίλυση μη-γραμμικών εξισώσεων (με τις μεθόδους Newton-Raphson, Διχοτόμησης, Regula-Falsi, Τέμνουσας)
- Αριθμητική Ολοκλήρωση (με τις μεθόδους Τραπεζίου και Simpson)
- Πολυωνυμική Παρεμβολή (με τις μεθόδους LaGrange και την Ευθεία Ελαχίστων Τετραγώνων)

Οι αριθμητικές μέθοδοι που χρησιμοποιούνται στην εργασία είναι ομαδοποιημένες σε κεφάλαια, το καθένα από τα οποία ασχολείται με διαφορετικά προβλήματα επίλυσης ή προσέγγισης εξισώσεων. Για κάθε μια από τις μεθόδους γίνεται μια σύντομη θεωρητική παρουσίαση, και παρέχονται παραδείγματα τόσο σε επίπεδο μαθηματικών, αλλά και παραδείγματα με τη χρήση της εφαρμογής που αναπτύχθηκε σε γλώσσα προγραμματισμού C++.

# Κεφάλαιο 1

# Αριθμητική ανάλυση

## 1.1. Με τι ασχολείται η αριθμητική ανάλυση;

Μέχρι τώρα γνωρίζαμε τον συγκεκριμένο τρόπο λύσης διαφόρων γνωστών τύπων εξισώσεων. Τέτοιες ήταν οι απλές, πρωτοβάθμιες εξισώσεις, ή οι δευτεροβάθμιές. Γνωρίζαμε λοιπόν ότι, για την επίλυση της δευτεροβάθμιας εξίσωσης έπρεπε απλά να υπολογίσουμε τη διακρίνουσα και στη συνέχεια, ανάλογα με το πρόσημο και την τιμή της να υπολογίσουμε τις ρίζες της που μπορεί να είναι είτε πραγματικές είτε φανταστικές. Ξέραμε επίσης ότι για άλλες, ανώτερου βαθμού, αλγεβρικές εξισώσεις δεν υπήρχαν κάποιες συγκεκριμένες γνωστές τεχνικές επίλυσης αλλά προσπαθούσαμε να επιλύσουμε μερικές συγκεκριμένες μορφές τους. Για παράδειγμα, μπορούσαμε να λύσουμε, με συστηματικό τρόπο, μία τεταρτοβάθμια εξίσωση μόνο στην περίπτωση που περιείχε μόνο όρους υψωμένους σε άρτια δύναμη (διτετράγωνη) ανάγοντας την σε μία δευτεροβάθμια. Σε άλλες περιπτώσεις προσπαθούσαμε να υπολογίσουμε κάποιες λύσεις των εξισώσεων ώστε με μεθόδους παραγοντοποίησης, όπως το σχήμα Horner, να τις φέρουμε σε κάποια μορφή που θα μπορούσαμε να εύκολα να επιλύσουμε.

Σε όλες τις παραπάνω περιπτώσεις αναφερθήκαμε σε συγκεκριμένες εξισώσεις (δευτεροβάθμια, πρωτοβάθμια, κλπ) ή σε εξισώσεις που γνωρίζουμε κάποια λύση τους και μπορούμε να κάνουμε παραγοντοποίηση (σχήμα Horner). Στην πράξη όμως τα πράγματα δεν είναι τόσο απλά. Οι εξισώσεις που μπορεί να συναντήσουμε κατά την επίλυση ενός προβλήματος, συνήθως, δεν μπορούν να αναχθούν σε κάποια από τις επιλύσιμες μορφές, ούτε μπορούν να παραγοντοποιηθούν με απλό τρόπο γιατί δεν γνωρίζουμε κάποια ρίζα τους. Οι περιπτώσεις αυτές (δηλ. εξισώσεις μη επιλύσιμες με αναλυτικές μεθόδους) δεν είναι κάποιες εξηζητημένες μορφές αλλά, αντίθετα, είναι οι πιο συχνά χρησιμοποιούμενες. Ετσι, για μία εξίσωση, για παράδειγμα, που είναι πέμπτου βαθμού, Εξ.(1.1) δεν υπάρχει κάποιος αναλυτικός τρόπος λύσης.

$$P(x) = x^5 + x + 1 = 0 \qquad (1.1)$$

Το σημαντικό όμως, στις περιπτώσεις αυτές, είναι ότι υπάρχουν θεωρήματα τα οποία αποδεικνύουν την ύπαρξη ριζών μέσα σε κάποια πεπερασμένα διαστήματα. Έτσι, επειδή η

Εξ.(1.1), είναι συνεχής σε όλο το R, και επειδή $P(-1) = -1 < 0$ και $P(1) = 3 > 0$ εφαρμόζουμε το θεώρημα Bolzano και συμπεραίνουμε ότι στο ανοικτό διάστημα (-1, 1) υπάρχει τουλάχιστον μία πραγματική ρίζα του πολυωνύμου. Γνωρίζοντας το διάστημα μέσα στο οποίο βρίσκεται τουλάχιστον μία από τις πραγματικές ρίζες του πολυωνύμου, χρησιμοποιούμε την αριθμητική ανάλυση για να την υπολογίσουμε.

Στο σημείο αυτό θα πρέπει να τονιστούν κάποιες πολύ σημαντικές διαφορές που παρουσιάζει η αριθμητική με την αναλυτική επίλυση των εξισώσεων. Οι αριθμητικές μέθοδοι στηρίζονται πάνω σε υπολογισμούς, πράξεις με πεπερασμένη ακρίβεια. Έτσι τα αποτελέσματα τα οποία προκύπτουν δεν είναι ποτέ απολύτως ακριβή αλλά μας οδηγούν πάρα πολύ κοντά στην περιοχή που πραγματικά βρίσκονται οι ρίζες των εξισώσεων και ανάλογα με την ακρίβεια υπολογισμού που απαιτούμε, πλησιάζουμε όσο το δυνατόν περισσότερο στις πραγματικές τιμές των ριζών αυτών. Όμως η όσο το δυνατόν κοντινότερη προσέγγιση έχει κόστος σε χρόνο, αφού οι μεγαλύτερης ακρίβειας υπολογισμοί απαιτούν περισσότερες πράξεις, αλλά και σε χρήμα αφού τα μηχανήματα τα οποία μπορούν να κάνουν υπολογισμούς ταχύτερα αλλά και με μεγαλύτερη ακρίβεια κοστίζουν.

Ένα ακόμα μειονέκτημα που παρουσιάζει η χρήση αριθμητικών μεθόδων είναι ότι πολύ συχνά πρέπει να γίνουν δοκιμές με περισσότερες από μία μεθόδους μέχρι να βρεθεί η κατάλληλη η οποία θα επιτρέπει την επίλυση των συγκεκριμένων εξισώσεων που μας ενδιαφέρουν. Πρέπει επίσης, πριν τη χρήση των αριθμητικών μεθόδων, να χρησιμοποιηθούν οι κατάλληλες αναλυτικές μέσω των οποίων θα εντοπιστούν οι περιοχές μέσα που βρίσκονται οι ρίζες. Τίθεται λοιπόν το ερώτημα: που υπερτερούν οι αριθμητικές μέθοδοι από τις αναλυτικές; Η απάντηση είναι απλή: στο ότι με την επιλογή των κατάλληλων αριθμητικών μεθόδων μπορούμε να επιλύσουμε πολύ μεγάλο αριθμό αριθμητικών προβλημάτων χωρίς να χρειάζεται να έχουν κάποια συγκεκριμένη μορφή. Ο τρόπος και η ακρίβεια με την οποία αυτά θα επιλυθούν, εξαρτώνται από την μέθοδο που θα χρειαστεί, από τις δυνατότητες των μηχανημάτων αλλά κυρίως από τον άνθρωπο ο οποίος θα επιλέξει τον τρόπο και τη μέθοδο που πρέπει κάθε φορά να χρησιμοποιήσει. Από τα παραπάνω συμπεραίνουμε ότι οι αριθμητικές μέθοδοι δεν μπορούν να δώσουν κανένα αποτέλεσμα αν δεν χρησιμοποιήσουν συγκεκριμένους αναλυτικούς υπολογισμούς και αποτελέσματα. Για το λόγο αυτό στις σημειώσεις αυτές γίνεται μια προσπάθεια να μελετηθούν οι αριθμητικές μέθοδοι τόσο από την πλευρά της θεωρίας (χρήση αναλυτικών εργαλείων και θεωρημάτων), όσο και από αυτή της υλοποίησης (δημιουργία αλγορίθμων).

## 1.2. Τι προβλήματα μπορούμε να λύσουμε με τη χρήση αριθμητικών μεθόδων;

Προηγουμένως αναφερθήκαμε στις δυνατότητες επίλυσης απλών πολυωνυμικών εξισώσεων με τη χρήση αριθμητικών μεθόδων. Η αναφορά στα προβλήματα αυτά έγινε για να μπορέσει ο αναγνώστης να κατανοήσει ευκολότερα το πόσο επιβεβλημένη είναι η χρήση αναλυτικών μεθόδων για τη λύση τους. Όμως με τις αναλυτικές μεθόδους μπορούμε να επιλύσουμε και πολλές άλλες μορφές προβλημάτων τα οποία θα ήταν αδύνατο να αντιμετωπιστούν με αναλυτικές μεθόδους. Στη συνέχεια λοιπόν δίνουμε ορισμένα παραδείγματα μαθηματικών προβλημάτων τα οποία επιλύονται με τη χρήση αριθμητικών μεθόδων.

**Α) Επίλυση γραμμικών συστημάτων με σταθερούς συντελεστές**

Ας θεωρήσουμε ένα γραμμικό σύστημα Ν εξισώσεων με Ν αγνώστους της μορφής:

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1N}x_N = b_1$$

$$a_{21}x_1 + a_{22}x_2 + ... + a_{2N}x_N = b_2$$

$$a_{N1}x_1 + a_{N2}x_2 + ... + a_{NN}x_N = b_N$$

Για ένα τέτοιο σύστημα υπάρχουν αρκετές αναλυτικές μέθοδοι οι οποίες επιτρέπουν την επίλυση του. Όμως, στην περίπτωση που το Ν είναι πολύ μεγάλο η επίλυση του με χρήση των γνωστών αναλυτικών μεθόδων είναι πρακτικά αδύνατη. Στο πρόβλημα αυτό δίνει λύση η αριθμητική ανάλυση. Με τη χρήση των κατάλληλων αλγορίθμων μπορούμε να επιλύσουμε προβλήματα τα οποία θεωρητικά μπορούν να επιλυθούν αλλά πρακτικά κάτι τέτοιο θα ήταν αδύνατο.

**Β) Επίλυση και υπολογισμός τιμής ορισμών ολοκληρωμάτων**

Είναι γνωστό ότι η πλειοψηφία των ορισμένων ολοκληρωμάτων που μπορεί να συναντήσουμε σε διάφορα προβλήματα δεν μπορούν να επιλυθούν αναλυτικά. Με τη χρήση όμως αριθμητικών μεθόδων ο υπολογισμός της τιμής ορισμένων ολοκληρωμάτων μπορεί να υπολογιστεί. Ετσι το ολοκλήρωμα της Εξ.(1.2), δεν μπορεί να υπολογιστεί με αναλυτικές αλλά μόνο με τη χρήση των κατάλληλων αριθμητικών μεθόδων.

$$\int_0^{\pi} \sin(x^2)dx \qquad (1.2)$$

Ο υπολογισμός της τιμής ορισμένων ολοκληρωμάτων με χρήση αναλυτικών μεθόδων, είναι μια διαδικασία αρκετά επίπονη και πολλές φορές αναποτελεσματική. Για το λόγο αυτό τα περισσότερα προβλήματα τα οποία εμπεριέχουν και τον υπολογισμό ορισμένων ολοκληρωμάτων επιλύονται με αριθμητικές μεθόδους. Η ακρίβεια των αποτελεσμάτων στην περίπτωση αυτή, εξαρτάται από τη μέθοδο που χρησιμοποιούμε καθώς και την ακρίβεια του υπολογιστή μας.

**Γ) Επίλυση διαφορικών εξισώσεων**

Αρκετές διαφορικές εξισώσεις έχουν λύσεις οι οποίες μπορούν να υπολογιστούν με αναλυτικές μεθόδους. Για παράδειγμα, η ακόλουθη συνήθης διαφορική εξίσωση (Εξ.(1.3)),

$$y''(x) + y(x) = 0 \qquad (1.3)$$

έχει τη γνωστή γενική λύση

$$y = y_0 \sin(x - x_0) \qquad (1.4)$$

όπου οι σταθερές $y_0$ και $x_0$ προσδιορίζονται από τις αρχικές συνθήκες. Υπάρχουν όμως κάποιες άλλες διαφορικές εξισώσεις των οποίων η λύση είτε είναι πολύ δύσκολο, είτε αδύνατο να υπολογιστεί με χρήση αναλυτικών μεθόδων. Ένα τέτοιο παράδειγμα είναι η διαφορική εξίσωση της Εξ.(1.5),

$$y'' + \sin y = 0 \qquad (1.5)$$

Η Εξ.(1.5) είναι πολύ δυσκολότερο να λυθεί από την Εξ.(1.4) και για αυτό το λόγο η μελέτη της γίνεται με τη χρήση αριθμητικών μεθόδων. Βέβαια δεν είναι δυνατόν καμία αριθμητική μέθοδος να «υπολογίσει» τον τύπο της λύσης της εξίσωσης (όπως γίνεται αναλυτικά με την Εξ.(1.4) που είναι αναλυτική λύση της Εξ.(1.3)) αλλά είναι δυνατόν να υπολογιστούν αρκετά ζεύγη τιμών (x,y) που να ικανοποιούν τη συνάρτηση y(x), οπότε, γνωρίζοντας τα μπορούμε να τη μελετήσουμε τη συμπεριφορά της (ακρότατα, μονοτονία, κλπ) με πολύ μεγάλη ακρίβεια, έστω και αν δεν γνωρίζουμε την αναλυτική της λύση.

**Δ) Επίλυση ειδικών συναρτήσεων**

Ως παραδείγματα ειδικών συναρτήσεων θα μπορούσαμε να αναφέρουμε είτε,

1) κάποιες τριγωνομετρικές συναρτήσεις, π.χ. $\sin(x) = 0.4$ , είτε 2) κάποιες συναρτήσεις που περιέχουν ρίζες, π.χ. $\sqrt[3]{x} + \sqrt{x} - 5 = 0$ , είτε οποιαδήποτε άλλη εξίσωση της οποίας οι ρίζες δεν είναι δυνατόν να υπολογιστούν με κάποια απλή αναλυτική μέθοδο.

## 1.3. Οι αλγόριθμοι στην αριθμητική ανάλυση

Ένα σημαντικό στοιχείο κάθε αριθμητικής μεθόδου είναι το γεγονός ότι οι υπολογισμοί γίνονται με τη χρήση υπολογιστών. Αυτό έχει σαν αποτέλεσμα ότι μπορούμε να επιτύχουμε μεγάλες ταχύτητες υπολογισμών αλλά και ότι τα αποτελέσματα μας δε θα περιέχουν λάθη. Για να μπορεί όμως να χρησιμοποιηθεί ο υπολογιστής για τον υπολογισμό των ποσοτήτων που μας ενδιαφέρουν πρέπει να προγραμματιστεί κατάλληλα. Ο προγραμματισμός αυτός γίνεται με τη δημιουργία των κατάλληλων αλγορίθμων οι οποίοι στη συνέχεια μεταφράζονται σε κώδικα της γλώσσας προγραμματισμού που θέλουμε να χρησιμοποιήσουμε.

# ΚΕΦΑΛΑΙΟ 2

# Επίλυση μη-γραμμικών εξισώσεων

## 2.1 Ποιες είναι οι μη γραμμικές εξισώσεις;

Με τον όρο μη γραμμικές εξισώσεις εννοούμε εξισώσεις τις μορφής

$$f(x) = 0$$

που προέρχονται από συναρτήσεις f(x) που είναι μη γραμμικές ως προς x. Περιέχουν δηλαδή και όρους ως προς x βαθμού διάφορου του πρώτου. Τέτοιες εξισώσεις μπορεί να είναι:

$$f(x) = x^5 + x + 1 = 0 \qquad (2.1)$$

$$f(x) = \sin(x^2) - x^2 + x + 1$$

Είναι φανερό ότι οι παραπάνω εξισώσεις δεν μπορούν να επιλυθούν με τις γνωστές αναλυτικές μεθόδους. Για να λυθούν λοιπόν θα πρέπει να χρησιμοποιηθούν αριθμητικές μέθοδοι. Ο υπολογισμός των ριζών τέτοιων εξισώσεων με χρήση αριθμητικών μεθόδων στηρίζεται σε κάποια θεωρήματα. Έτσι, αποδεικνύεται ότι π.χ. η Εξ.(2.1) έχει τουλάχιστον μία πραγματική ρίζα μέσα στο ανοιχτό διάστημα (-1,1).

## 2.2 Μέθοδος Newton-Raphson

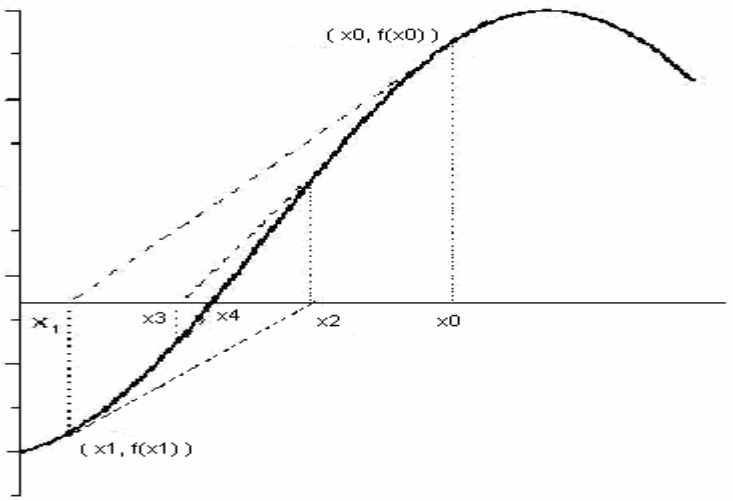
Η μέθοδος Newton-Raphson χρησιμοποιείται για την προσεγγιστική λύση εξισώσεων. Για να βρεθεί μια ακριβής τιμή της ρίζας απαιτείται ο ορισμός μιας αρχικής τιμής και ο υπολογισμός της παραγώγου της συνάρτησης σε κάθε διαδοχική προσέγγιση. Ο γενικός τύπος της Newton-Raphson για τη διαδοχική προσέγγιση της τελικής ρίζας είναι:

$$x_{i+1} = \frac{x_i - f(x_i)}{f'(x_i)} \quad , \; i = 0,1,2....$$

όπου $x_{i+1}$ είναι η τελική ρίζα, $x_i$ η ρίζα που βρέθηκε με την προηγούμενη προσέγγιση και $f'(x_i)$ η παράγωγος της συνάρτησης με την τιμή της προηγούμενης ρίζας.

Η διαδικασία σταματάει όταν δύο διαδοχικές προσεγγίσεις είναι πολύ κοντά, δηλαδή όταν $|x_{n+1}, x_n| \leq \varepsilon$, όπου $x_{n+1}$ και $x_n$ δύο διαδοχικές προσεγγίσεις της ρίζας και $\varepsilon$ η επιθυμητή ακρίβεια, και/ή όταν ξεπεραστεί ένα ορισμένο πλήθος επαναλήψεων της μεθόδου.

Γεωμετρικά η μέθοδος Newton δίνεται από το παρακάτω σχήμα




Εικόνα 2.1

### 2.2.1 Παράδειγμα της μεθόδου *Newton-Raphson*

Παράδειγμα χρήσης της μεθόδου *Newton-Raphson για την εύρεση της ρίζας της εξίσωσης:* $f(x) = x^2 - 3$ *, στο διάστημα (-2,1) με ακρίβεια 4 δεκαδικών ψηφίων,* $x_0 = 1$ *και αριθμό επαναλήψεων* $n = 5$. *Τότε* $f(x) = 2x$

*1η επανάληψη : αντικαθιστώντας στην αναδρομική σχέση* $x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$ *, προκύπτει* $x_1 = 2$

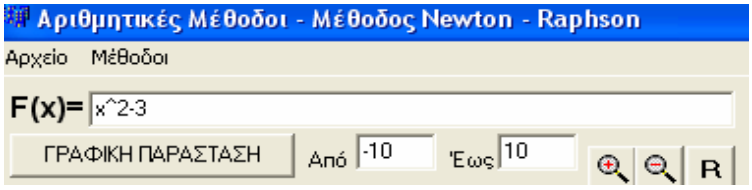
*2η επανάληψη:* $x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)} \Rightarrow x_2 = 1{,}75$

*3η επανάληψη:* $x_3 = 1{,}7321$

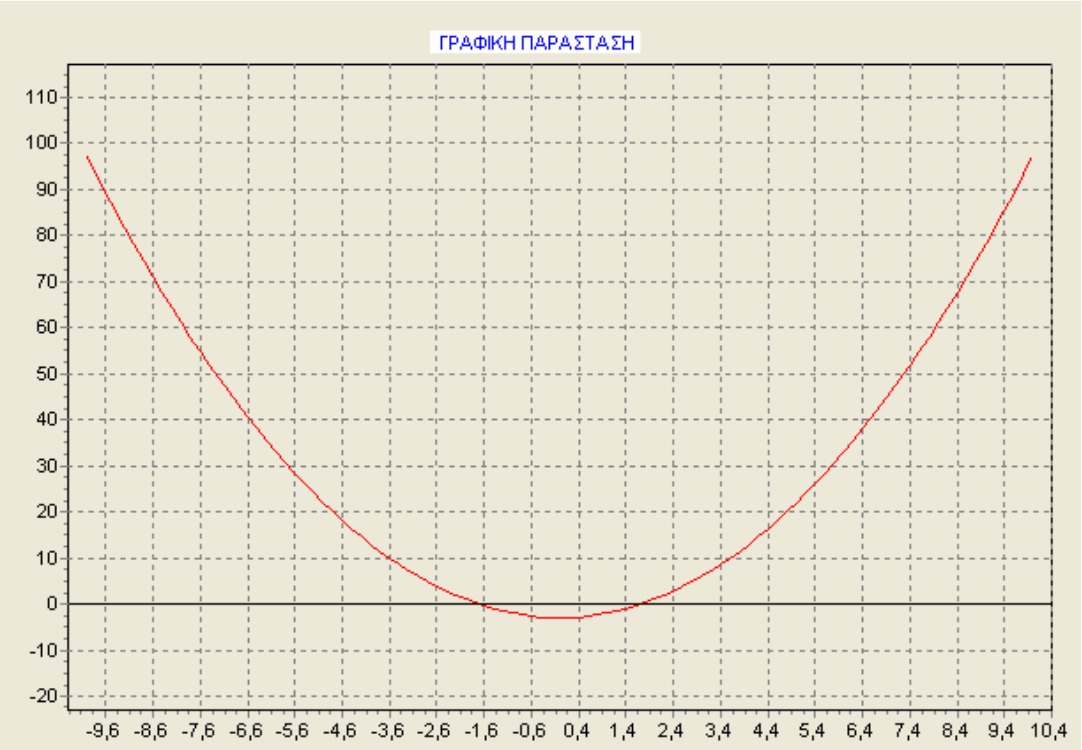
*4η επανάληψη:* $x_4 = 1{,}7320$

*Η ρίζα βρέθηκε πριν ολοκληρωθούν οι επαναλήψεις γιατί ικανοποιήθηκε το κριτήριο της ακρίβειας.*

Παρακάτω φαίνεται η λύση της εξίσωσης στην εφαρμογή που αναπτύχθηκε με τη γλώσσα C++. Αρχικά πληκτρολογείται η εξίσωση προς λύση, ώστε να εμφανιστεί η γραφική της παράσταση (Εικ 2.3) εντός των ορίων που έχει ορίσει ο χρήστης (Εικ 2.2)



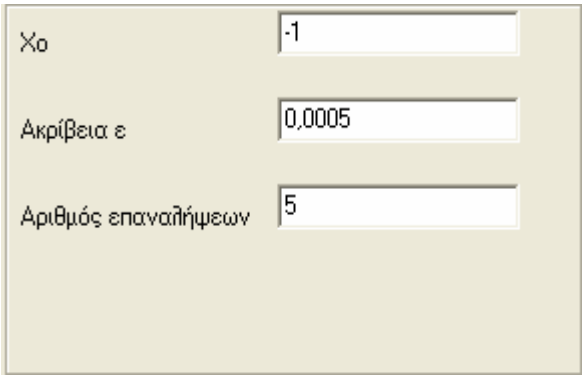

Εικόνα 2.2: Εισαγωγή συνάρτησης και ορίων της γραφικής της παράστασης



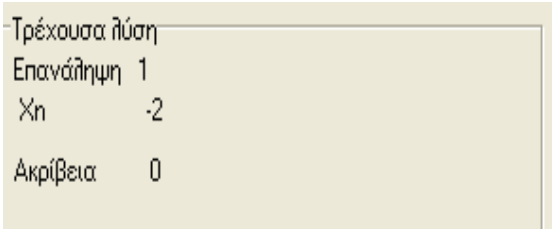

Εικόνα 2.3: Γραφική παράσταση της συνάρτησης προς λύση

Στη συνέχεια, ο χρήστης πληκτρολογεί την αρχική τιμή $x_0 = -1$, την επιθυμητή ακρίβεια $\varepsilon = 0{,}0005$ και τον αριθμό επαναλήψεων της εκτέλεσης της μεθόδου n =5 (Εικ. 2.4). Με το πάτημα του κουμπιού "ΕΠΟΜΕΝΟ ΒΗΜΑ" η γραφική παράσταση εμφανίζεται μεγενθυμένη γύρω από την περιοχή όπου η παράγωγος της συνάρτησης τέμνει την συνάρτηση (Εικ. 2.6), εμφανίζοντας παράλληλα και τις συντεταγμένες του σημείου τομής τους και του σημείου που

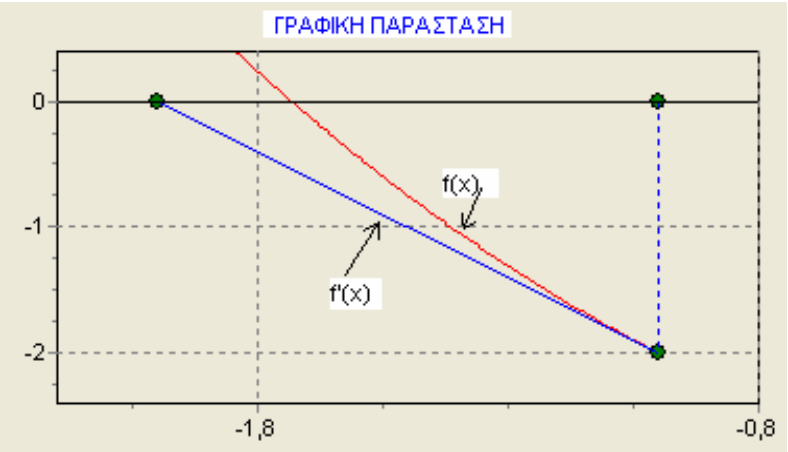
τέμνει τον οριζόντιο άξονα . Επίσης στην καρτέλα “τρέχουσα λύση” φαίνεται ο αριθμός της επανάληψης, η τιμή της τρέχουσας λύσης και η ακρίβεια με την οποία βρέθηκε (Εικ. 2.5)




Εικόνα 2.4: Εισαγωγή παραμέτρων της μεθόδου

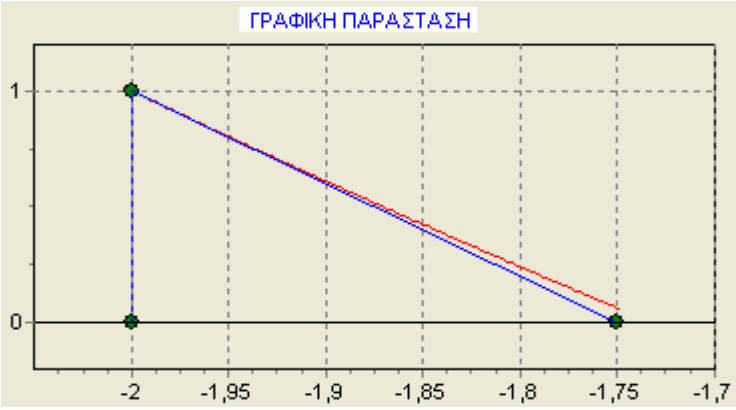



Εικόνα 2.5: Τρέχουσα λύση σε κάθε επανάληψη της μεθόδου




Εικόνα 2.6: Η γραφική παράσταση γύρω από το σημείο όπου η παράγωγος της συνάρτησης τέμνει την συνάρτηση

Πατώντας το κουμπί "επόμενο βήμα" η προσέγγιση της ρίζας θα επαναληφθεί με αρχική τιμή την τιμή της ρίζας που βρέθηκε στην προηγούμενη επανάληψη και τα αποτελέσματά της θα εμφανιστούν με τον ίδιο τρόπο (Εικ. 2.7, 2.8)




Εικόνα 2.7

Τρέχουσα λύση
Επανάληψη 2
Xn -1,75
Ακρίβεια 0,25

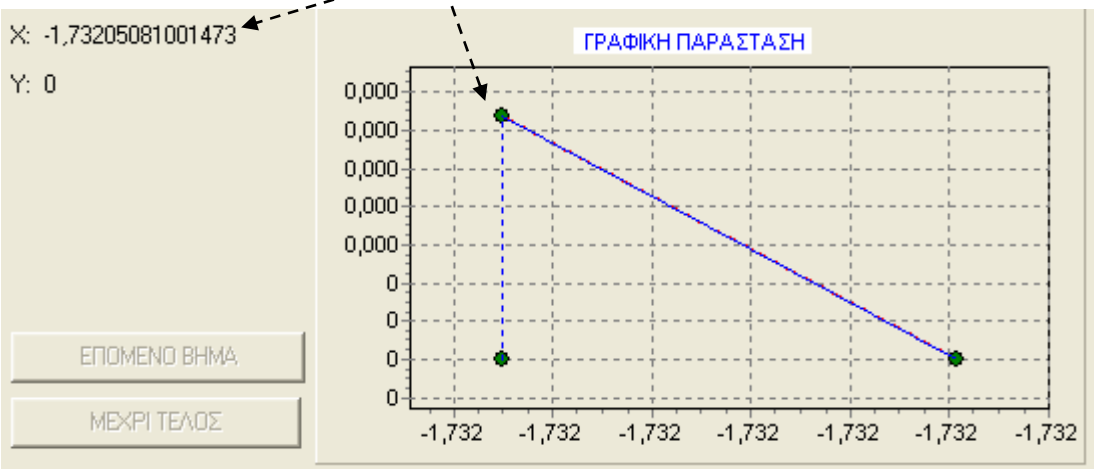
Εικόνα 2.8

Επαναλαμβάνοντας τη διαδικασία για 4 φορές η εκτέλεση της μεθόδου τερματίζεται εμφανίζοντας τα τελικά αποτελέσματα της προσέγγισης της ρίζας (Εικ. 2.9). Παρατηρείται ότι η γραφική παράσταση της παραγώγου ταυτίζεται με αυτήν της αρχικής εξίσωσης (Εικ. 2.10):

Τελική λύση
Επανάληψη 4
Ρίζα εξίσωσης -1,73205081001473
Ακρίβεια 9,20471281324886E-5

Εικόνα 2.9: Τελικά αποτελέσματα της εφαρμογής της μεθόδου

(Οι συντεταγμένες της τελικής ρίζας εμφανίζονται στη φόρμα της εφαρμογής επιλέγοντας το σημείο επάνω στη γραφική παράσταση)



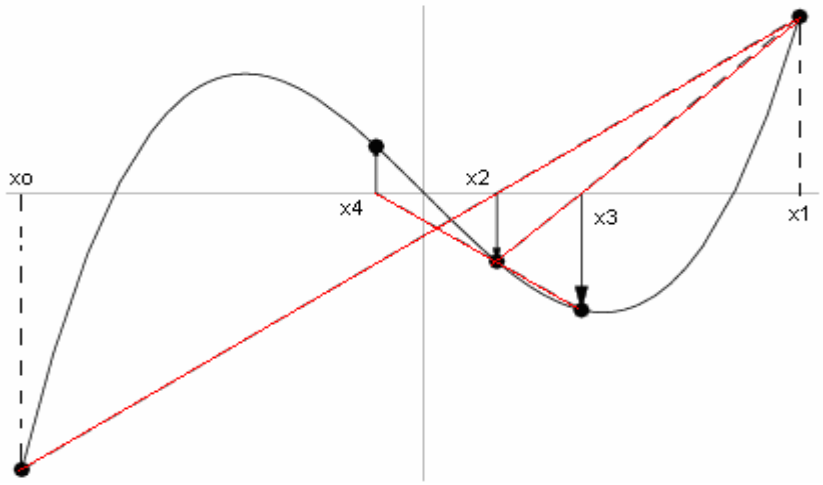

Εικόνα 2.10

## 2.3 ΜΕΘΟΔΟΣ ΤΕΜΝΟΥΣΑΣ

Για την εύρεση μιας ρίζας($x_{k+1}$) μιας εξίσωσης, η μέθοδος της τέμνουσας χρησιμοποιεί τις δύο προηγούμενες προσεγγίσεις ($x_k$ και $x_{k-1}$) της ρίζας. Σε κάθε επανάληψη κ, δημιουργείται η ευθεία που διέρχεται από τα σημεία ($x_{k-1}$, $f(x_{k-1})$),($x_k$,$f(x_k)$) και υπολογίζεται ως νέα ρίζα η $x_{k+1}$, δηλαδή το σημείο τομής της ευθείας αυτής με τον άξονα x :

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$$

Γεωμετρικά η μέθοδος της τέμνουσας δίνεται από το παρακάτω σχήμα




Εικόνα 2.11

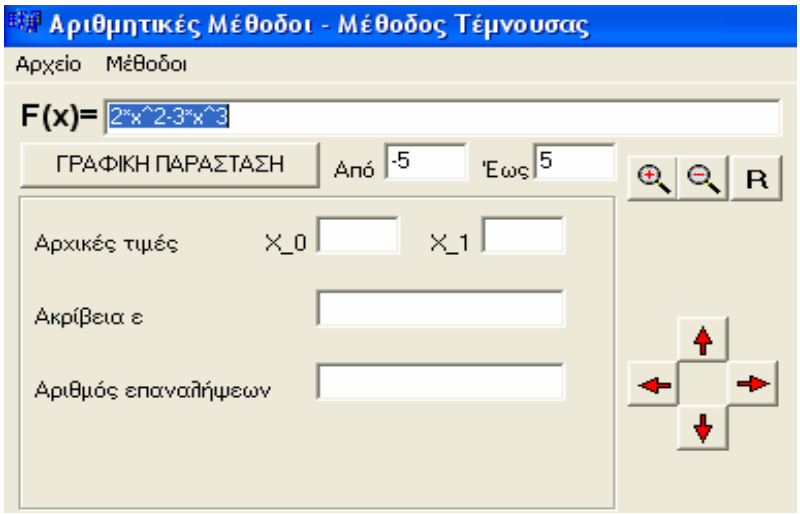
Η μέθοδος τερματίζεται είτε όταν τηρείται ένα κριτήριο τερματισμού (π.χ. εάν η ακρίβεια της ρίζας που βρέθηκε είναι μικρότερη από μια δεδομένη ακρίβεια) είτε όταν δε βρεθεί η κατάλληλη ρίζα μετά από ορισμένο αριθμό επαναλήψεων.

### 2.3.1 Παράδειγμα της μεθόδου *τέμνουσας*

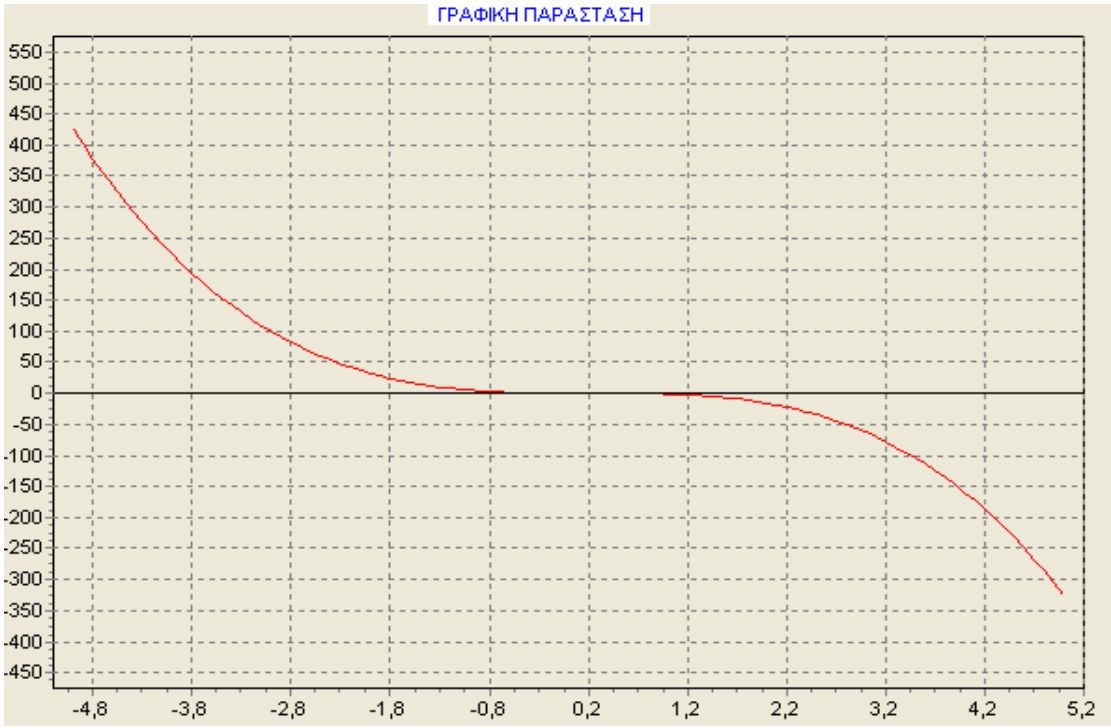
Για παράδειγμα, η συνάρτηση $f(x) = 2x^2 - 3x^3$ με αρχικές τιμές $x_0 = -3$ και $x_1 = 3$, και δεδομένης ακρίβειας 5 δεκαδικών ψηφίων, εκτελείται ως εξής:

*1η επανάληψη*: $x_2 = x_1 - \dfrac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_1)$ Με αντικατάσταση στην εξίσωση των τιμών -3 και 3 (όπου $x_0$ και $x_1$ αντίστοιχα), προκύπτει ότι η ρίζα της συνάρτησης είναι 0,66666

Παρακάτω φαίνεται η λύση της εξίσωσης στην εφαρμογή που αναπτύχθηκε με τη γλώσσα C++. Αρχικά πληκτρολογείται η εξίσωση προς λύση (Εικ. 2.12), ώστε να εμφανιστεί η γραφική της παράσταση εντός των ορίων που έχει ορίσει ο χρήστης (Εικ. 2.13):

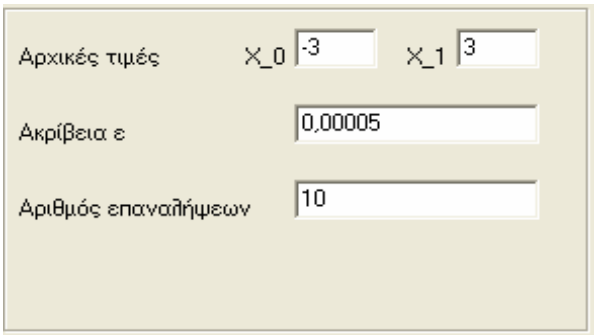



Εικόνα 2.12: Εισαγωγή συνάρτησης και ορίων της γραφικής της παράστασης




Εικόνα 2.13: Γραφική παράσταση της συνάρτησης προς λύση

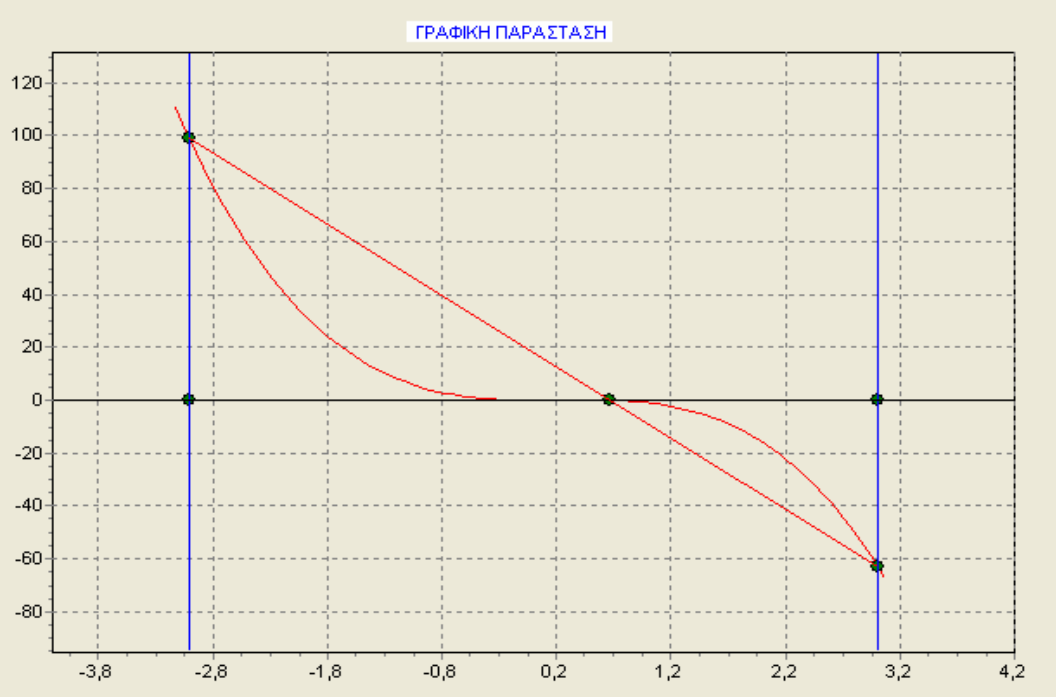
Στη συνέχεια, ο χρήστης πληκτρολογεί τις αρχικές τιμές (-3, 3) , την επιθυμητή ακρίβεια και τον αριθμό των επαναλήψεων της εκτέλεσης της μεθόδου (Εικ. 2.14), με αποτέλεσμα η γραφική παράσταση να περιοριστεί εντός των ορίων (Εικ. 2.16), εμφανίζοντας παράλληλα και τις συντεταγμένες των ορίων, των σημείων που τέμνουν τον οριζόντιο άξονα και την τρέχουσα λύση στην επανάληψη όπου βρισκόμαστε (Εικ. 2.15, 2.17).




Εικόνα 2.14: Εισαγωγή παραμέτρων της μεθόδου

Τρέχουσα λύση
Επανάληψη 1
Xn 0,666666666666667
Ακρίβεια 0

Εικόνα 2.15: Τρέχουσα λύση σε κάθε επανάληψη της μεθόδου

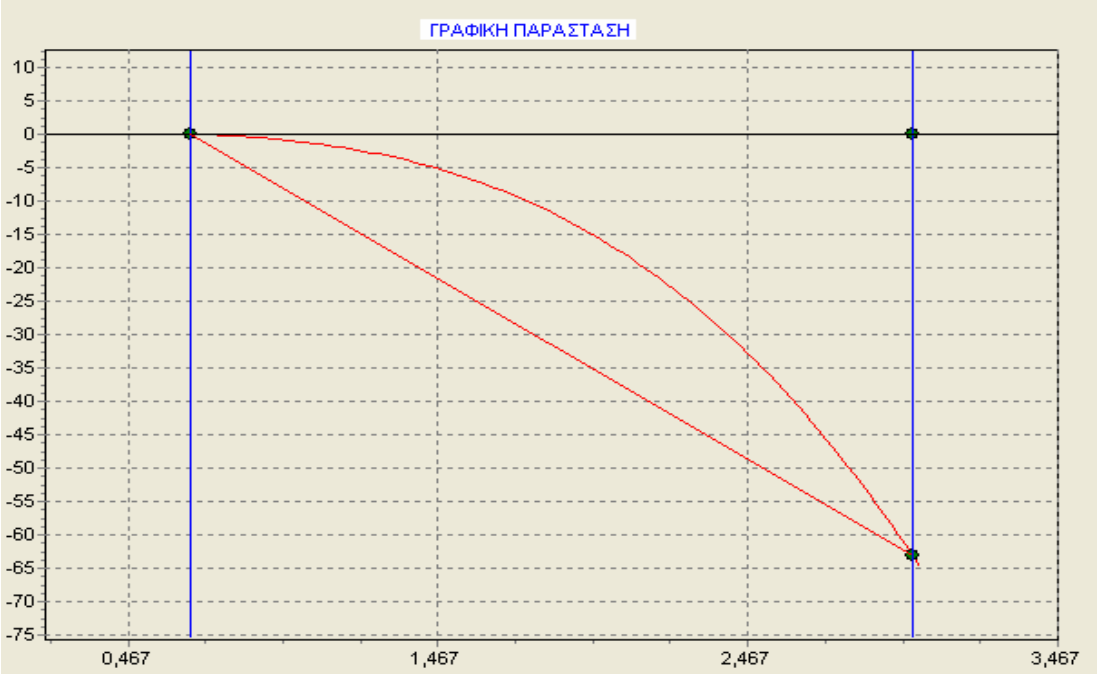



Εικόνα 2.16*:* Η γραφική παράσταση γύρω από το σημείο όπου η παράγωγος της συνάρτησης τέμνει την συνάρτηση

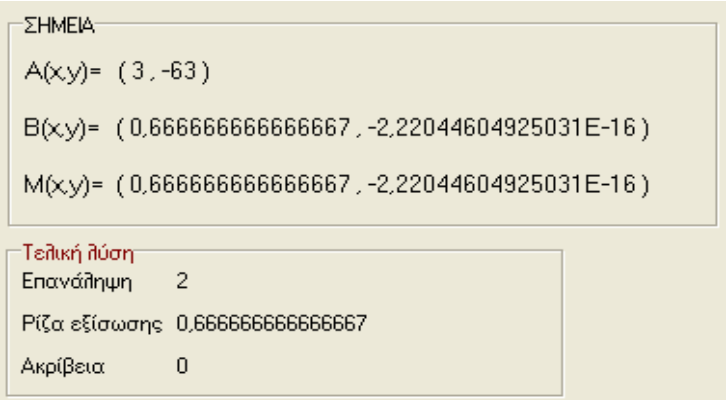
ΣΗΜΕΙΑ
A(x,y)= (-3, 99)
B(x,y)= (3, -63)
M(x,y)= (0,666666666666667, -2,22044604925031E-16)

Εικόνα 2.17: Οι συντεταγμένες των ορίων της γραφικής παράστασης και του σημείου που προσεγγίζει τον οριζόντιο άξονα

Πατώντας το κουμπί "επόμενο βήμα" σχεδιάζεται εκ νέου η γραφική παράσταση εντός των νέων ορίων (Εικ. 2.18) και εμφανίζεται και η τελική ρίζα της εξίσωσης (Εικ. 2.19), μιας και ικανοποιείται το κριτήριο της ακρίβειας που έθεσε ο χρήστης




Εικόνα 2.18: Η γραφική παράσταση της συνάρτησης μετά την εύρεση της ρίζας της



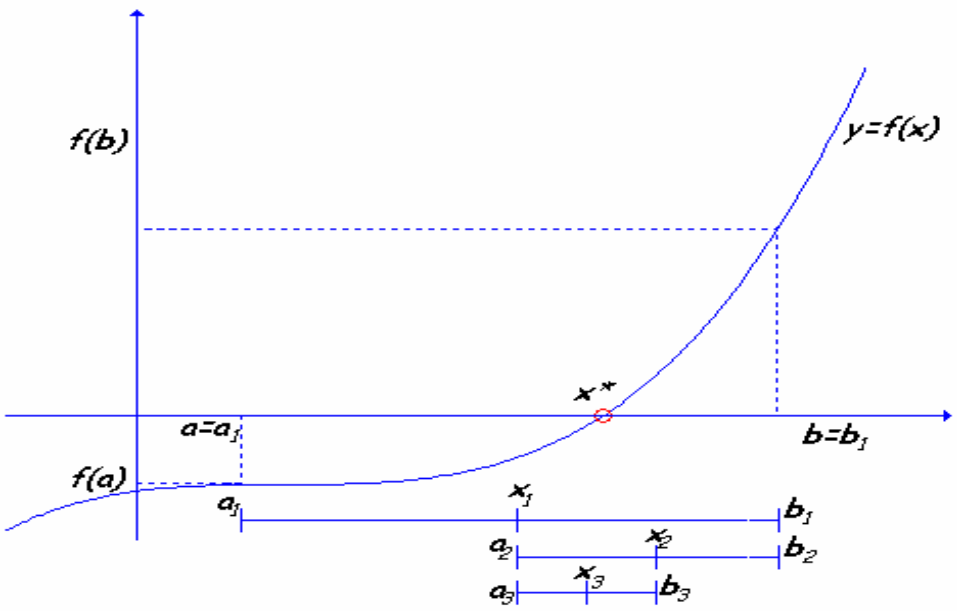

Εικόνα 2.19: Η τελική ρίζα της εξίσωσης

## 2.4 Μέθοδος Διχοτόμησης

Η μέθοδος της διχοτόμησης είναι ένας αλγόριθμος εύρεσης ριζών, ο οποίος λειτουργεί με τον επαναλαμβανόμενο διαμοιρασμό ενός αρχικού διαστήματος κατά το ήμισυ. Έπειτα, διαλέγει το ένα από τα δύο μέρη στα οποίο υπάρχει η ρίζα. Για να οριστεί το αρχικό διάστημα (π.χ. $(a,b)$), πρέπει να ισχύει $f(a)*f(b)\neq 0$, να υπάρχει δηλαδή ρίζα στο διάστημα αυτό. Αφού επιλεγεί ένα από τα δύο υποδιαστήματα, η μέθοδος το διαχωρίζει εκ νέου στα δύο, θέτοντας τα νέα όρια $(a,b)$. Η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σε ένα πολύ μικρό διάστημα όπου το μέσο του θα ταυτίζεται με την τελική ρίζα της εξίσωσης. Για να αρχίσει η μέθοδος, πρέπει να οριστεί η επιθυμητή ακρίβεια με την οποία θέλουμε να βρεθεί η

ρίζα, αλλά και ο αριθμός των επαναλήψεων. Η μέθοδος της διχοτόμησης συγκλίνει πολύ αργά σε σχέση με τις άλλες μεθόδους.

Γραφική παράσταση όπου φαίνονται τα διαδοχικά βήματα της μεθόδου:




Εικόνα 2.20

### 2.4.1 Παράδειγμα της μεθόδου *Διχοτόμησης*

Να βρεθεί με τη μέθοδο της διχοτόμησης η ρίζα της εξίσωσης $x^2 - x^3$, με αρχικό διάστημα το (-1, 2) , επιθυμητή ακρίβεια δύο δεκαδικών ψηφίων και αριθμό επαναλήψεων 10.

Πρέπει να διαπιστωθεί αν υπάρχει ρίζα εντός του διαστήματος (-1, 2). Αυτό ισχύει, αφού $f(-1) = 2 > 0, f(2) = -8 < 0$ και $f(-1) * f(2) < 0$.

*1η επανάληψη :* $(a, b)$= (-1, 2) . Το μέσο του διαστήματος αυτού $x_\mu$ είναι το 0,5 με ακρίβεια $\varepsilon = 0{,}1875$

2η επανάληψη: $(a, b)$= (0,5 , 2) . $x_\mu = 1{,}25$ , $\varepsilon$ =0,75

3η επανάληψη: $(a, b)$= ( 0,5 , 1,25 ) . $x_\mu = 0{,}875$ , $\varepsilon = 0{,}375$

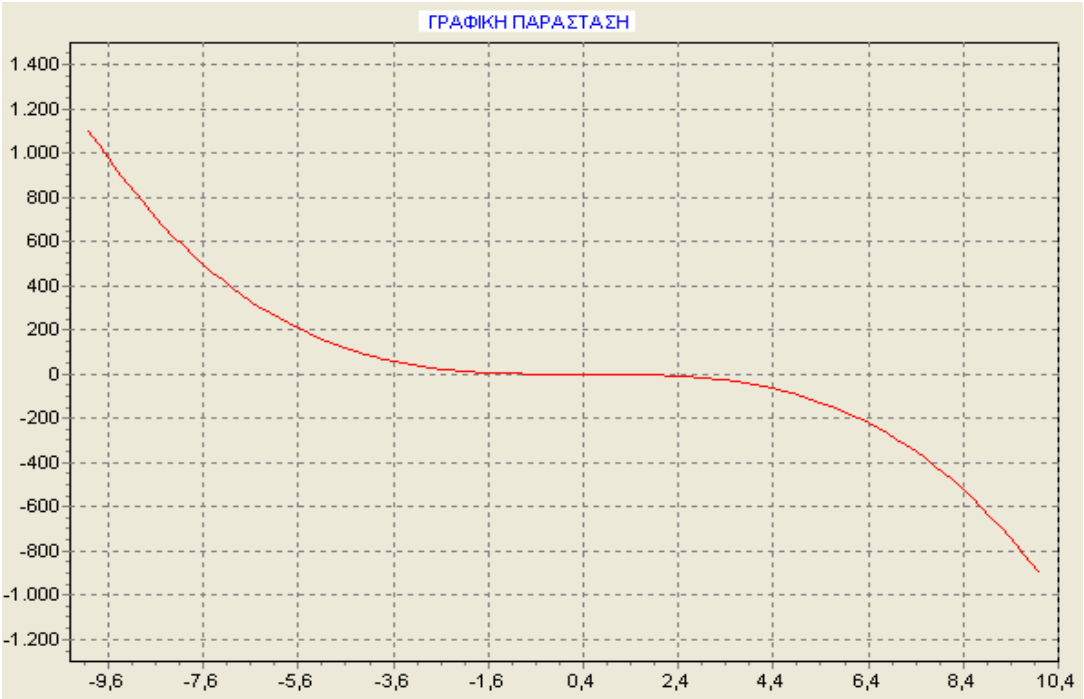
4η επανάληψη: $(a, b)$= (0,875 , 1,25 ) . $x_\mu = 1{,}0625$ , $\varepsilon = 0{,}1875$

5η επανάληψη: $(a, b)$= (0,875 , 1,0625 ) . $x_\mu = 0{,}96875$ , $\varepsilon = 0{,}09375$

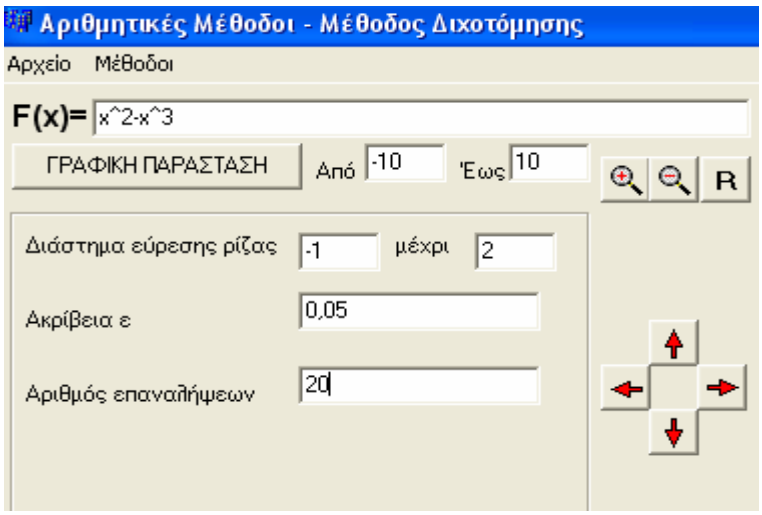
6η επανάληψη: $(a,b)$= (0,96875 , 1,0625 ) . $x_\mu$ = 1,015625 , $\varepsilon$ = 0,046875

Στην έκτη επανάληψη η μέθοδος τερματίζεται γιατί η ακρίβεια είναι μικρότερη από αυτή που ορίστηκε στην αρχή $(0{,}046875 < 0{,}05)$ και το μέσο του διαστήματος $(a,b)$αποτελεί και την τελική ρίζα $x = 1{,}015625$ .

Παρακάτω δίνεται η επίλυση της εξίσωσης με τη χρήση της εφαρμογής της πτυχιακής Μετά την πληκτρολόγηση της συνάρτησης και των ορίων σχεδίασης της γραφικής παράστασης, ο χρήστης θέτει τα αρχικά όρια (-1, 2), την επιθυμητή ακρίβεια (0,05) και τον αριθμό των επαναλήψεων (π.χ 20)

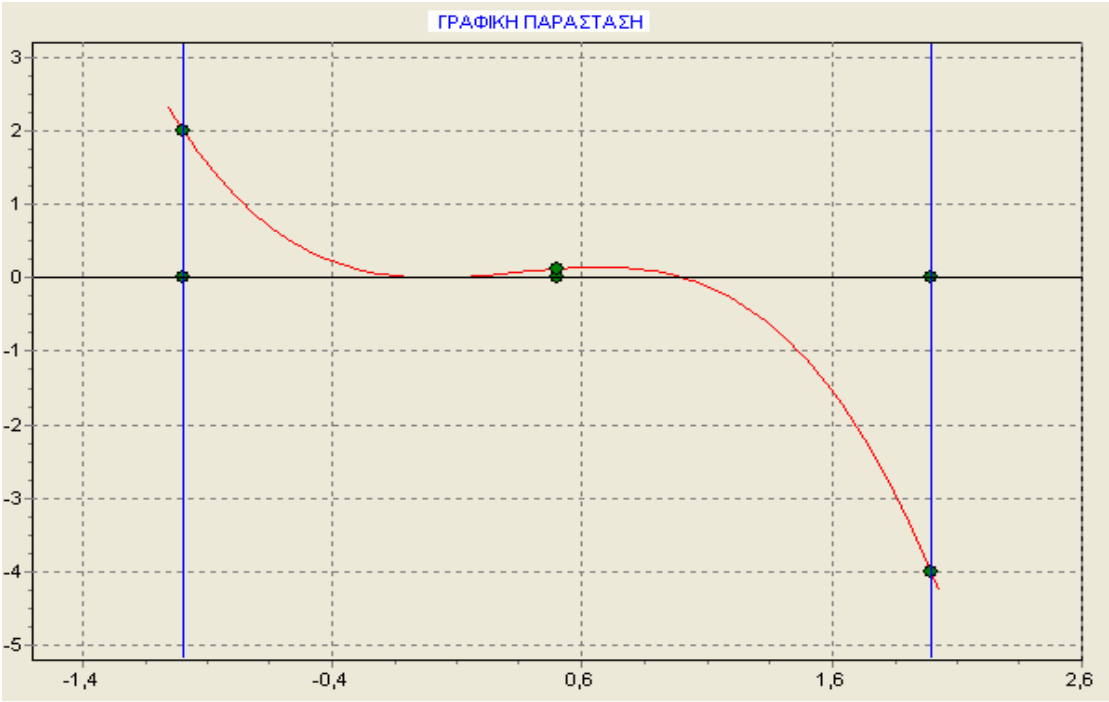



Εικόνα 2.21: Η γραφική παράσταση της συνάρτησης

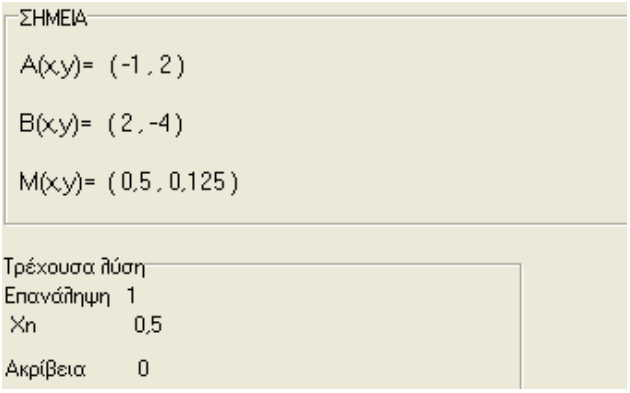



Εικόνα 2.22: Εισαγωγή παραμέτρων της συνάρτησης

Με το πάτημα του "επόμενο βήμα" εμφανίζεται το μισό από το αρχικό διάστημα, (Εικ. 2.23) καθώς και οι συντεταγμένες των νέων ορίων αλλά και του σημείου που αποτελεί το νέο μέσο (τρέχουσα λύση)(Εικ. 2.24).

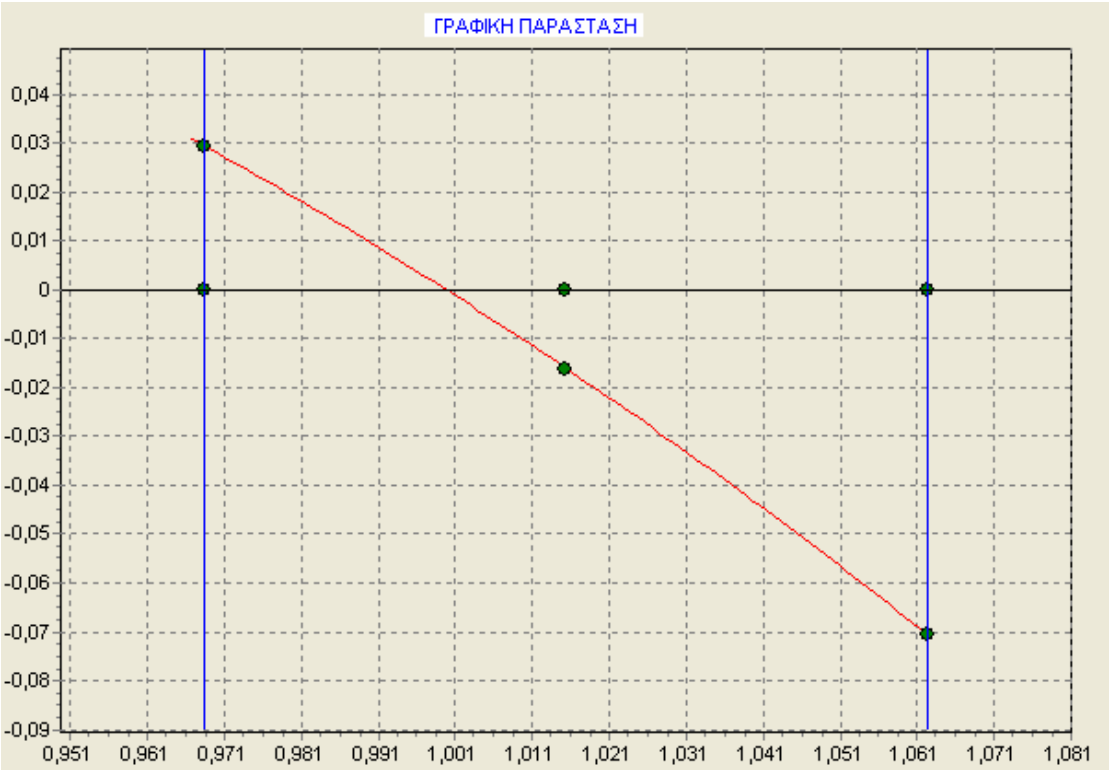



Εικόνα 2.23: Γραφική παράσταση της συνάρτησης εντός των νέων ορίων




Εικόνα 2.24: Τρέχουσα λύση της εξίσωσης και τα νέα όρια σχεδίασης της γραφικής παράστασης

Μετά τις έξι επαναλήψεις, θα έχει βρεθεί η τελική ρίζα και θα εμφανίζεται τόσο στη γραφική παράσταση (Εικ. 2.25) όσο και στις καρτέλες των αποτελεσμάτων (Εικ. 2.26):




Εικόνα 2.25: Η γραφική παράσταση της συνάρτησης μετά την εύρεση της ρίζας της

ΣΗΜΕΙΑ

A(x,y)= ( 0,96875 , 0,029327392578125 )

B(x,y)= ( 1,0625 , -0,070556640625 )

M(x,y)= ( 1,015625 , -0,0161170959472656 )

Τελική λύση

| | |
|---|---|
| Επανάληψη | 6 |
| Ρίζα εξίσωσης | 1,015625 |
| Ακρίβεια | 0,046875 |

Εικόνα 2.26: Η τελική ρίζα της εξίσωσης

Τα αποτελέσματα της κάθε επανάληψης (τιμές νέων ορίων, συντεταγμένες των μέσων) εμφανίζονται και συνολικά στον πίνακα που αυξάνεται σε κάθε εφαρμογή της μεθόδου:

| ΕΠΑΝΑΛΗΨΕΙΣ | X_A | X_B | X_M | Y_M |
|---|---|---|---|---|
| 1 | -1 | 2 | 0,5 | 0,125 |
| 2 | 0,5 | 2 | 1,25 | -0,390625 |
| 3 | 0,5 | 1,25 | 0,875 | 0,095703125 |
| 4 | 0,875 | 1,25 | 1,0625 | -0,070556640625 |
| 5 | 0,875 | 1,0625 | 0,96875 | 0,029327392578125 |
| 6 | 0,96875 | 1,0625 | 1,015625 | -0,0161170959472656 |

Εικόνα 2.27: Τα αποτελέσματα της κάθε επανάληψης της μεθόδου

## 2.5 Μέθοδος Regula falsi

Η μέθοδος της Regula falsi στηρίζεται στη δημιουργία μιας ακολουθίας σημείων, τα οποία συγκλίνουν προς την ρίζα, με τον ακόλουθο τρόπο:

Δοσμένων των αρχικών σημείων $(a,b)$, για τα οποία ισχύει $f(a)*f(b)<0$ θεωρούμε την ευθεία που περνά από τα σημεία $f(a)$ και $f(b)$. Το σημείο όπου τέμνει αυτή η ευθεία τον οριζόντιο άξονα (και αποτελεί την τρέχουσα λύση της εξίσωσης) δίνεται από τον τύπο:

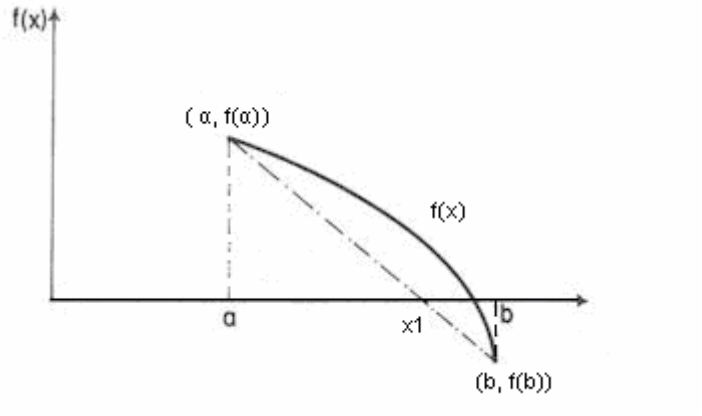
$$x_1 = b - \frac{b-a}{f(b)-f(a)} f(b)$$

Η μέθοδος επιλέγει για νέο διάστημα στο οποίο θα αναζητήσει λύση αυτό όπου θα ισχύει και πάλι $f(a)*f(b)<0$. Έτσι, αν ισχύει $f(x_1)*f(b)<0$, η νέα ρίζα θα είναι

$$x_2 = b - \frac{b-x_1}{f(b)-f(x_1)} f(b)$$

Μετά από έναν ορισμένο αριθμό επαναλήψεων ή μέχρι να φτάσει η ακρίβεια της ρίζας σε μια επιθυμητή τιμή, η μέθοδος θα τερματιστεί.

Γραφικά η μέθοδος δίνεται έτσι:




Εικόνα 2.28

### 2.5.1 Παράδειγμα της μεθόδου Regula-falsi

Έστω ότι θέλουμε να εφαρμόσουμε τη Regula-falsi για την επίλυση της εξίσωσης $f(x) = x^2 - 3$, με δοσμένα τα αρχικά σημεία $(a, b)=(0, 2)$, ακρίβεια ($\varepsilon$) τριών δεκαδικών ψηφίων και αριθμό επαναλήψεων 8. Τότε ισχύει $f(a) = -3$ και $f(b) = 1$.

1η επανάληψη: $x_1 = 2 - \dfrac{2-0}{1+3} = 1{,}5$ , $\varepsilon = 0{,}2142$

Αντικαθιστώντας την τιμή αυτή στην εξίσωση, προκύπτει ότι $f(x_1) < 0$. Και επειδή ισχύει $f(b) > 0$, το νέο διάστημα για τον προσδιορισμό της ρίζας θα είναι το $(x_1, b)$

2η επανάληψη: $x_2 = b - \dfrac{b - x_1}{f(b) - f(x_1)} f(b) = 2 - \dfrac{2-1{,}5}{1+0{,}75} = 1{,}715$ , $\varepsilon = 0{,}2142$

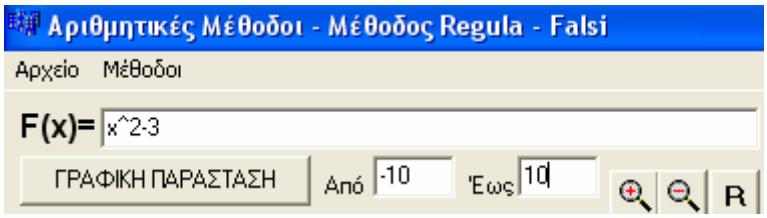
3η επανάληψη: $x_3 = 1{,}7307$ , $\varepsilon = 0{,}0164$

4η επανάληψη: $x_4 = 1{,}7319$ , $\varepsilon = 0{,}0011$

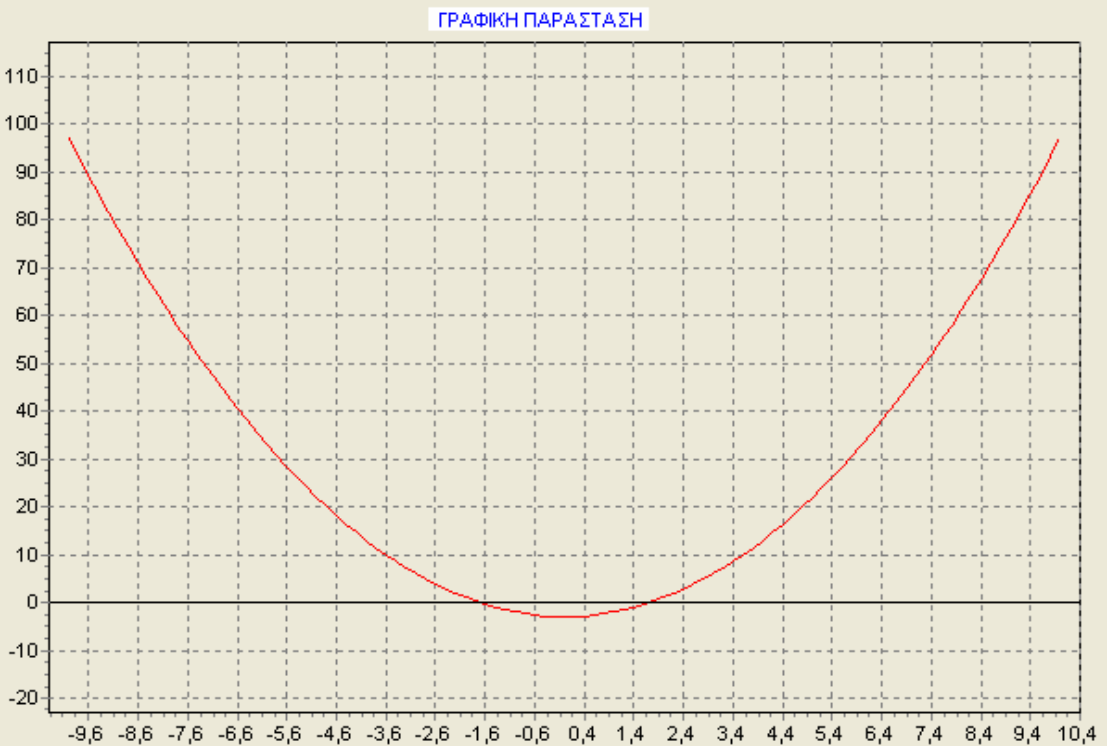
Η εκτέλεση της μεθόδου τερματίζεται στην 4η επανάληψη λόγω της τιμής της ακρίβειας ($0{,}0011 < 0{,}005$)

Με χρήση της εφαρμογής που αναπτύχθηκε στη C++, το παράδειγμα θα εκτελεστεί ως εξής:

Αρχικά πληκτρολογείται η εξίσωση προς λύση (Εικ. 2.29), ώστε να εμφανιστεί η γραφική της παράσταση (Εικ. 2.30) εντός των ορίων που έχει ορίσει ο χρήστης:

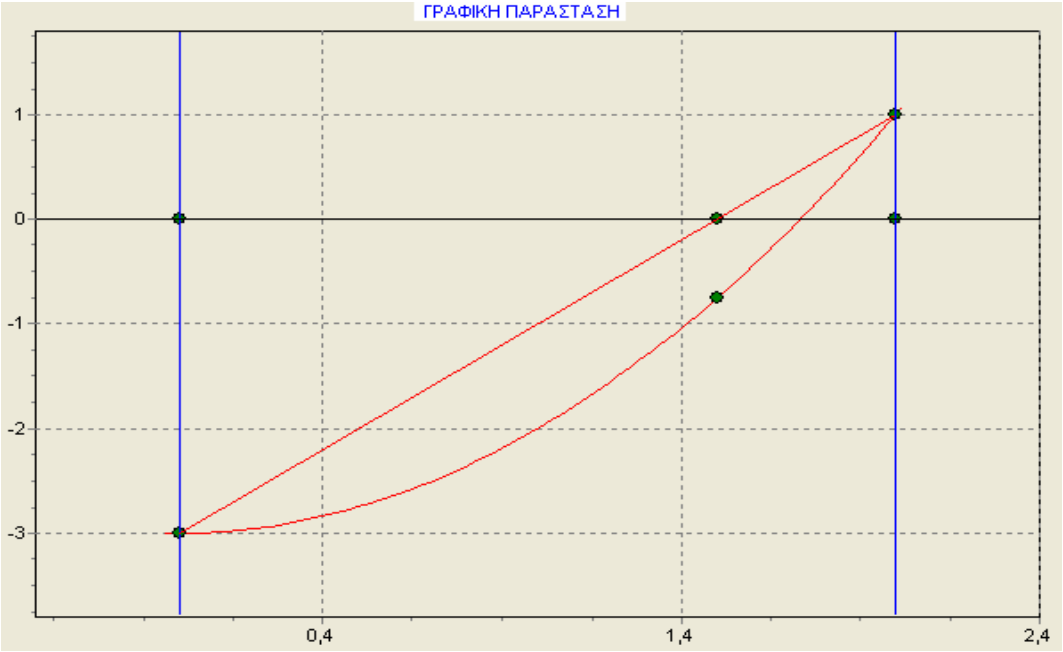



Εικόνα 2.29: Εισαγωγή συνάρτησης και ορίων της γραφικής της παράστασης

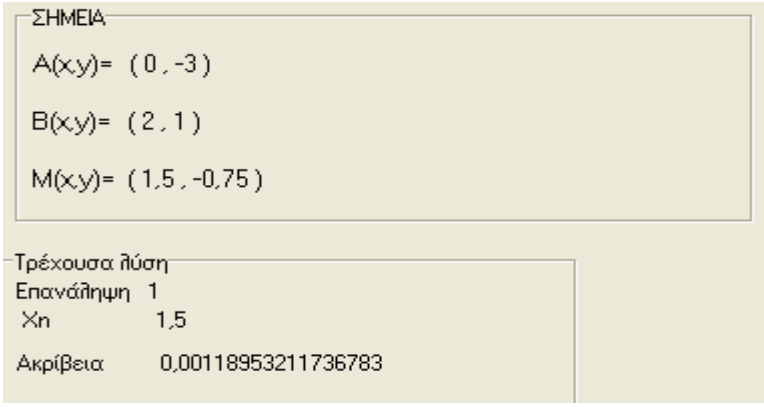



Εικόνα 2.30: Γραφική παράσταση της συνάρτησης

Στη συνέχεια, ο χρήστης πληκτρολογεί τις αρχικές τιμές (0, 2) , την επιθυμητή ακρίβεια ( $\varepsilon$=0,005 ) και τον αριθμό των επαναλήψεων της εκτέλεσης της μεθόδου, με αποτέλεσμα μόλις πατηθεί το κουμπί “επόμενο βήμα” η γραφική παράσταση να περιοριστεί εντός των ορίων (Εικ. 2.31), εμφανίζοντας παράλληλα και τις συντεταγμένες των ορίων, των σημείων που τέμνουν τον οριζόντιο άξονα και την τρέχουσα λύση στην επανάληψη όπου βρισκόμαστε (Εικ. 2.32).



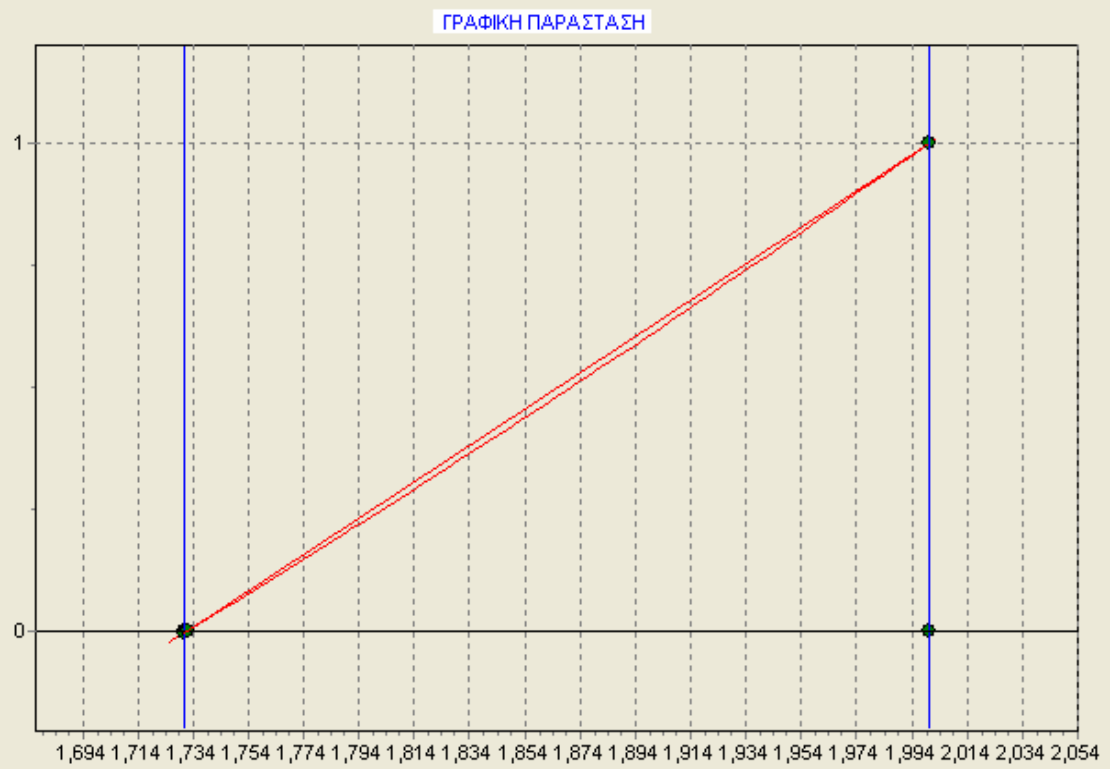

Εικόνα 2.31: η γραφική παράσταση εντός των ορίων της συνάρτησης




Εικόνα 2.32: Τρέχουσα λύση της εξίσωσης και τα νέα όρια σχεδίασης της γραφικής παράστασης

Αφού η διαδικασία επαναληφθεί συνολικά 4 φορές, η μέθοδος θα τερματιστεί εμφανίζοντας την τελική ρίζα, και την ακριβή της θέση στη γραφική παράσταση (Εικ. 2.33, 2.34):

Λόγω της μικρής ακρίβειας, παρατηρείται ότι η ευθεία που ενώνει τα δύο άκρα σχεδόν ταυτίζεται με την καμπύλη της εξίσωσης.




Εικόνα 2.33: Η γραφική παράσταση της συνάρτησης μετά την εύρεση της ρίζας της

ΣΗΜΕΙΑ

A(x,y)= ( 1,73076923076923 , -0,00443786982248806 )

B(x,y)= ( 2 , 1 )

M(x,y)= ( 1,7319587628866 , -0,000318843660318091 )

Τελική λύση

| | |
|---|---|
| Επανάληψη | 4 |
| Ρίζα εξίσωσης | 1,7319587628866 |
| Ακρίβεια | 0,00118953211736783 |

Εικόνα 2.34: Η τελική ρίζα της εξίσωσης

Κατά τη διάρκεια της επίλυσης, αυξάνεται το μέγεθος του πίνακα που περιέχει συνοπτικά τις διαδοχικές ρίζες της εξίσωσης:

| ΕΠΑΝΑΛΗΨΕΙΣ | X_A | X_B | X_M | Y_M |
|---|---|---|---|---|
| 1 | 0 | 2 | 1,5 | -0,75 |
| 2 | 1,5 | 2 | 1,71428571428571 | -0,0612244897959333 |
| 3 | 1,71428571428571 | 2 | 1,73076923076923 | -0,00443786982248806 |
| 4 | 1,73076923076923 | 2 | 1,7319587628866 | -0,000318843660318091 |
| | | | | |

Εικόνα 2.35: Τα αποτελέσματα της κάθε επανάληψης της μεθόδου

# ΚΕΦΑΛΑΙΟ 3

## Αριθμητική Ολοκλήρωση

Είναι συχνά δύσκολο ή και αδύνατο να υπολογιστεί ο αναλυτικός τύπος που δίνει το ορισμένο ολοκλήρωμα μιας συνεχούς συνάρτησης $f(x)$ σε ένα κλειστό διάστημα $[a,b]$

$$I = \int_a^b f(x)dx \qquad (3.1)$$

Εντούτοις, μπορούμε να χρησιμοποιήσουμε αριθμητικές μεθόδους μέσω των οποίων υπολογίζουμε την αριθμητική τιμή του ολοκληρώματος (3.1) με ακρίβεια πολλών σημαντικών ψηφίων. Θα πρέπει να τονισθεί ότι οι αριθμητικές μέθοδοι ολοκλήρωσης έχουν ως εξαγόμενο την αριθμητική τιμή του ολοκληρώματος για συγκεκριμένες αριθμητικές τιμές των ορίων a και b, και όχι ένα γενικό τύπο για το ολοκλήρωμα.

Στις περισσότερες αριθμητικές μεθόδους ολοκλήρωσης, ακολουθείται η "τεχνική" μιας κατάλληλης διαμέρισης του διαστήματος ολοκλήρωσης $[a,b]$ σε ένα μεγάλο πλήθος υποδιαστημάτων μικρού πλάτους, και χρησιμοποιούνται προσεγγιστικοί τύποι ολοκλήρωσης σε κάθε υποδιάστημα.

Θεωρώντας μία διαμέριση του διαστήματος $[a,b]$ σε N διαστήματα ίσου πλάτους, το πλάτος του κάθε διαστήματος είναι

$$h = \frac{b-\alpha}{N}$$

και ορίζονται τα διαδοχικά ισαπέχοντα σημεία $x_0 = a, x_1 = a+h, x_2 = a+2h, ..., x_N = b$, με βήμα h, που δίνονται από το γενικό τύπο

$$x_i = a + ih, i = 0,1,2,...,N$$

Το ολοκλήρωμα (3.1) εκφράζεται ως άθροισμα των ολοκληρωμάτων σε κάθε υποδιάστημα της μορφής $[x_i, x_i + 1]$:

$$I = \int_a^b f(x)dx = \sum_{i=0}^{N-1} I_i = \int_{X0}^{X1} f(x)dx + \int_{X1}^{X2} f(x)dx + \ldots + \int_{X_{N-1}}^{X_N} f(x)dx \qquad (3.2)$$

Και επομένως το πρόβλημα επικεντρώνεται στον υπολογισμό των επιμέρους ολοκληρωμάτων

$$I_i = \int_{X_i}^{X_{i+1}} f(x)dx = \int_{X_i}^{X_i+h} f(x)dx \qquad (3.3)$$

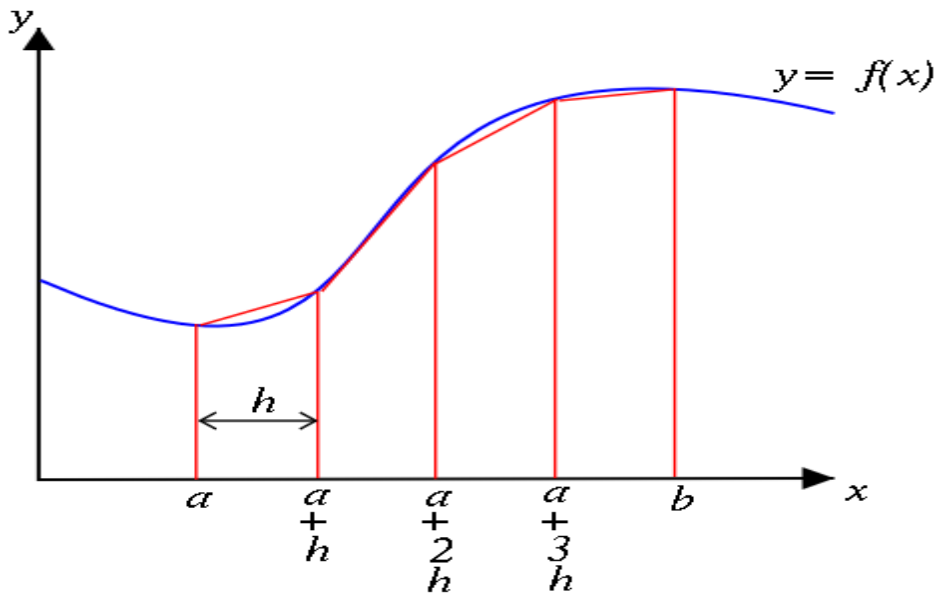
Μία αριθμητική μέθοδος υπολογισμού του ολοκληρώματος (3.3) σε διάστημα μικρού πλάτους h ονομάζεται τοπικός κανόνας ολοκλήρωσης. Διαθέτοντας ένα τοπικό κανόνα ολοκλήρωσης, υπολογίζουμε χωριστά κάθε όρο του αθροίσματος (3.2), και στη συνέχεια το άθροισμα όλων των όρων. Το άθροισμα N όρων υπολογισμένων με βάση ένα τοπικό κανόνα ολοκλήρωσης ονομάζεται εκτεταμένος κανόνας ολοκλήρωσης.

### 3.1 Κανόνας του τραπεζίου

Ο τραπεζοειδής κανόνας είναι μία μέθοδος για την εξεύρεση, κατά προσέγγιση, μίας τιμής για ένα συγκεκριμένο ολοκλήρωμα. Ας υποθέσουμε ότι έχουμε το ολοκλήρωμα:

$$\int_a^b f(x)dx$$

Πρώτα διαιρείται η περιοχή κάτω από την καμπύλη σε n τμήματα, κάθε ένα από τα οποία έχει πλάτος. $h = \frac{(b-a)}{n}$.




Εικόνα 3.1

Η περιοχή κάτω από την καμπύλη διαιρείται για παράδειγμα σε n=4 λωρίδες. Κάθε τμήμα προσεγγίζει ένα τραπέζιο. Το άθροισμα των εμβαδών αυτών των τμημάτων δίνει προσεγγιστικά το ζητούμενο ολοκλήρωμα

Το εμβαδό του κάθε τραπεζίου δίνεται από τον τύπο:

$$\frac{h}{2}(f(\alpha)+f(\alpha+h))$$

Κάθε i τμήμα (όπου i το βήμα με το οποίο μεταβαίνουμε από το ένα τραπέζιο στο επόμενο) θα ισούται με:

$$\frac{h}{2}(f(a+(i-1)h)+f(a+ih))$$

Το άθροισμα αυτών των τμημάτων μας δίνει προσεγγιστικά μια τιμή για το ολοκλήρωμά μας:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n}\frac{h}{2}(f(\alpha+(i-1)h)+f(a+ih))$$

Αυτή η προσέγγιση για το ολοκλήρωμα βελτιώνεται όσο αυξάνεται το n, όσο δηλαδή χωρίζεται σε μικρότερα τμήματα το τμήμα κάτω από την καμπύλη.

### 3.1.1 Παράδειγμα του κανόνα τραπεζίου

Να υπολογιστεί με τους τύπους του τραπεζίου για n=4 το ολοκλήρωμα $I = \int_0^4 \frac{x^3}{12} dx$.

Έχουμε την εξίσωση $f(x) = \frac{x^3}{12}$. Για n=4, στο διάστημα $[0,4]$ το βήμα είναι $h = \frac{b-a}{n} = \frac{4-0}{4} = 1$.

Με αντικατάσταση των τιμών 0, 1, 2, 3, 4 στην $f(x)$, προκύπτει ο παρακάτω πίνακας τιμών:

| x | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| f(x) | 0 | 1/12 | 8/12 | 27/12 | 64/12 |

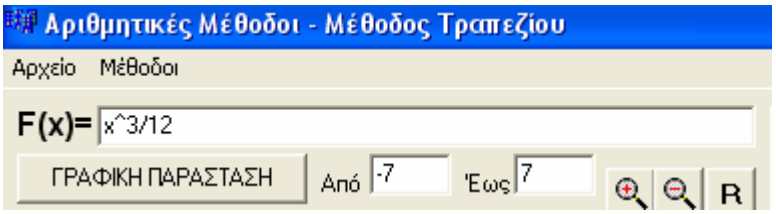
Ο σύνθετος τύπος του τραπεζίου για n=4 δίνει

$$T_4 = \frac{h}{2}[f(0) + 2f(1) + 2f(2) + 2f(3) + f(4)] = \frac{1}{2}[0 + 2\frac{1}{12} + 2\frac{8}{12} + 2\frac{27}{12} + \frac{64}{12}]$$

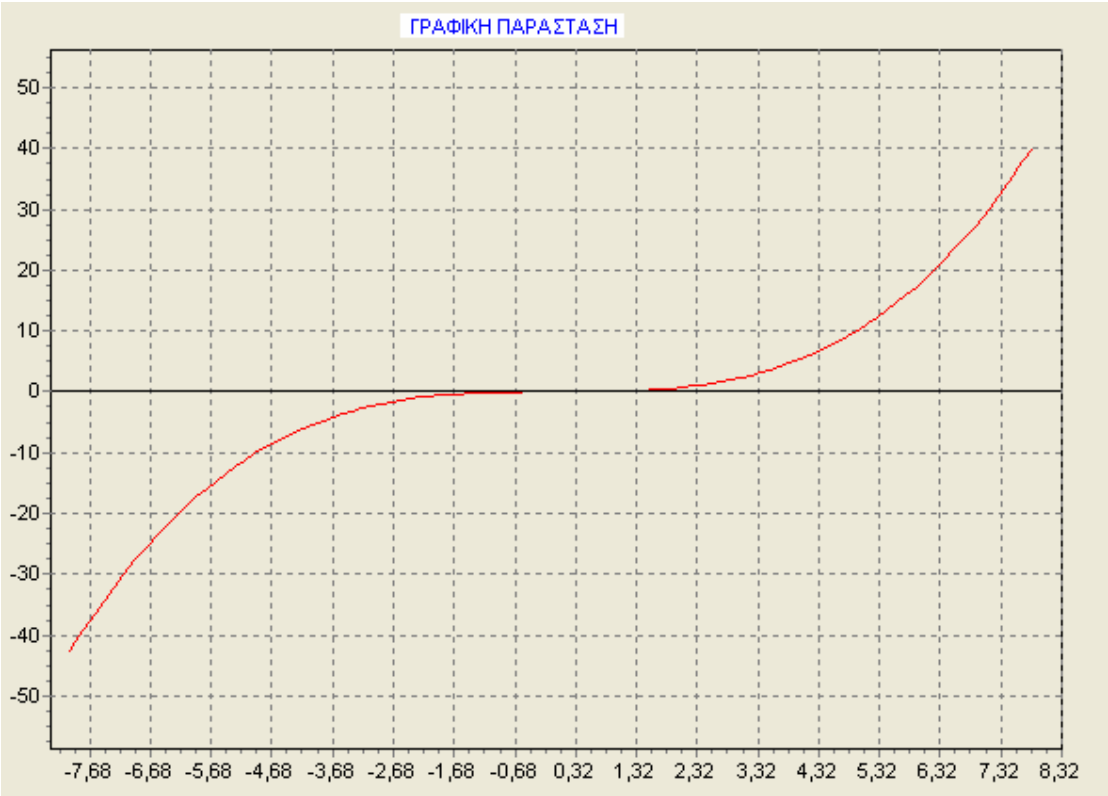
$$= \frac{136}{24} = 5.66666$$

Παρακάτω δίνεται ο υπολογισμός του ολοκληρώματος με χρήση της εφαρμογής της πτυχιακής:

Αρχικά πληκτρολογείται η εξίσωση $f(x)$, τα όρια σχεδίασης της γραφικής παράστασης και με το πάτημα του κουμπιού "ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ" αναπαρίσταται γραφικά η εξίσωση (Εικ. 3.2, 3.3).

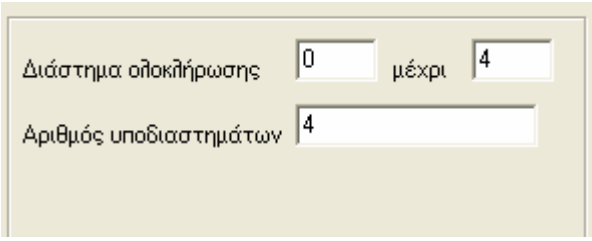



Εικόνα 3.2: Εισαγωγή της συνάρτησης και των ορίων της σχεδίασής της

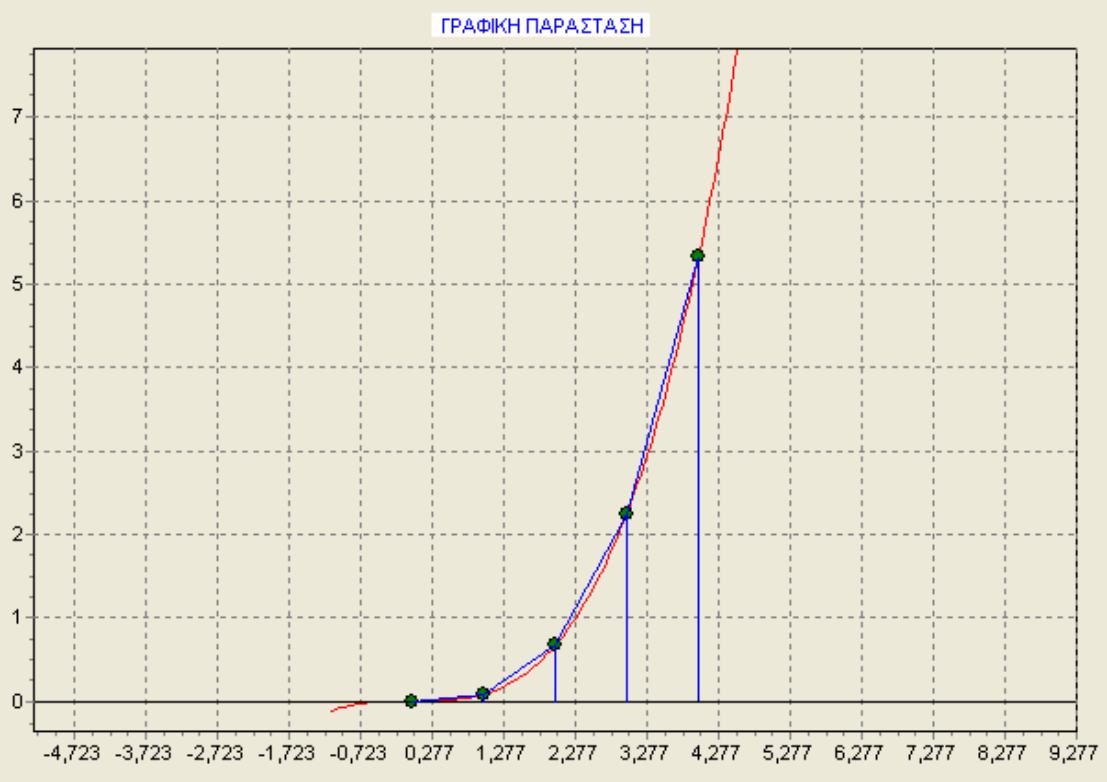



Εικόνα 3.3: Γραφική παράσταση της συνάρτησης προς λύση

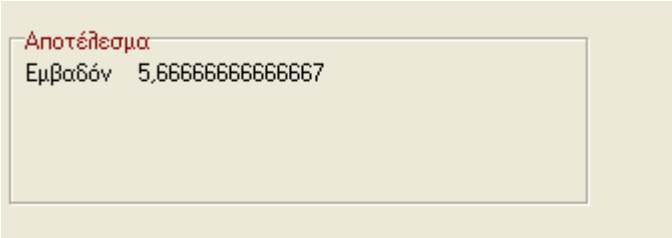
Στη συνέχεια ο χρήστης συμπληρώνει τα όρια του ολοκληρώματος (0, 4) , τον αριθμό των υποδιαστημάτων *n* (Εικ. 3.4) και κλικάροντας στο κουμπί "ΕΚΤΕΛΕΣΗ", εμφανίζεται γραφικά η ακολουθία των τραπεζίων (Εικ. 3.5) και το αποτέλεσμα της μεθόδου (Εικ.3.6) .




Εικόνα 3.4: Εισαγωγή του διαστήματος ολοκλήρωσης και του αριθμού υποδιαστημάτων




Εικόνα 3.5: Γραφική αναπαράσταση της ακολουθίας των τραπεζίων




Εικόνα 3.6: Το αποτέλεσμα της μεθόδου (εμβαδόν της περιοχής των τραπεζίων)

Οι συντεταγμένες στις κορυφές των τραπεζίων εμφανίζονται στη φόρμα της εφαρμογής αν τις επιλέξουμε επάνω στη γραφική παράσταση:

| | | | | |
|---|---|---|---|---|
| X: 0 | X: 1 | X: 2 | X: 3 | X: 4 |
| Y: 0 | Y: 0,0833333333333333 | Y: 0,666666666666667 | Y: 2,25 | Y: 5,33333333333333 |

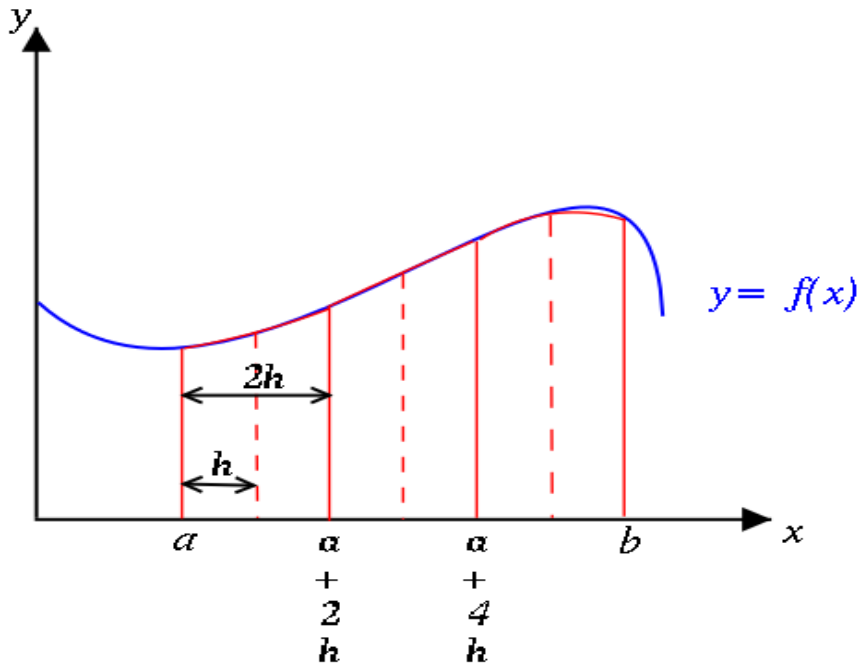
Εικόνα 3.7: Οι συντεταγμένες των κορυφών των τραπεζίων

## 3.2 Μέθοδος Simpson

Ο κανόνας του Simpson είναι ένας τύπος Newton-Cotes για να προσεγγίσει το ολοκλήρωμα μιας συνάρτησης που χρησιμοποιεί τα τετραγωνικά πολυώνυμα (δηλαδή παραβολικά τόξα αντί των τμημάτων ευθειών γραμμών που χρησιμοποιούνται στον τραπεζοειδή κανόνα).

Δεδομένου ότι χρησιμοποιεί τα τετραγωνικά πολυώνυμα στις κατά προσέγγιση συναρτήσεις, ο κανόνας του Simpson δίνει πραγματικά τα ακριβή αποτελέσματα όταν προσεγγίζει τα ολοκληρώματα πολυωνύμων τρίτου βαθμού.

Γραφική παράσταση του κανόνα Simpson:




Εικόνα 3.8

Ο γενικός τύπος του κανόνα του Simpson είναι:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{3}\{f(x_0)+4[f(x_1)+f(x_3)+...+f(x_{n-1})]+2[f(x_2)+f(x_4)+...+f(x_{n-2})]+f(x_n)\}$$

Όπου $x_i = a + i\Delta x, i = 0,1,2,...,n,\ \Delta x = \dfrac{b-a}{n}$ , και n άρτιος.

### 3.2.1 Παράδειγμα της μεθόδου *Simpson*

Θα χρησιμοποιήσουμε τον κανόνα Simpson για n=6. για την προσέγγιση της τιμής του:

$$\int_1^2 \frac{1}{x}dx \quad \text{Έχουμε:} \quad \Delta x = \frac{2-1}{6} = \frac{1}{6} \quad \text{και} \quad x_i = 1 + \frac{i}{6} = \frac{6+i}{6} \quad \text{για } i=1,2,3,\ldots$$

Με αντικατάσταση στον παραπάνω τύπο, υπολογίζονται τα $x_i$:

$$x_0 = 1 + \frac{0}{6} = 1 \ , \quad x_1 = 1 + \frac{1}{6} = \frac{7}{6} \ , \quad x_2 = 1 + \frac{2}{6} = \frac{8}{6} \ , \quad x_3 = 1 + \frac{3}{6} = \frac{9}{6},$$

$$x_4 = 1 + \frac{4}{6} = \frac{10}{6}, \quad x_5 = 1 + \frac{5}{6} = \frac{11}{6}, \quad x_6 = 1 + \frac{6}{6} = 2$$

και με αντικατάσταση των τιμών στον τύπο της εξίσωσης, υπολογίζονται τα $f(x_i)$:

$$f(x_0) = \frac{1}{1} = 1 \ , \quad f(x_1) = \frac{1}{7/6} = \frac{6}{7}, \quad f(x_2) = \frac{6}{8}, \quad f(x_3) = \frac{6}{9}$$

$$f(x_4) = \frac{6}{10} \ , \quad f(x_5) = \frac{6}{11}, \quad f(x_6) = \frac{1}{2}$$

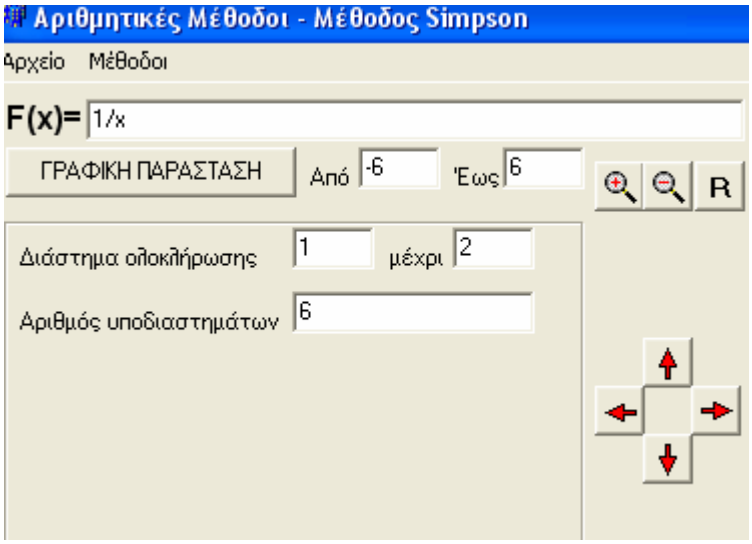
Σύμφωνα με τον κανόνα του Simpson προκύπτει:

$$\int_1^2 \frac{1}{x}dx \approx \frac{\Delta x}{3}\{f(x_0) + 4[f(x_1) + f(x_3) + f(x_5)] + 2[f(x_2) + f(x_4)] + f(x_6)\} =$$

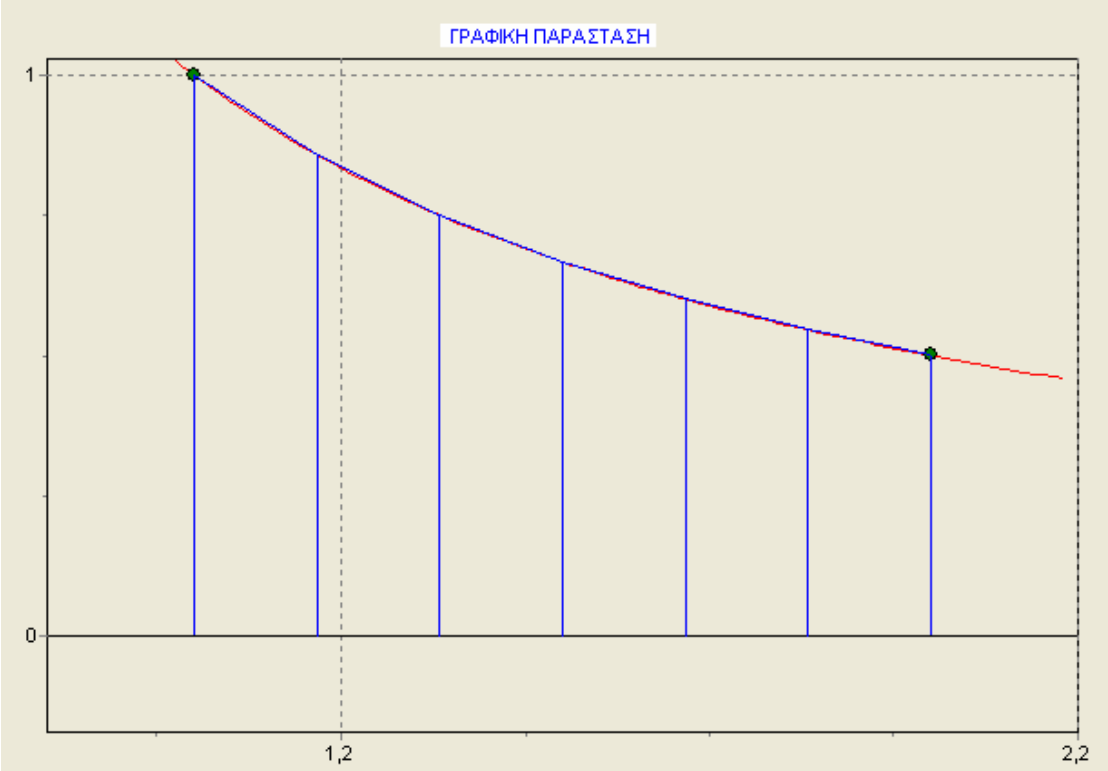
$$\frac{1/6}{3}\{1 + 4[(6/7) + (2/3) + (6/11)] + 2[(6/8) + (6/10)] + 1/2\} = 0{,}69316979$$

Με χρήση της εφαρμογής που αναπτύχθηκε στη C++, το παράδειγμα θα εκτελεστεί ως εξής:

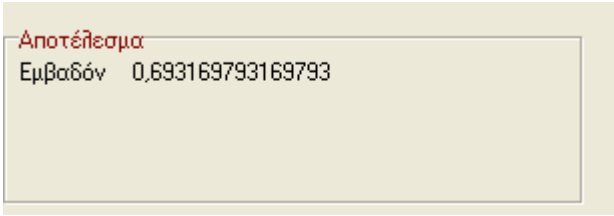
Αρχικά πληκτρολογείται η εξίσωση προς λύση, ώστε να εμφανιστεί η γραφική της παράσταση εντός των ορίων που έχει ορίσει ο χρήστης (Εικ. 3.9). Στη συνέχεια ο χρήστης πληκτρολογεί τα άκρα του ολοκληρώματος (1, 2) και τον αριθμό των υποδιαστημάτων (6), με αποτέλεσμα όταν πατηθεί το κουμπί "ΕΚΤΕΛΕΣΗ", να εμφανιστεί στη γραφική παράσταση η ακολουθία των σχημάτων των οποίων υπολογίζεται το εμβαδό (Εικ. 3.10). Στην καρτέλα αποτελέσματα θα φανεί το συνολικό εμβαδό, δηλαδή η τιμή του ολοκληρώματος (εικ. 3.11).




Εικόνα 3.9 Εισαγωγή των παραμέτρων




Εικόνα 3.10: Η ακολουθία των σχημάτων με τα οποία υπολογίζεται το συνολικό εμβαδό




Εικόνα 3.11 Το αποτέλεσμα της μεθόδου (εμβαδόν)

# ΚΕΦΑΛΑΙΟ 4

# Πολυωνυμική Παρεμβολή

Για πολλές, ακόμα και στοιχειώδεις, συναρτήσεις των Μαθηματικών δεν μπορούμε να υπολογίσουμε ακριβώς παρά μόνο τις τιμές τους $f(x_i)$ σ' ένα πεπερασμένο σύνολο σημείων $\{x_i\}$. Τίθεται λοιπόν το ερώτημα αν μπορούμε να *προσεγγίσουμε* την *f* με μια άλλη συνάρτηση (π.χ. πολυώνυμο, τμηματικά πολυωνυμική συνάρτηση, λόγο πολυωνύμων), η οποία να είναι εύκολο να υπολογισθεί για κάθε *x*. Μια τέτοια προσέγγιση *της f είναι η παρεμβολή.* Γενικότερα, αν μας δίνονται στο επίπεδο τα σημεία $(x_i, y_i)$, $i = 0,1,..., n$, τα οποία αντιστοιχούν σε $n + 1$ "μετρήσεις" στα σημεία $x_i$ της μεταβλητής *y* (που εξαρτάται μόνο από τη μεταβλητή *x*), τίθεται το πρόβλημα της προσέγγισης των δεδομένων $(x_i, y_i)$ από μια απλή συνάρτηση $y = \varphi(x)$, της οποίας η γραφική παράσταση στο επίπεδο *x, y* να είναι "κοντά" στα σημεία $(x_i, y_i)$.

## 4.1 Πολυώνυμο παρεμβολής Lagrange

Το απλούστερο πρόβλημα παρεμβολής, η *παρεμβολή τύπου Lagrange,* είναι να προσδιορίσουμε μια συνάρτηση $\varphi$ από ένα ορισμένο σύνολο συναρτήσεων Σ, η οποία να προσδιορίζεται μόνο από τις τιμές $f(x_i)$ και η οποία να *παρεμβάλλεται* στα σημεία $(x_i, f(x_i))$, δηλαδή να πληρεί τις συνθήκες $\varphi(x_i) = f(x_i)$ $i = 0,1...,n$. Αν το σύνολο Σ είναι αρκετά "πλούσιο" και η *παρεμβάλλουσα* $\varphi$ υπολογίζεται εύκολα, τότε η τιμή της $\varphi(x)$ για $x \neq x_i$, μπορεί να θεωρηθεί ως η ζητούμενη προσέγγιση της *f(x)*

Έστω $x_0,..x_n \in \Re$ ανά δύο διαφορετικά μεταξύ τους σημεία. Σύμφωνα με τα προηγούμενα, για κάθε *i*, $i = 0,...n$, υπάρχει ακριβώς ένα πολυώνυμο $L_i$ τέτοιο ώστε

$$L_i(x_j) = \delta_{ij}, j = 0,\ldots n$$

Το πολυώνυμο $L_i$ μηδενίζεται στα σημεία $x_0,...,x_{i-1},x_{i+1},...,x_n$ και συνεπώς γράφεται στη μορφή

$$L_i(x) = a_i \prod_{\substack{j=0 \\ j\neq i}}^{n} (x - x_j)$$

Προσδιορίζοντας τη σταθερά $\alpha_i$ από τη σχέση $L_i(x_i) = 1$ παίρνουμε

$$L_i(x) = a_i \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad i = 0,\ldots n$$

Τα πολυώνυμα $L_0,...,L_n$ λέγονται *πολυώνυμα του Lagrange* ως προς τα σημεία $\{x_0,..x_n\}$

Αν $p$ είναι το πολυώνυμο παρεμβολής της $f$ στα σημεία $x_0,..x_n$, τότε το $p$ γράφεται στη μορφή

$$p(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$$

η οποία και λέγεται *παράσταση* σε *μορφή του Lagrange* του πολυωνύμου παρεμβολής.

### 4.1.1 Παράδειγμα πολυώνυμου παρεμβολής τύπου Lagrange

Με δεδομένα τα σημεία $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ και αντίστοιχες τιμές f(0)=1, f(1)=2 , f(2)=4, το παρεμβολικό πολυώνυμο υπολογίζεται ως εξής:

Τα πολυώνυμα Lagrange για τα τρία σημεία $x_0$, $x_1$, $x_2$ θα είναι :

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \qquad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

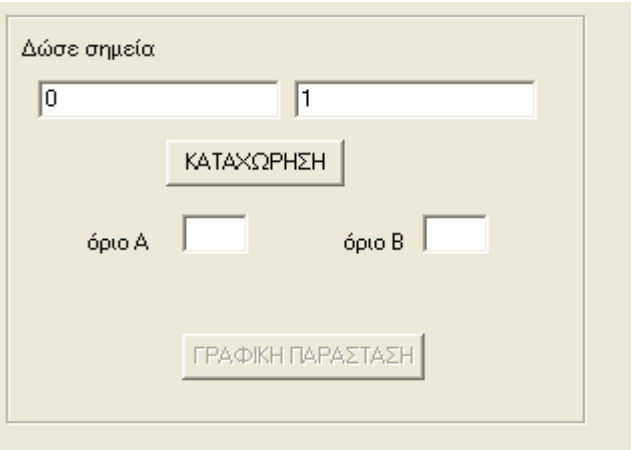
$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Με αντικατάσταση των τιμών των f(0) , f(1) , f(2) και των $L_0(x)$, $L_1(x)$, $L_2(x)$ προκύπτει ότι το πολυώνυμο παρεμβολής είναι:

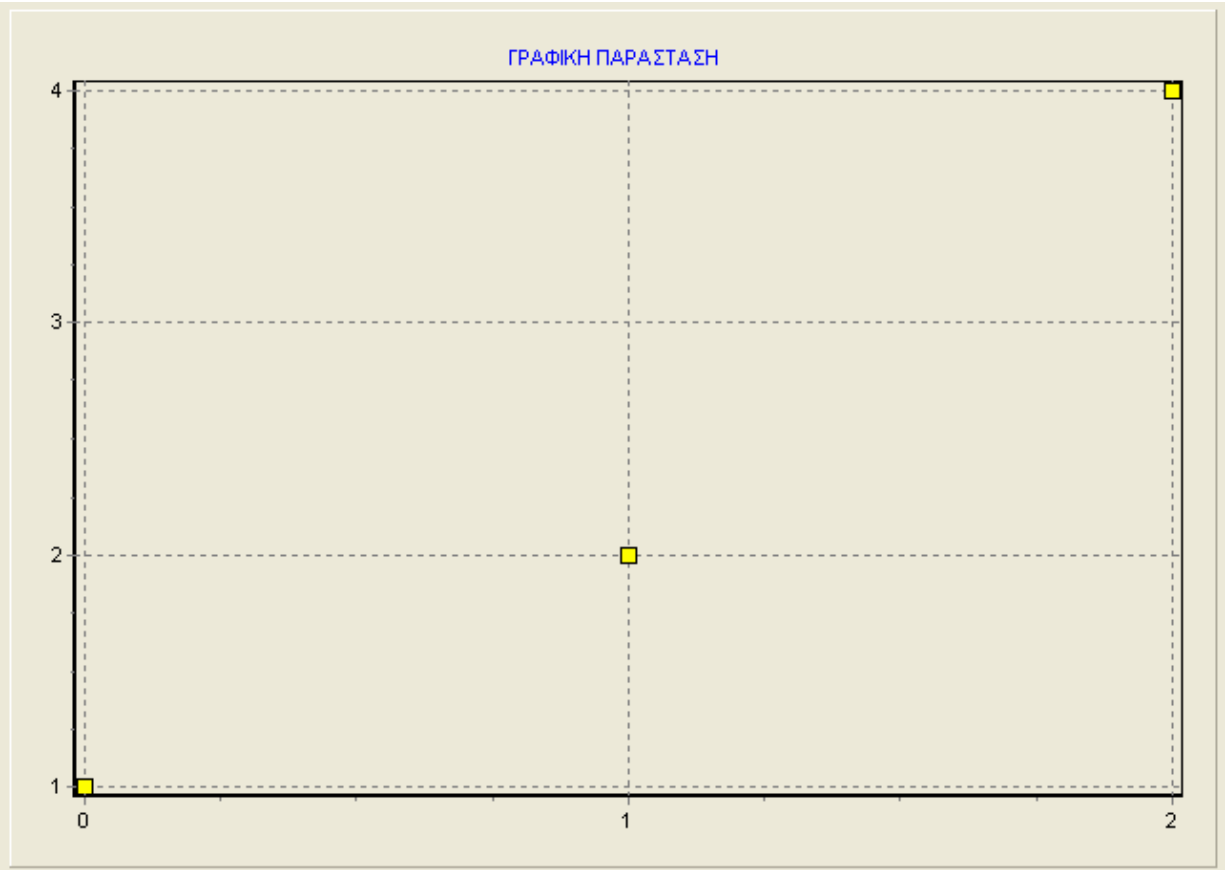
$$p(x) = \sum_{i=0}^{n} f(x_i)L_i(x) = f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x) = \frac{1}{2}x^2 + \frac{1}{2}x + 1$$

Χρησιμοποιώντας την εφαρμογή της πτυχιακής, το παράδειγμα εκτελείται ως εξής:

Αρχικά ο χρήστης πληκτρολογεί τα ζεύγη των τιμών $x_i$ και $f(x_i)$ (Εικ.4.1), πατώντας το κουμπί "ΚΑΤΑΧΩΡΗΣΗ" κάθε φορά μετά την εισαγωγή ενός ζεύγους τιμών, ώστε να αναπαρασταθεί γραφικά το εκάστοτε σημείο (Εικ. 4.2 )

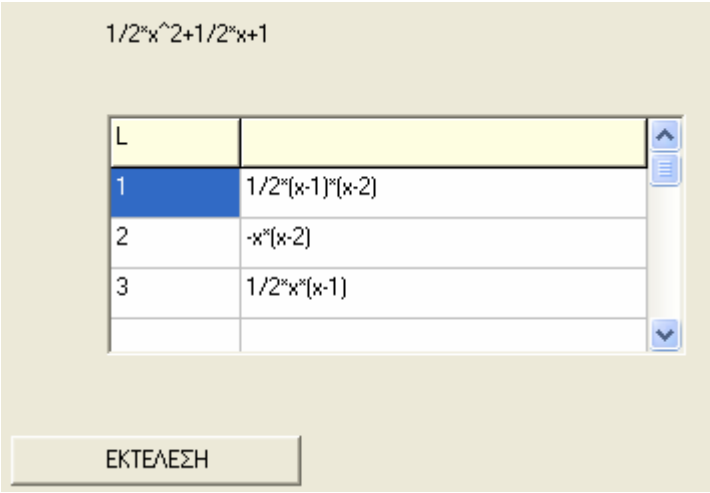



Εικόνα 4.1 Εισαγωγή τιμών $x_i$ και $f(x_i)$



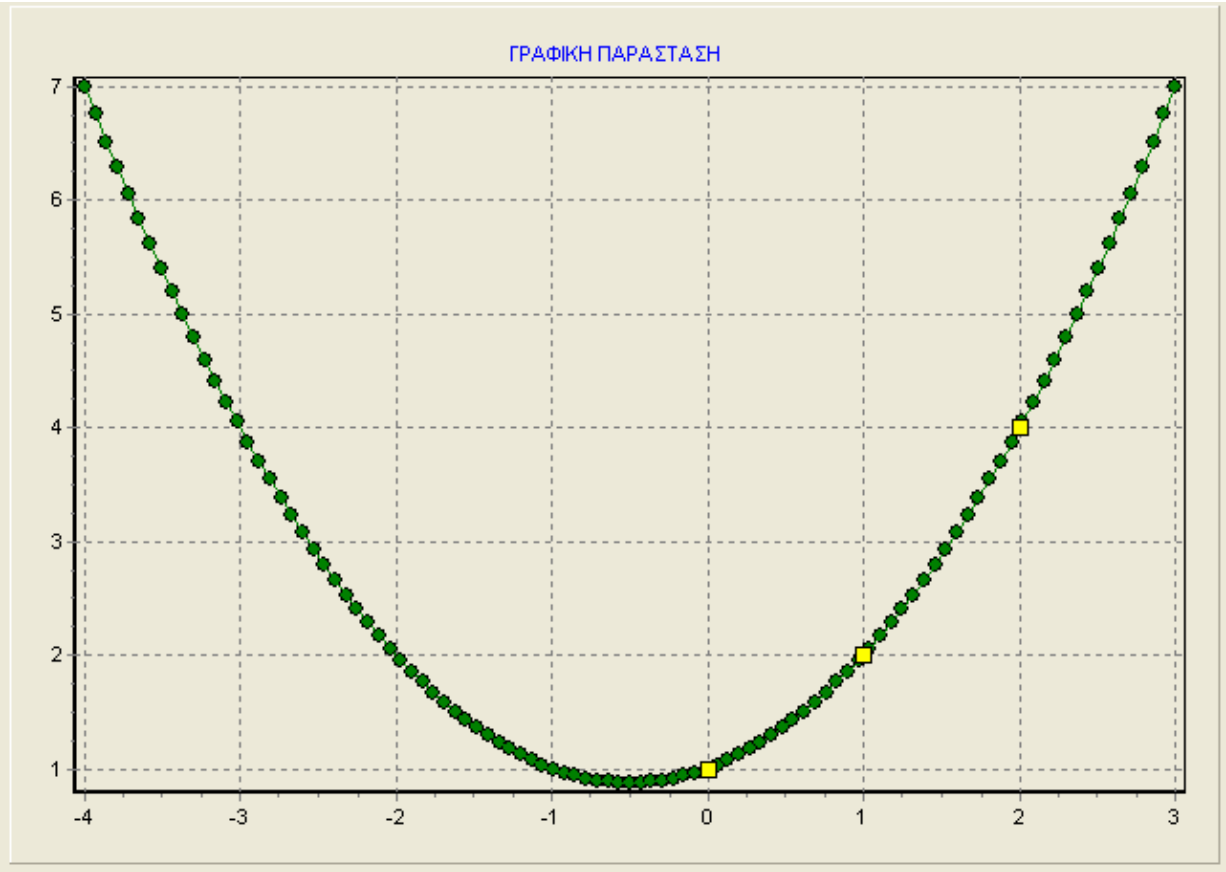

Εικόνα 4.2 Γραφική αναπαράσταση των ζεύγων τιμών $(x_i, f(x_i))$

Μετά την εισαγωγή των τιμών, ο χρήστης πατάει το κουμπί "ΕΚΤΕΛΕΣΗ" και εμφανίζεται το πολυώνυμο το οποίο προσδιορίστηκε με τη μέθοδο Lagrange, καθώς και οι τιμές των αντίστοιχων πολυωνύμων $L_i$ (Εικ. 4.3).




Εικόνα 4.3 Το πολυώνυμο παρεμβολής και τα πολυώνυμα $L_i$

Ορίζοντας τα επιθυμητά όρια, και πατώντας το κουμπί "ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ", το πολυώνυμο αναπαρίσταται γραφικά (Εικ. 4.4)




Εικόνα 4.4 Η γραφική παράσταση του πολυωνύμου που προσδιορίστηκε

## 4.2 Μέθοδος Ελαχίστων Τετραγώνων

Η μέθοδος των Ελαχίστων Τετραγώνων χρησιμοποιείται συνήθως όταν θέλουμε να μελετήσουμε μεγάλο αριθμό σημείων, όπου η προσέγγιση τους με συμπτωτικό πολυώνυμο είναι δύσκολη τις περισσότερες φορές. Σ' αυτή την περίπτωση προσπαθούμε να βρούμε ένα πολυώνυμο *P(x)* ή γενικότερα μια συνάρτηση *f(x)* με συμπεριφορά παρόμοια με αυτή της ομάδας των σημείων που διαθέτουμε $(x_i , y_i)$, $i=1,\ldots n$ και για την οποία η διαφορά που δίνεται από την εξίσωση 4.1 να είναι ελάχιστη.

$$S = \sum_{i=0}^{n} \left[ y_i - P(x_i) \right]^2 \qquad 4.1$$

Στην πράξη, ζητάμε την ελαχιστοποίηση του αθροίσματος $P(x) = ax + b$ όπου *P(x)* το ζητούμενο πολυώνυμο συγκεκριμένου βαθμού.

Οι συντελεστές *a* και *b* του πολυωνύμου προσδιορίζονται από τις σχέσεις 4.1 και 4.2 :

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \qquad 4.1$$

$$b = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} x_i}{n\sum_{i=1}^{n} x_i^2 - \left( \sum_{i=1}^{n} x_i \right)^2} \qquad 4.2$$

### 4.2.1 Παράδειγμα υπολογισμού πολυωνύμου με τη χρήση της μεθόδου των Ελαχίστων Τετραγώνων

Με δεδομένο τον πίνακα τιμών 4.1, ζητείται να υπολογισθεί το πολυώνυμο $P(x) = ax + b$ που προσδιορίζει καλύτερα την ακολουθία των σημείων του πίνακα 4.1:

| x | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| y | 5,3 | 7,5 | 8,7 | 10,6 | 13 | 14,7 |

Πίνακας 4.1

Αρχικά συμπληρώνεται ο πίνακας 4.2 με τα $x_i\, y_i$, $x_i^2$, $y_i^2$ και υπολογίζονται τα αθροίσματα $\sum_{i=1}^{n} x_i$ , $\sum_{i=1}^{n} y_i$ , $\sum_{i=1}^{n} x_i y_i$ , $\sum_{i=1}^{n} x_i^2$ , $\sum_{i=1}^{n} x_i^2$ που χρησιμοποιούνται για τον υπολογισμό των συντελεστών *a* και *b* του ζητούμενου πολυωνύμου.

| $x_i$ | $y_i$ | $x_i\, y_i$ | $x_i^2$ | $y_i^2$ |
|---|---|---|---|---|
| 0 | 5,3 | 0 | 0 | 28,09 |
| 1 | 7,5 | 7,5 | 1 | 56,25 |
| 2 | 8,7 | 17,4 | 4 | 75,69 |
| 3 | 10,6 | 31,8 | 9 | 112,36 |
| 4 | 13 | 52 | 16 | 169 |
| 5 | 14,7 | 73,5 | 25 | 216,09 |
| $\sum_{i=1}^{n} x_i = 15$ | $\sum_{i=1}^{n} y_i = 59{,}8$ | $\sum_{i=1}^{n} x_i y_i = 182{,}2$ | $\sum_{i=1}^{n} x_i^2 = 55$ | $\sum_{i=1}^{n} y_i^2 = 657{,}48$ |

Πίνακας 4.2

Με αντικατάσταση των αθροισμάτων στις σχέσεις 4.1 και 4.2 , προκύπτει ότι:

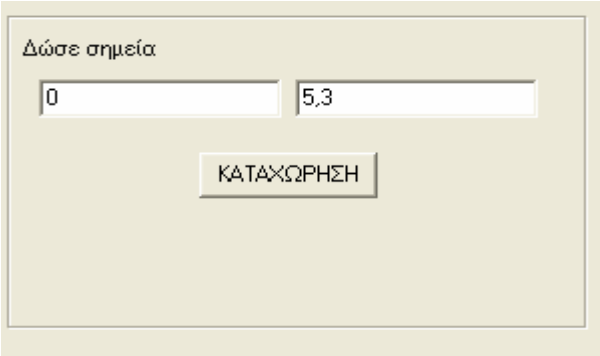
$$a = \frac{6 \cdot 182{,}2 - 15 \cdot 59{,}8}{6 \cdot 55 - 15^2} = \frac{196{,}2}{105} = 1{,}8685 \approx 1{,}87$$

$$b = \frac{55 \cdot 59{,}8 - 182{,}2 \cdot 15}{6 \cdot 55 - 15^2} = \frac{556}{105} = 5{,}2952 \approx 5{,}3$$

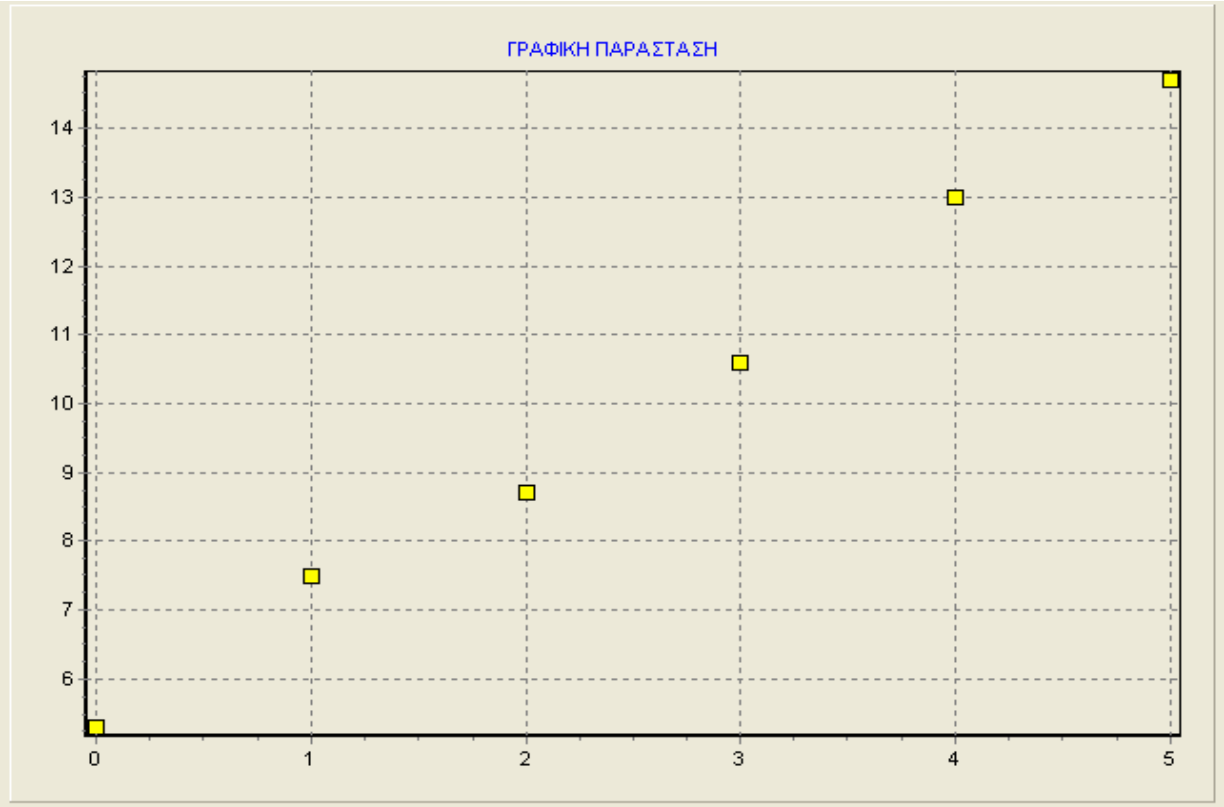
Άρα το ζητούμενο πολυώνυμο είναι το $P(x) = 1{,}87x + 5{,}3$

Με χρήση της εφαρμογής της πτυχιακής εργασίας, η μέθοδος εκτελείται με τα εξής βήματα:

Αρχικά δίνονται από το χρήστη οι συντεταγμένες των σημείων από τα οποία θα προκύψει η ευθεία των Ελαχίστων Τετραγώνων (Εικ. 4.5), πατώντας κάθε φορά το κουμπί "ΚΑΤΑΧΩΡΗΣΗ" ώστε να αναπαριστώνται γραφικά τα δεδομένα σημεία (Εικ.4.6).

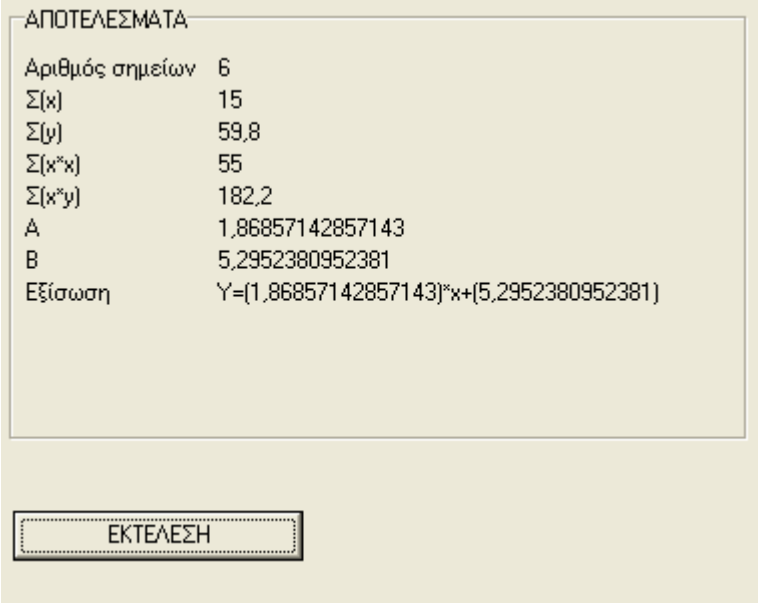



Εικόνα 4.5 Εισαγωγή των τιμών $x_i$ , $y_i$

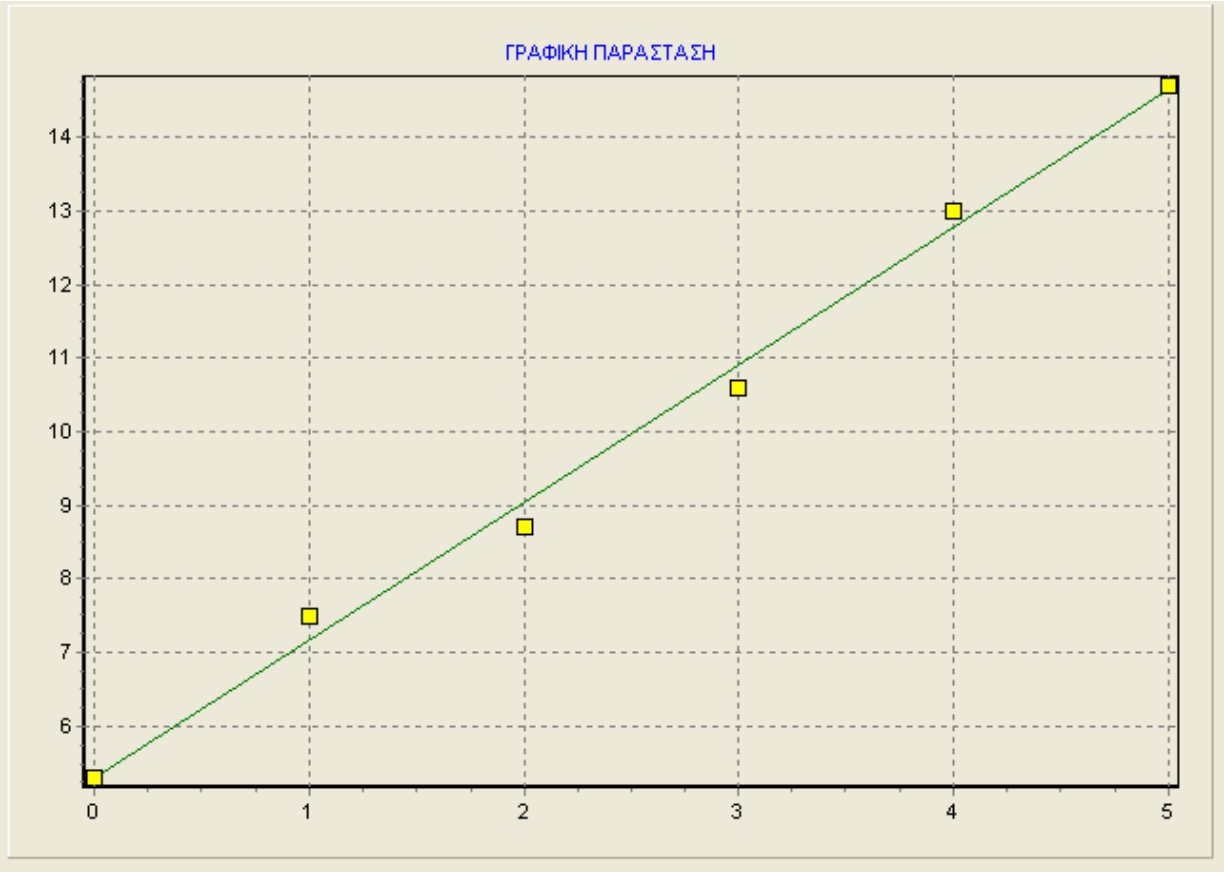



Εικόνα 4.6 Γραφική αναπαράσταση των δεδομένων σημείων $(x_i , y_i)$

Πατώντας το κουμπί "ΕΚΤΕΛΕΣΗ" εμφανίζεται η κάρτα των αποτελεσμάτων (Εικ 4.7) που περιλαμβάνει τα αθροίσματα, τις τιμές των συντελεστών $a$ και $b$ και την ευθεία που προέκυψε, καθώς και η γραφική της παράσταση (Εικ. 4.8).




Εικόνα 4.7 Αποτελέσματα της μεθόδου Ελαχίστων Τετραγώνων




Εικόνα 4.8 Γραφική παράσταση της ευθείας των Ελαχίστων Τετραγώνων, η οποία προσδιορίζει καλύτερα τα ζεύγη τιμών $(x_i , y_i)$

## Παράρτημα

Μετά την επίλυση τη χρήση των μεθόδων Τραπεζίου και Simpson, εάν ο χρήστης θέλει να επαναχρησιμοποιήσει την ίδια ή κάποια άλλη μέθοδο, πρέπει πρώτα να βγαίνει από την εφαρμογή, γιατί κρατούνται οι προηγούμενες παράμετροι που χρησιμοποιούνται για τη σχεδίαση των γραφικών παραστάσεων.

Η εισαγωγή των συναρτήσεων γίνεται με τρόπο παρόμοιο με αυτόν στο Matlab ή σε άλλες μορφές κώδικα

Π.χ η εξίσωση $3x^2 - 5x + \cos(x) - \sqrt{2x}$, γράφεται ως $3*x^2-5*x+\cos(x)-\mathrm{sqrt}(2*x)$

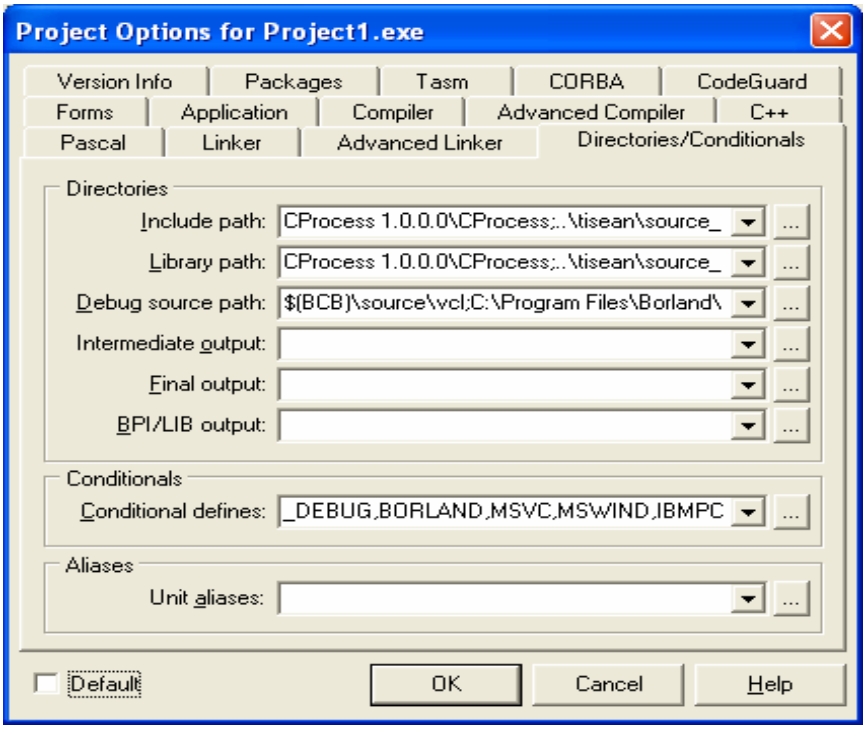
Συναρτήσεις που περιέχουν το π=3,14, το π εισάγεται ως pi, ή ως π, που μετατρέπεται αυτόματα σε pi.

Η μαθηματική σταθερά e εισάγεται ως exp.

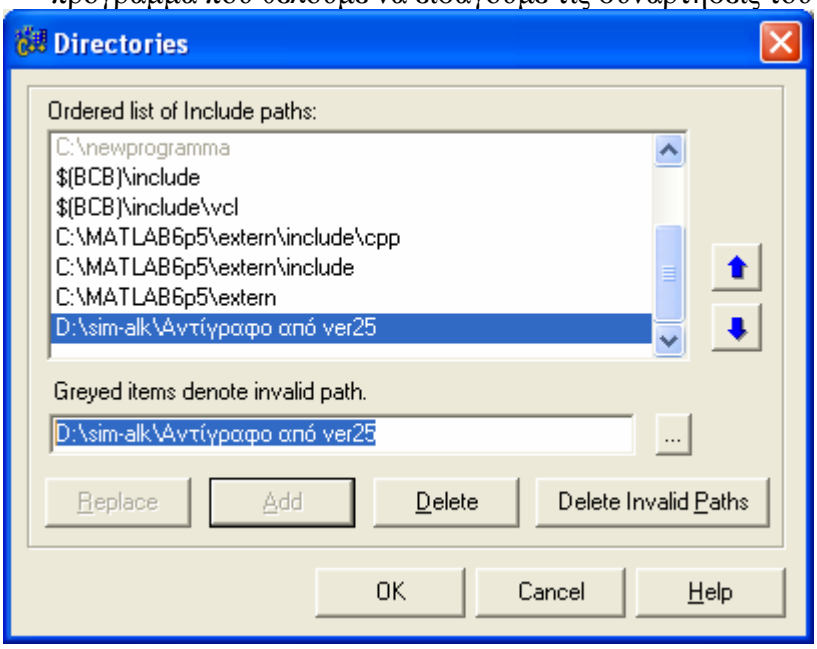
**Απαιτήσεις συστήματος:** Για να τρέξει η εφαρμογή, πρέπει να είναι εγκατεστημένο το Matlab (version 6.5 +) και η γλώσσα προγραμματισμού C++ (version 6+). Επίσης, λόγω χρήσης συναρτήσεων του Matlab από την εφαρμογή, χρειάζεται αυτές να εισαχθούν στη C++ σύμφωνα με τις εξής οδηγίες:

# ΕΙΣΑΓΩΓΗ ΣΥΝΑΡΤΗΣΕΩΝ ΤΟΥ MATLAB ΣΕ C++

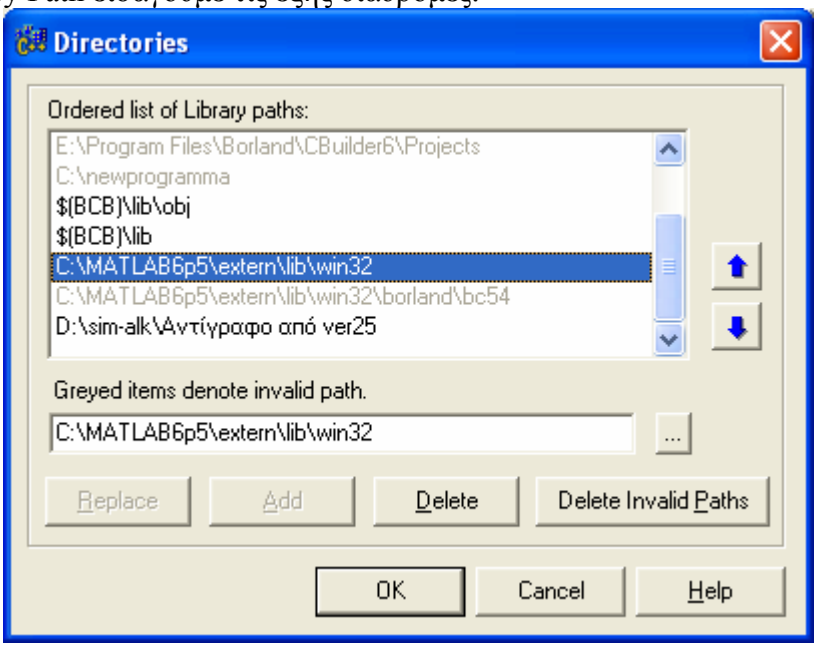
1. Από το μενου της Borland επιλέγουμε Project→Options
2. Στην Καρτέλα Directories/Conditional




3. Στο Include Path εισάγουμε τις εξής διαδρομές:
   - C:\MATLAB6p5\extern\include\cpp
   - C:\MATLAB6p5\extern\include
   - C:\MATLAB6p5\extern
   - D:\sim-alk\ver25 Αντιστοιχεί στην διαδρομή στην οποία βρίσκεται το πρόγραμμα που θέλουμε να εισάγουμε τις συναρτήσεις του Matlab.

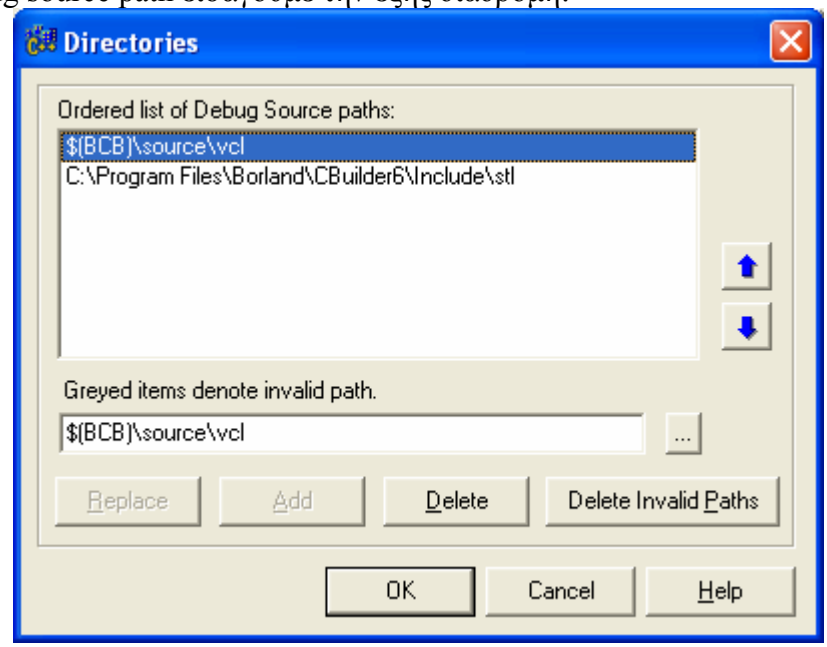


4. Στο Library Path εισάγουμε τις εξής διαδρομές:



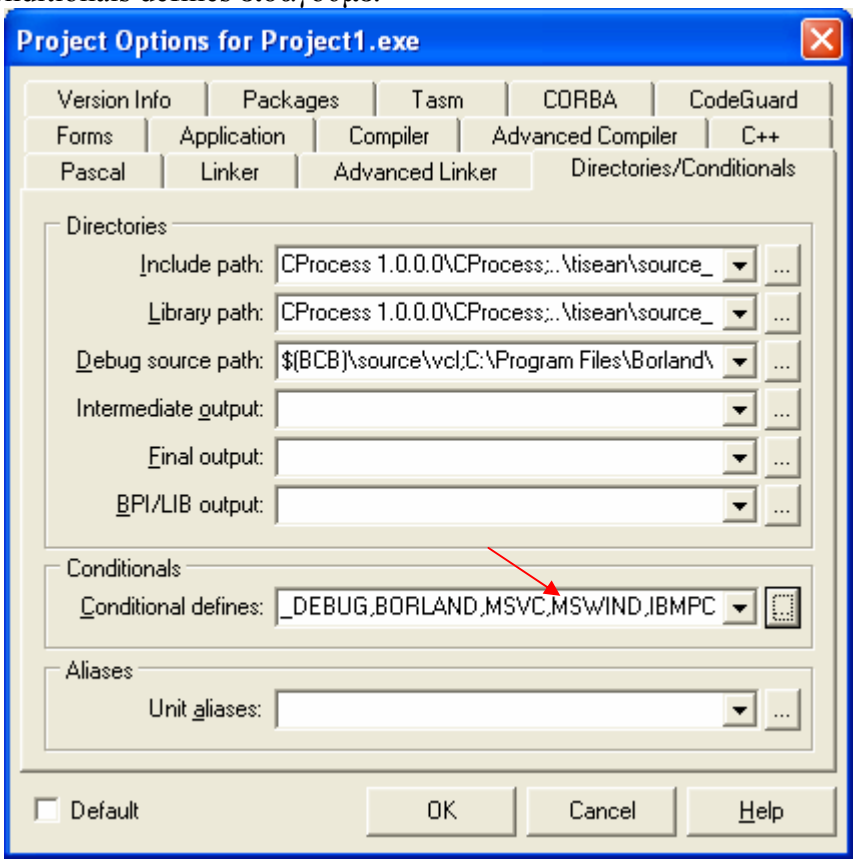

- C:\MATLAB6p5\extern\lib\win32
- C:\MATLAB6p5\extern\lib\win32\borland\bc54
- D:\sim-alk\ ver25 Αντιστοιχεί στην διαδρομή στην οποία βρίσκεται το πρόγραμμα που θέλουμε να εισάγουμε τις συναρτήσεις του Matlab.

5. Στο Debug source path εισάγουμε την εξής διαδρομή:



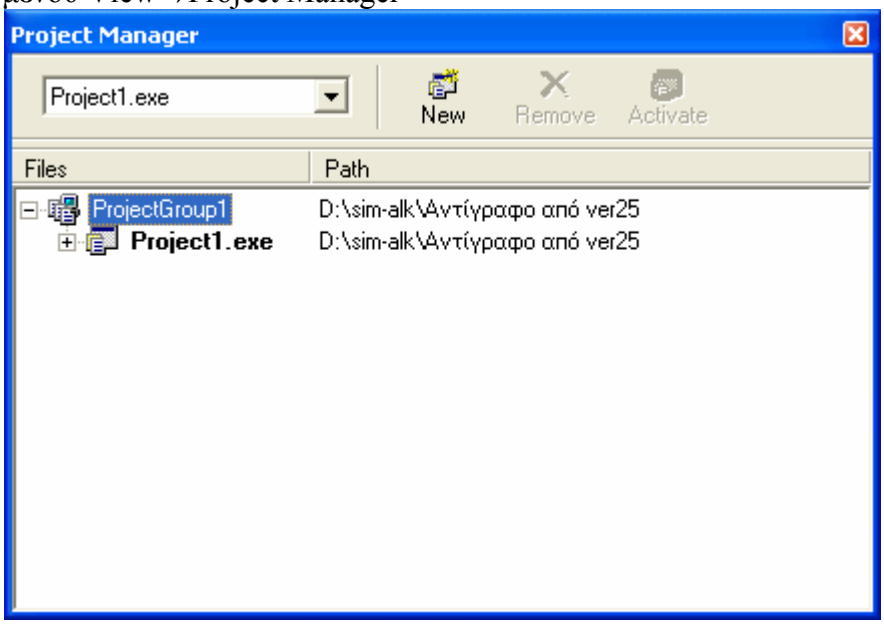

- C:\Program Files\Borland\CBuilder6\Include\stl

6. Στο Conditionals defines εισάγουμε:

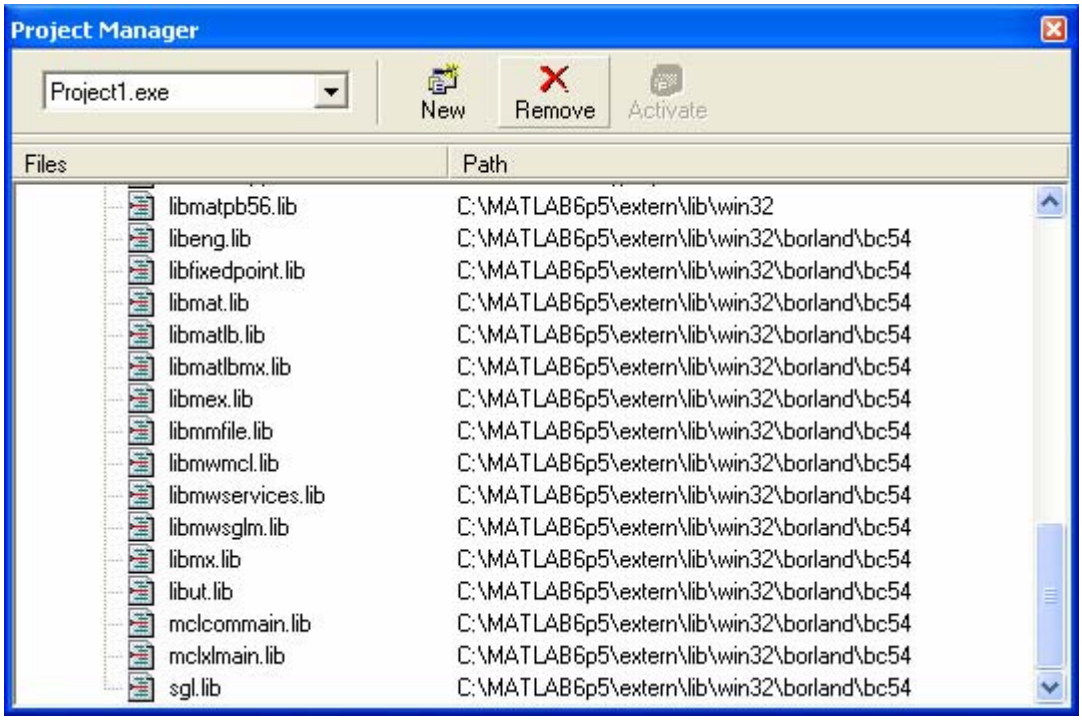



_DEBUG,BORLAND,MSVC,MSWIND,IBMPC

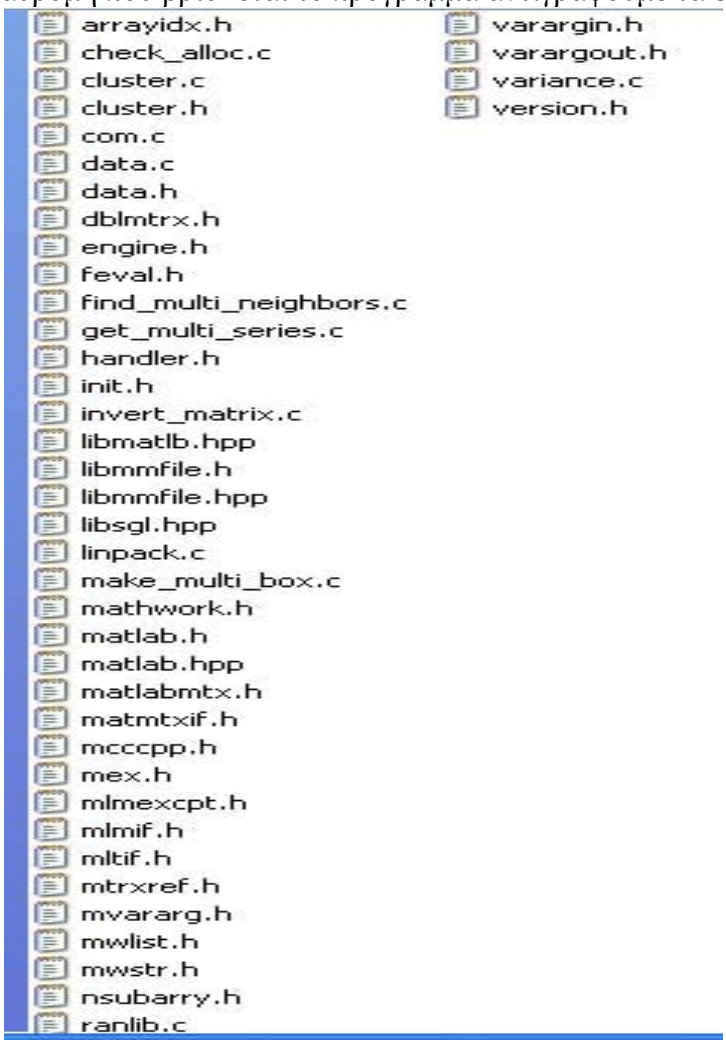
7. Από το μενού View→Project Manager




Κάνουμε δεξί κλικ στο Project1.exe, επιλέγουμε Add και εισάγουμε τις παρακάτω βιβλιοθήκες (φαίνονται στην παρακάτω εικόνα) , που βρίσκονται στην διαδρομή C:\MATLAB6p5\ extern\lib\win32\borland\bc54 και C:\MATLAB6p5\ extern\lib\win32\libmatpb56lib.




8. Μέσα στην διαδρομή που βρίσκεται το πρόγραμμα αντιγράφουμε τα εξής αρχεία:



9. Τα παραπάνω αρχεία πρέπει να τα εισάγουμε με την ίδια διαδικασία όπως το βήμα 7.
10. Στο αρχείο mltif.h θα πρέπει οι μεγάλες γραμμές που έχει να σπάσουν σε μικρότερες.
11. Πηγαίνουμε στον φάκελο … και εκτελούμε τα αρχεία … και … για να ενεργοποιηθεί το runtime του Matlab για ton Builder C++
12. Προσοχή!!!! Στο παράθυρο του Matlab θα πρέπει να έχουμε ήδη μεταφερθεί στον φάκελο που είναι τα αρχεία με τις συναρτήσεις του Matlab αλλιώς δεν τις βλέπει για τις εκτελέσει δηλ. cd d:/ver25 και cd ./Matm

# Μέθοδος Müller

Αν θέλαμε να ερμηνεύσουμε γεωμετρικά τη μέθοδο Secant θα βλέπαμε ότι σε κάθε βήμα φέρουμε την ευθεία που διέρχονται από τις εικόνες δύο σημείων π.χ. x0 και x1. Το επόμενο σημείο της προσέγγισης x2 είναι η τομή της ευθείας με τον άξονα των x. Η διαδικασία επαναλαμβάνεται για τα σημεία x1 και x2.
Αν αντί για δύο δοθέντα σημεία σε κάθε βήμα είχαμε τρία, π.χ. x0, x1 και x2, τότε θα μπορούσαμε να φέρουμε από τις εικόνες τους μία παραβολή. Το επόμενο σημείο της προσέγγισης (το x3) είναι η τομή της παραβολής με τον άξονα των x. Στο επόμενο βήμα, η διαδικασία επαναλαμβάνεται για τα σημεία x1, x2 και x3.
Η μέθοδος που περιγράψαμε ονομάζεται μέθοδος του Müller και ο επαναληπτικός της τύπος δίνεται από τους ακόλουθους τύπους:
Στην κ επανάληψη η προσεγγιστική εκτίμηση της ρίζας υπολογίζεται ως ρίζα της παραβολής που διέρχεται από τα σημεία

$$\big(x(k-2),\, y(k-2)\big), \big(x(k-1),\, y(k-1)\big), \big(x(k),\, y(k)\big)$$

$$x(k+1) = x(k) - \frac{2y(k)}{s + sign(s)\cdot\sqrt{s^2 - 4\cdot y(k)\cdot d_1}}$$

Όπου

*sign(s)* η συνάρτηση “πρόσημο” με όρισμα το *s*

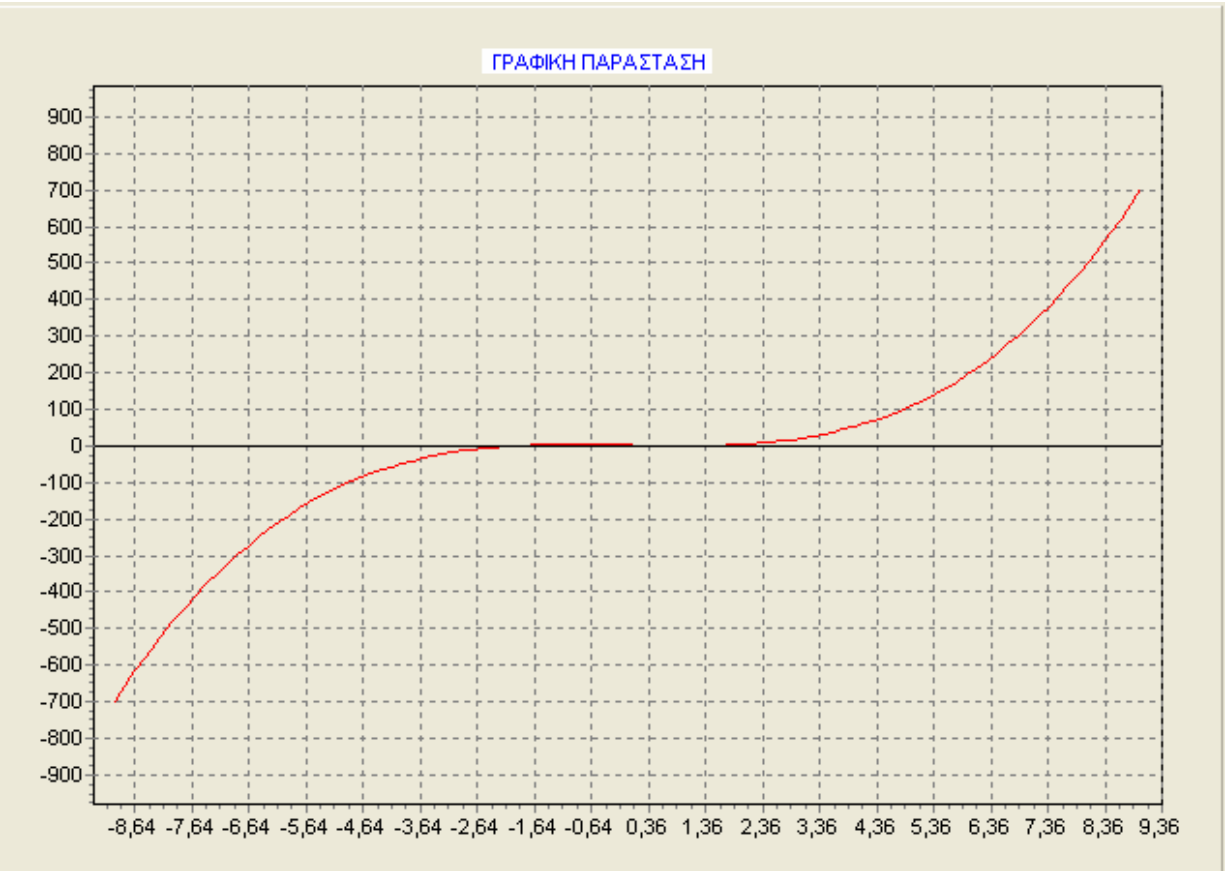
$$c_1 = \frac{y(k-1) - y(k-2)}{x(k-1) - x(k-2)}$$

$$c_2 = \frac{y(k) - y(k-1)}{x(k) - x(k-1)}$$

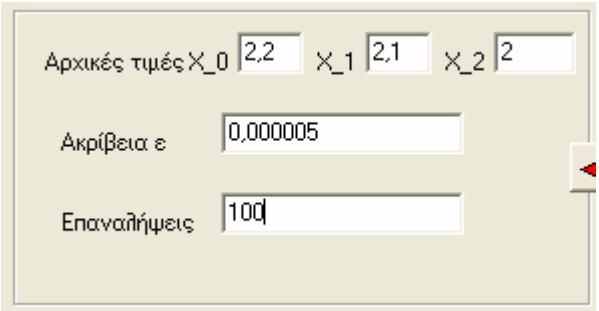
$$d_1 = \frac{c_2 - c_1}{x(k) - x(k-2)}$$

### Παράδειγμα της μεθόδου Muller

Με δεδομένη την εξίσωση $f(x) = 2 - 3x + x^3$
τα αρχικά σημεία x0 = 2,2 x1= 2,1 και x2 = 2, και με ακρίβεια 6 δεκαδικών ψηφίων, η μέθοδος εκτελείται ως εξής

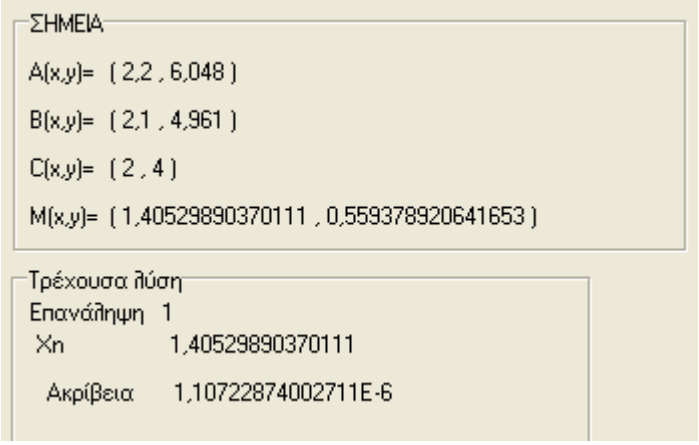



Η γραφική παράσταση της συνάρτησης




Εισαγωγή των παραμέτρων

Μετά την πρώτη επανάληψη τα αποτελέσματα φαίνονται στους παρακάτω πίνακες



Μετά από 8 επαναλήψεις, η ρίζα βρέθηκε με την επιθυμητή ακρίβεια:

ΣΗΜΕΙΑ

A(x,y)= ( 1,00062295794333 , 1,16447155296129E-6 )

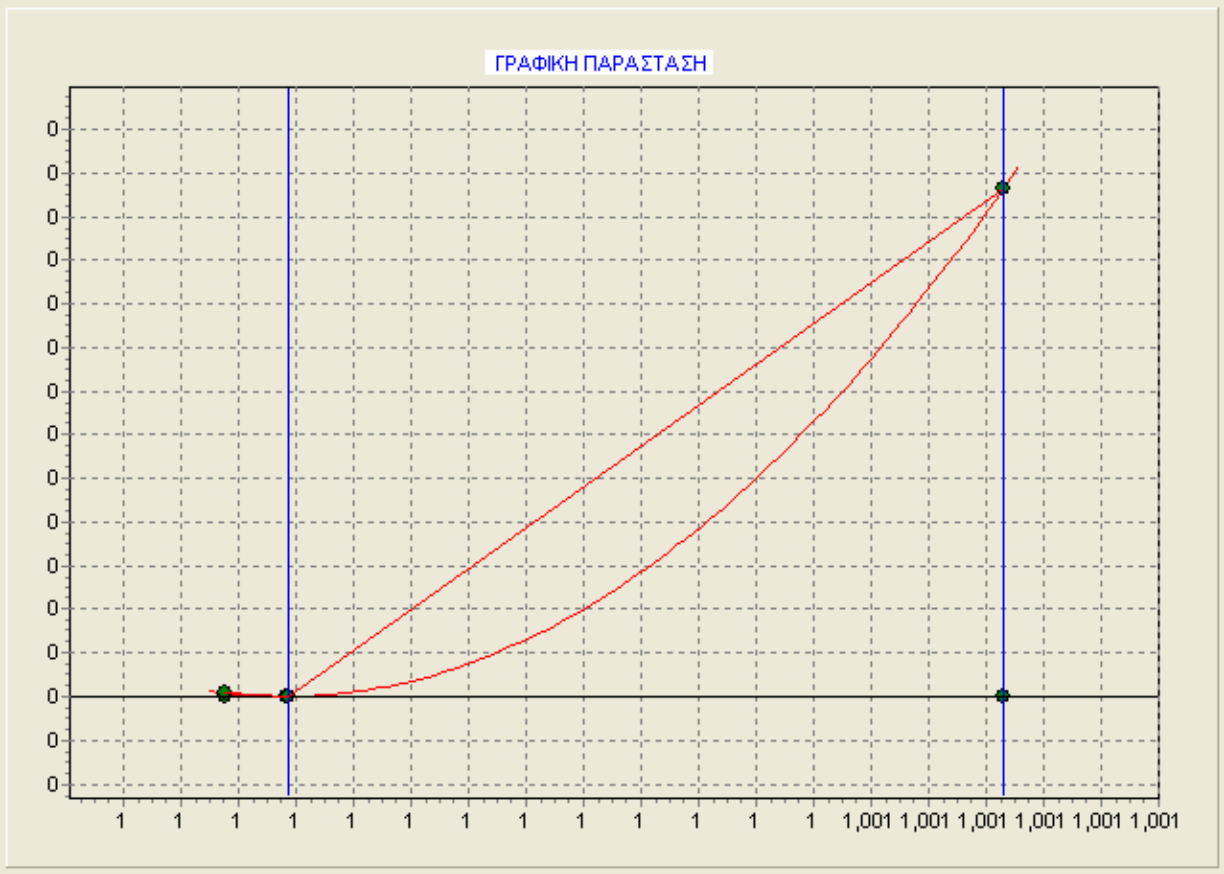
B(x,y)= ( 0,999947088002046 , 8,3988906940391E-9 )

C(x,y)= ( 1,00000111281746 , 3,7150282850007E-12 )

M(x,y)= ( 1,00000000558872 , -2,22044604925031E-16 )

Τελική λύση

| | |
|---|---|
| Επανάληψη | 8 |
| Ρίζα εξίσωσης | 1,00000000558872 |
| Ακρίβεια | 1,10722874002711E-6 |




Η γραφική παράσταση της συνάρτησης μετά την εύρεση της ρίζας της

**Κώδικας**

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "math.h"
#include "methodoi.h"
#include <engine.h>
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
#pragma comment( lib, "Libmx.lib"  )
#pragma comment( lib, "libmex.lib" )
#pragma comment( lib, "libeng.lib" )
TForm1 *Form1;
Methodoi methodoi;
Engine *m_pEngine;
//---------------------------------------------------------------------------
//apotelesma sinartisis
double TForm1::f(double x,AnsiString sinartisi)
{
	AnsiString xx=FloatToStr(x);
	for(int i=1;i<=xx.Length();i++)
		if(xx[i]==',')
			xx[i]='.';
	AnsiString sin=sinartisi;
	for(int i=1;i<=sin.Length();i++)
		if(sin[i]==',')
			sin[i]='.';
	engEvalString(m_pEngine,"clear");
	engEvalString(m_pEngine,xx.c_str());
	engEvalString(m_pEngine,"x=ans");
	engEvalString(m_pEngine,sin.c_str());
	engEvalString(m_pEngine,"f=ans");

	double *cresult;
	mxArray *mresult;
	mresult = engGetVariable(m_pEngine,"f");
	cresult = mxGetPr(mresult);
	return cresult[0];

}
//---------------------------------------------------------------------------
//euresi paragogou tis sinartisis stin parametro
double TForm1::diff(double x,AnsiString sinartisi)
{
	double par=0.0;
	AnsiString xx=FloatToStr(x);
```

```
        for(int i=1;i<=xx.Length();i++)
                if(xx[i]==',')
                        xx[i]='.';
        engEvalString(m_pEngine,"clear");
        engEvalString(m_pEngine,xx.c_str());
        engEvalString(m_pEngine,"x=ans");
        engEvalString(m_pEngine,sinartisi.c_str());
        engEvalString(m_pEngine,"ff=ans");
        engEvalString(m_pEngine,"par1=diff(ff)");
        engEvalString(m_pEngine,"par2=char(par1)");
        engEvalString(m_pEngine,"par=eval(par2,x)");

        double *cresult;
        mxArray *mresult;
        mresult = engGetVariable(m_pEngine,"par");
        cresult = mxGetPr(mresult);
        par=cresult[0];
        return par;
}
//---------------------------------------------------------------------------
//kanei tin dieresi toy arithmiti me ton paronomasti
//stin methodo la grange kai epistrefei to apotelesma
AnsiString TForm1::dieresi(AnsiString ar, AnsiString par)
{
        engEvalString(m_pEngine,"clear");
        engEvalString(m_pEngine,"syms x");
        engEvalString(m_pEngine,ar.c_str());
        engEvalString(m_pEngine,"ar=ans");
        engEvalString(m_pEngine,par.c_str());
        engEvalString(m_pEngine,"par=ans");
        engEvalString(m_pEngine,"ap=ar/par");
        engEvalString(m_pEngine,"apot=char(ap)");

        AnsiString
teliko="",ap="@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@";
        mxArray *mresult;
        mresult = engGetVariable(m_pEngine,"apot");
        mxGetString(mresult,ap.c_str(),100000000);

        int point=0;
        for(int i=1;i<=ap.Length();i++)
        {
                if(ap[i]=='@'||ap[i]==' ')
                {
                        point=i;
                        break;
                }
        }
        for(int i=1;i<point;i++)
```

```
	{
		teliko+=ap[i];
	}
	teliko.SetLength(teliko.Length()-1);
	return teliko;
}
//---------------------------------------------------------------------------
//kanei tin aplopioisi tis sinartisis kai tin epistrefei
AnsiString TForm1::aplopiisi(AnsiString sin)
{
	engEvalString(m_pEngine,"clear");
	engEvalString(m_pEngine,"syms x");
	engEvalString(m_pEngine,sin.c_str());
	engEvalString(m_pEngine,"ap=ans");
	engEvalString(m_pEngine,"ap=expand(ans)");
	engEvalString(m_pEngine,"apot=char(ap)");

	AnsiString
ap="@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@";
	mxArray *mresult;
	mresult = engGetVariable(m_pEngine,"apot");
	mxGetString(mresult,ap.c_str(),100000000);
	return ap;
}
//---------------------------------------------------------------------------
//sxediasi grafikis me ta oria tvn parametron
void TForm1::makeGraf(double left,double right,int simeia, TLineSeries *ser)//apo pou
xekinaei,
//pou telionei, kai to plithos ton simion tis grafikis parastasis
{
	//sbisimo tis palias grafikis
	ser->Clear();
	//simeia stin nea grafiki
	double bima=(right-left)/simeia;//se posa ypodiastimata
	//xwrizete i grafiki parastasi
	for(double i=(left-(2*bima));i<=(right+(2*bima));i+=bima)//apo ligo prin to proto simeio
	//mexri ligo meta to telefteo simeio
	{
		//prostheto ena ena ta simeia stin grammi
		double y=this->f(i,Form1->LabelSinartisi->Caption);//ypologizetai
		//i sintetagmeni y kalontas ti synartisi f
		ser->AddXY(i,y);//topothetisi ton simion sti grafiki
		Application->ProcessMessages();
	}
}
//---------------------------------------------------------------------------
//ftiaxnei ta oria sthn grafiki
void TForm1::makeBordersGraf(double a,double b,double ya,double yb, TChart *chart)
{
```

```
	//orizontios aksonas +- 1/5 apo tin apostasi ton simeion
	double min=0.0,max=0.0;//dilosi tou aristerou(min) kai tou dexiou oriou(max)
	double diff=(b-a)/5;//ypologismow toy 1/5 tis apostasis tou protoy
	//me to telefteo simeio tis grafikis
	min=a-diff;//to aristero orio einai kata 1/5 aristera apo to proto simeio
	max=b+diff;// to dexi orio einai kata 1/5 dexia apo to telefteo simeio
	chart->BottomAxis->Minimum=min;//orismos tou min os aristero orio
	//ston orizontio aksona
	chart->BottomAxis->Maximum=max;//orismos tou max os dexi orio
	//ston orizontio aksona
	//kathetos aksonas +- 1/5 apo tin apostasi ton simeion
	min=0.0,max=0.0;//dilosi tou kato oriou (min kai tou pano  oriou (max)
	max=0.0;
	if(ya<yb)//an to y tou protou simeiou einai mikrotero (pio kato)apo to y
	//toy telefteou simeiou
	{
		diff=(yb-ya)/5;//to 1/5 tis postasis tha einai (yb-ya)/5
		min=ya-diff;//to kato orio einai kata 1/5 pio kato apo to proto simeio
		max=yb+diff;//to pano orio einai kata 1/5 pio pano apo to telefteo simeio
	}
	else //allios an to y toy protou simeiou einai pio pano apo to y
	//tou telefteou simeiou
	{
		diff=(ya-yb)/5; //to 1/5 tis apostasis tha einai (ya-yb)/5
		min=yb-diff;//to kato orio einai kata 1/5 pio kato apo to telefteo simeio
		max=ya+diff;//to pano orio einai kata 1/5 pio pano apo to proto simeio
	}
	chart->LeftAxis->Minimum=min;//orismos tou min os katheto orio ston
	//katheto aksona
	chart->LeftAxis->Maximum=max;//orismos tou max os pano orio
	//ston katheto aksona

}
//---------------------------------------------------------------------------
//bazei tis dyo kathetes apo ta simeia a kai b
void TForm1::makeOutBorders(double a,double b,double min,double max,double min2,double
max2)
{
	Form1->Series3->Clear();//arxika svinetai i aristeri kathetos
	Form1->Series3->AddXY(a,max);//topothetisi tis kathetis grammis apo to
	//telefteo simeio tis grafikis mexri to pano orio
	Form1->Series3->AddXY(a,min); //totpothetisi tis kathetis grammis apo to proto
	//simeio tis grafikis mexri to kato orio
	Form1->Series4->Clear(); //arxika svinetai i dexia kathetos
	Form1->Series4->AddXY(b,max);//topothetisi tis kathetis grammis apo to
	//telefteo simeio tis grafikis mexri to pano orio
	Form1->Series4->AddXY(b,min);//totpothetisi tis kathetis grammis apo to proto
	//simeio tis grafikis mexri to kato orio
	Form1->Series5->Clear();//
	Form1->Series5->AddXY(max2,0);
	Form1->Series5->AddXY(min2,0);
```

```
}
//---------------------------------------------------------------------------
//sunartisi pou metatrepei tin teleia dekasdikvn psifion se koma
void TForm1::kommaTelia(TEdit *edit)//++++++++++++++++++++++
{
	AnsiString ed="";//dilosi kenou string
	for(int i=1;i<=edit->Text.Length();i++)//kato apo tin arxi mexri to telos tou
	//keimenou sto plaisio tis synartisis
	{
		if(edit->Text[i]=='.')//an enas xaraktiras einai teleia
			ed+=',';	//na ginei koma
		else
			ed+=edit->Text[i];//allios na prostetehi sto ed
	}
	edit->Text=ed;//to keimeno sto plaisio einai to string ed
}
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
	: TForm(Owner)
{
}
//---------------------------------------------------------------------------
//bimatiki ektelesi
void __fastcall TForm1::ButtonBimaClick(TObject *Sender)
{
	if(!this->GbDixotomisi->Visible&&!this->GbNewtonRaphson->Visible&&!
	  this->GbRegulaFalsi->Visible&&!this->GbSimspon->Visible&&!
	  this->GbTemnousa->Visible&&!this->GbTrapexiou->Visible)
	{
		ShowMessage("ΔΕΝ ΕΧΕΤΕ ΕΠΙΛΕΞΕΙ ΜΕΘΟΔΟ");
	}
	else
	{
		//apenergopoiisi btn
		this->ButtonBima->Enabled=false;
		this->ButtonEnd->Enabled=false;
		//åðáíÜëçøç
		int a=this->LabelRow->Caption.ToInt();
		a++;
		this->LabelRow->Caption=a;
		//topothetisi ton a kai b ston pinaka
		this->SgValuesGramEks->Cells[1][a]=this->LabelNewA->Caption;
		this->SgValuesGramEks->Cells[2][a]=this->LabelNewB->Caption;
		if(a==1)
		{
			//i grafiki mia fora stin arxi
			this->ButtonGrafiki->Visible=false;
			this->LabelWait->Visible=false;
			//energopoiisi tou koumpiou vimatiki ektelesi
			this->GbBimaGramEks->Visible=true;
			this->GbFinalGramEks->Visible=false;
```

```
        //epikefalides ston pinaka analoga me ti methodo
        this->SgValuesGramEks->Cells[0][0]="ΕΠΑΝΑΛΗΨΕΙΣ";
        this->SgValuesGramEks->Cells[1][0]="X_A";
        this->SgValuesGramEks->Cells[2][0]="X_B";
        this->SgValuesGramEks->Cells[3][0]="X_M";
        this->SgValuesGramEks->Cells[4][0]="Y_M";
        this->SgValuesGramEks->Cells[0][1]=1;
        this->SgNewton->Cells[0][0]="ΕΠΑΝΑΛΗΨΕΙΣ";
        this->SgNewton->Cells[1][0]="Xo";
        this->SgNewton->Cells[2][0]="Yo";
        //
        if(this->GbDixotomisi->Visible)
        {
            this->LabelNewA->Caption=this->EditDixotomisiA->Text;
            this->LabelNewB->Caption=this->EditDixotomisiB->Text;
        }
        else if(this->GbRegulaFalsi->Visible)
        {
            this->LabelNewA->Caption=this->EditRegulaFalsiA->Text;
            this->LabelNewB->Caption=this->EditRegulaFalsiB->Text;
        }
        else if(this->GbTemnousa->Visible)
        {
            this->LabelNewA->Caption=this->EditTemnousaA->Text;
            this->LabelNewB->Caption=this->EditTemnousaB->Text;
        }
    }
    //topothetisi timon ston pinaka analoga me timethodo pou epilexthike
    if(this->GbDixotomisi->Visible)
    {
        //topothetisi ton orion a kai b sta prota kelia
        this->SgValuesGramEks->Cells[1][1]=this->EditDixotomisiA->Text;
        this->SgValuesGramEks->Cells[2][1]=this->EditDixotomisiB->Text;
        this->SgValuesGramEks->Visible=true;//o pinakas ginetai oratos
        this->GbPoints->Visible=true;//fenontai se kathe epanalipsi ta simeia
        //Á, Â, kai to meso Ì
        methodoi.dixotomisi(); //klisi tis methodou
    }
    else if(this->GbRegulaFalsi->Visible)
    {
        //topothetisi ton orion a kai b sta prota kelia
        this->SgValuesGramEks->Cells[1][1]=this->EditRegulaFalsiA->Text;
        this->SgValuesGramEks->Cells[2][1]=this->EditRegulaFalsiB->Text;
        this->SgValuesGramEks->Visible=true;
        this->GbPoints->Visible=true;
        methodoi.regulafalsi();
    }
    else if(this->GbTemnousa->Visible)
    {
        this->SgValuesGramEks->Cells[1][1]=this->EditTemnousaA->Text;
        this->SgValuesGramEks->Cells[2][1]=this->EditTemnousaB->Text;
```

```
                this->SgValuesGramEks->Visible=true;
                this->GbPoints->Visible=true;
                methodoi.temnousa();
            }
            else if(this->GbNewtonRaphson->Visible)
            {       //topothetisi sto proto keli tou arxikou ×ï
                this->SgNewton->Cells[1][1]=this->EditNewtonXo->Text;
                this->SgNewton->Visible=true;
                this->Chart2->Visible=true;//orato kai to chart2
                methodoi.newton();//klisi tis newton
            }
        }
}
//---------------------------------------------------------------------------
//methodos mexri telos
void __fastcall TForm1::ButtonEndClick(TObject *Sender)
{
        if(!this->GbDixotomisi->Visible&&!this->GbNewtonRaphson->Visible&&!
          this->GbRegulaFalsi->Visible&&!this->GbSimspon->Visible&&!
          this->GbTemnousa->Visible&&!this->GbTrapexiou->Visible)
        {
            ShowMessage("ΔΕΝ ΕΧΕΤΕ ΕΠΙΛΕΞΕΙ ΜΕΘΟΔΟ");
        }
        else
        {
            this->ButtonBima->Enabled=false;
            this->ButtonEnd->Enabled=false;
            this->Label6->Caption="true";  //mexri telos label true
            //sxediazetai i grafiki mia fora stin arxi
            this->ButtonGrafiki->Visible=false;
            this->EditSinartisi->Visible=false;
            this->LabelSinartisi->Caption=this->EditSinartisi->Text;
            this->LabelWait->Visible=false;
            //emfanisi
            this->GbBimaGramEks->Visible=true;
            //arxikes times
            this->SgValuesGramEks->Cells[0][0]="ΕΠΑΝΑΛΗΨΕΙΣ";
            this->SgValuesGramEks->Cells[1][0]="X_A";
            this->SgValuesGramEks->Cells[2][0]="X_B";
            this->SgValuesGramEks->Cells[3][0]="X_M";
            this->SgValuesGramEks->Cells[4][0]="Y_M";
            this->SgValuesGramEks->Cells[0][1]=1;
            this->SgNewton->Cells[0][0]=" ΕΠΑΝΑΛΗΨΕΙΣ ";
            this->SgNewton->Cells[1][0]="Xo";
            this->SgNewton->Cells[2][0]="Yo";

            if(this->SgValuesGramEks->RowCount==2)
            {
                if(this->GbDixotomisi->Visible)
                {
                    this->LabelNewA->Caption=this->EditDixotomisiA->Text;
```

```
                this->LabelNewB->Caption=this->EditDixotomisiB->Text;
            }
            else if(this->GbRegulaFalsi->Visible)
            {
                this->LabelNewA->Caption=this->EditRegulaFalsiA->Text;
                this->LabelNewB->Caption=this->EditRegulaFalsiB->Text;
            }
            else if(this->GbTemnousa->Visible)
            {
                this->LabelNewA->Caption=this->EditTemnousaA->Text;
                this->LabelNewB->Caption=this->EditTemnousaB->Text;
            }
            else if(this->GbTemnousa->Visible)
            {

            }
        }
        //ksekina methodo
        if(this->GbDixotomisi->Visible)
        {
            this->SgValuesGramEks->Visible=true;
            this->GbPoints->Visible=true;
            while(methodoi.dixotomisi()) {}
        }
        else if(this->GbRegulaFalsi->Visible)
        {
            this->SgValuesGramEks->Visible=true;
            this->GbPoints->Visible=true;
            while(methodoi.regulafalsi()) {}
        }
        else if(this->GbTemnousa->Visible)
        {
            this->SgValuesGramEks->Visible=true;
            this->GbPoints->Visible=true;
            while(methodoi.temnousa()) {}
        }
        else if(this->GbNewtonRaphson->Visible)
        {
            this->SgNewton->Visible=true;
            this->Chart2->Visible=true;
            while(methodoi.newton()) {}
        }
    }
}
//---------------------------------------------------------------------------
//arxiki grafiki
void __fastcall TForm1::ButtonGrafikiClick(TObject *Sender)
{
    if(this->EditSinartisi->Text=="")
    {
        ShowMessage("ΔΕΝ ΔΩΣΑΤΕ ΣΥΝΑΡΤΗΣΗ");
```

```
}
else
{
	//έλεγχος για ‘π’ ή 'p' -> 'pi'
	AnsiString start=this->EditSinartisi->Text;
	AnsiString s="";
	for(int i=1;i<=start.Length();i++)
	{
		if(start[i]=='p'||start[i]=='P'||
		  start[i]=='π'||start[i]=='Π')
		{
			//elegxos gia  exp
			if(start[i]=='p'&&start[i-1]=='x')
				s+='p';
			else
				s+="pi";
		}
		else
			s+=start[i];
	}
	this->EditSinartisi->Text=s;

	this->ButtonGrafiki->Enabled=false;
	this->Chart1->Visible=false;
	this->LabelWait->Visible=true;
	//h grafiki mia fora mono stin arxi
	this->Series1->Clear();
	this->Series7->Clear();
	double min=0.0,max=0.0;
	double a=0.0,b=0.0;
	a=this->EditGrafikiiA->Text.ToDouble();
	b=this->EditGrafikiiB->Text.ToDouble();
	double bima=(b-a)/100;  //100 simeia
	for(double i=a;i<=b;i+=bima)
	{
		this->LabelSinartisi->Caption=this->EditSinartisi->Text;
		double y=this->f(i,this->LabelSinartisi->Caption);
		this->Series1->AddXY(i,y);
		this->Series7->AddXY(i,y);
		Application->ProcessMessages();
	}

	double minC=a-2*bima;
	double maxC=b+2*bima;
	max=this->Series1->MaxYValue();
	min=this->Series1->MinYValue();
	double dif=(abs(max)+abs(min))/5;
	min=min-dif;
	max=max+dif;
	this->Chart1->BottomAxis->Maximum=maxC;
	this->Chart1->BottomAxis->Minimum=minC;
```

```
        this->Chart2->BottomAxis->Maximum=maxC;
        this->Chart2->BottomAxis->Minimum=minC;
        if(max>0)
        {
            this->Chart1->LeftAxis->Maximum=max;
            this->Chart1->LeftAxis->Minimum=min;
            this->Chart2->LeftAxis->Maximum=max;
            this->Chart2->LeftAxis->Minimum=min;
        }
        else
        {
            this->Chart1->LeftAxis->Minimum=min;
            this->Chart1->LeftAxis->Maximum=max;
            this->Chart2->LeftAxis->Minimum=min;
            this->Chart2->LeftAxis->Maximum=max;
        }
            Form1->Series5->Clear();
            Form1->Series5->AddXY(max,0);
            Form1->Series5->AddXY(min,0);
        }
        //make btns enabled
        this->btnDown->Visible=true;
        this->btnUp->Visible=true;
        this->btnLeft->Visible=true;
        this->btnRight->Visible=true;
        this->btnZoomIn->Visible=true;
        this->btnZoomOut->Visible=true;
        this->btnResetZoom->Visible=true;
        this->Chart1->Visible=true;
        this->ButtonGrafiki->Enabled=true;

    if(this->LabelMethodos->Caption=="newton")
    {
        this->GbDixotomisi->Visible=false;
        this->GbTetrogonou->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbNewtonRaphson->Visible=true;
        this->GbRegulaFalsi->Visible=false;
        this->GbSimspon->Visible=false;
        this->GbTemnousa->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbTrapexiou->Visible=false;
        this->ButtonBima->Visible=true;
        this->ButtonEnd->Visible=true;
        this->ButtonOloklirosi->Visible=false;
    }
    else if(this->LabelMethodos->Caption=="dixotomisi")
    {
        this->GbDixotomisi->Visible=true;
        this->GbTetrogonou->Visible=false;
        this->GbLaGrange->Visible=false;
```

```
        this->GbNewtonRaphson->Visible=false;
        this->GbRegulaFalsi->Visible=false;
        this->GbSimspon->Visible=false;
        this->GbTemnousa->Visible=false;
        this->GbTrapexiou->Visible=false;
        this->GbLaGrange->Visible=false;
        this->ButtonBima->Visible=true;
        this->ButtonEnd->Visible=true;
        this->ButtonOloklirosi->Visible=false;
    }
    else if(this->LabelMethodos->Caption=="regulaFalsi")
    {
        this->GbDixotomisi->Visible=false;
        this->GbTetrogonou->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbNewtonRaphson->Visible=false;
        this->GbRegulaFalsi->Visible=true;
        this->GbSimspon->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbTemnousa->Visible=false;
        this->GbTrapexiou->Visible=false;
        this->ButtonBima->Visible=true;
        this->ButtonEnd->Visible=true;
    }
    else if(this->LabelMethodos->Caption=="temnousa")
    {
        this->GbDixotomisi->Visible=false;
        this->GbTetrogonou->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbNewtonRaphson->Visible=false;
        this->GbRegulaFalsi->Visible=false;
        this->GbSimspon->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbTemnousa->Visible=true;
        this->GbTrapexiou->Visible=false;
        this->ButtonBima->Visible=true;
        this->ButtonEnd->Visible=true;
        this->ButtonOloklirosi->Visible=false;
    }
    else if(this->LabelMethodos->Caption=="trapeziou")
    {
        this->GbDixotomisi->Visible=false;
        this->GbTetrogonou->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbNewtonRaphson->Visible=false;
        this->GbRegulaFalsi->Visible=false;
        this->GbSimspon->Visible=false;
        this->GbLaGrange->Visible=false;
        this->GbTemnousa->Visible=false;
        this->GbTrapexiou->Visible=true;
        this->ButtonBima->Visible=false;
```

```
            this->ButtonEnd->Visible=false;
            this->ButtonOloklirosi->Visible=true;
        }
        else if(this->LabelMethodos->Caption=="simpson")
        {
            this->GbDixotomisi->Visible=false;
            this->GbTetrogonou->Visible=false;
            this->GbLaGrange->Visible=false;
            this->GbNewtonRaphson->Visible=false;
            this->GbRegulaFalsi->Visible=false;
            this->GbSimspon->Visible=true;
            this->GbLaGrange->Visible=false;
            this->GbTemnousa->Visible=false;
            this->GbTrapexiou->Visible=false;
            this->ButtonBima->Visible=false;
            this->ButtonEnd->Visible=false;
            this->ButtonOloklirosi->Visible=true;
        }
}
//---------------------------------------------------------------------------
void __fastcall TForm1::N2Click(TObject *Sender)  //close menu
{
        Application->Terminate();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::NewtonRaphson1Click(TObject *Sender)
{
        this->btnClear->Click();
        this->LabelMethodos->Caption="newton";
        this->Caption="Αριθμητικές Μέθοδοι - Μέθοδος Newton - Raphson";
        this->EditSinartisi->Visible=true;
        this->EditGrafikiiA->Visible=true;
        this->EditGrafikiiB->Visible=true;
        this->ButtonGrafiki->Visible=true;
        this->Label33->Visible=true;
        this->Label45->Visible=true;
        this->Label46->Visible=true;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::N7Click(TObject *Sender)
{
        this->btnClear->Click();
        this->LabelMethodos->Caption="dixotomisi";
        this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Διχοτόμησης";
        this->EditSinartisi->Visible=true;
        this->EditGrafikiiA->Visible=true;
        this->EditGrafikiiB->Visible=true;
        this->ButtonGrafiki->Visible=true;
        this->Label33->Visible=true;
        this->Label45->Visible=true;
        this->Label46->Visible=true;
```

```
}
//---------------------------------------------------------------------------
void __fastcall TForm1::RegularFalsi1Click(TObject *Sender)
{
	this->btnClear->Click();
	this->LabelMethodos->Caption="regulaFalsi";
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Regula - Falsi";
	this->EditSinartisi->Visible=true;
	this->EditGrafikiiA->Visible=true;
	this->EditGrafikiiB->Visible=true;
	this->ButtonGrafiki->Visible=true;
	this->Label33->Visible=true;
	this->Label45->Visible=true;
	this->Label46->Visible=true;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::N8Click(TObject *Sender)
{
	this->btnClear->Click();
	this->LabelMethodos->Caption="temnousa";
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Τέμνουσας";
	this->EditSinartisi->Visible=true;
	this->EditGrafikiiA->Visible=true;
	this->EditGrafikiiB->Visible=true;
	this->ButtonGrafiki->Visible=true;
	this->Label33->Visible=true;
	this->Label45->Visible=true;
	this->Label46->Visible=true;

}
//---------------------------------------------------------------------------
void __fastcall TForm1::N9Click(TObject *Sender)
{
	this->btnClear->Click();
	this->LabelMethodos->Caption="trapeziou";
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Τραπεζίου";
	this->EditSinartisi->Visible=true;
	this->EditGrafikiiA->Visible=true;
	this->EditGrafikiiB->Visible=true;
	this->ButtonGrafiki->Visible=true;
	this->Label33->Visible=true;
	this->Label45->Visible=true;
	this->Label46->Visible=true;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Simpson1Click(TObject *Sender)
{
	this->btnClear->Click();
	this->LabelMethodos->Caption="simpson";
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Simpson";
	this->EditSinartisi->Visible=true;
```

```
	this->EditGrafikiiA->Visible=true;
	this->EditGrafikiiB->Visible=true;
	this->ButtonGrafiki->Visible=true;
	this->Label33->Visible=true;
	this->Label45->Visible=true;
	this->Label46->Visible=true;
}
//---------------------------------------------------------------------------
//zoon 115 apo 100% (+)
void __fastcall TForm1::btnZoomInClick(TObject *Sender)
{
	this->Chart1->ZoomPercent(115);
}
//---------------------------------------------------------------------------
//zoon 85 apo 100% (-)
void __fastcall TForm1::btnZoomOutClick(TObject *Sender)
{
	this->Chart1->ZoomPercent(85);
}
//---------------------------------------------------------------------------
//grafiki pros ta aristera
void __fastcall TForm1::btnLeftClick(TObject *Sender)
{
	double max=this->Chart1->BottomAxis->Maximum;
	double min=this->Chart1->BottomAxis->Minimum;
	double diff=max-min;
	double move=diff/20;
	this->Chart1->BottomAxis->Maximum=max+move;
	this->Chart1->BottomAxis->Minimum=min+move;
}
//---------------------------------------------------------------------------
//grafiki pros ta deksia
void __fastcall TForm1::btnRightClick(TObject *Sender)
{
	double max=this->Chart1->BottomAxis->Maximum;
	double min=this->Chart1->BottomAxis->Minimum;
	double diff=max-min;
	double move=diff/20;
	this->Chart1->BottomAxis->Maximum=max-move;
	this->Chart1->BottomAxis->Minimum=min-move;
}
//---------------------------------------------------------------------------
//grafiki pros ta pano
void __fastcall TForm1::btnUpClick(TObject *Sender)
{
	double max=this->Chart1->LeftAxis->Maximum;
	double min=this->Chart1->LeftAxis->Minimum;
	double diff=max-min;
	double move=diff/20;
	this->Chart1->LeftAxis->Maximum=max-move;
	this->Chart1->LeftAxis->Minimum=min-move;
```

```
}
//---------------------------------------------------------------------------
//grafiki pros ta kato
void __fastcall TForm1::btnDownClick(TObject *Sender)
{
	double max=this->Chart1->LeftAxis->Maximum;
	double min=this->Chart1->LeftAxis->Minimum;
	double diff=max-min;
	double move=diff/20;
	this->Chart1->LeftAxis->Maximum=max+move;
	this->Chart1->LeftAxis->Minimum=min+move;
}
//---------------------------------------------------------------------------
//on enter make graf
void __fastcall TForm1::EditSinartisiKeyDown(TObject *Sender, WORD &Key,
    TShiftState Shift)
{
	if(Key==VK_RETURN&&this->ButtonGrafiki->Enabled)
		this->ButtonGrafiki->Click();
}
//---------------------------------------------------------------------------
//reset zoom
void __fastcall TForm1::btnResetZoomClick(TObject *Sender)
{
	this->Chart1->UndoZoom();
}
//---------------------------------------------------------------------------
//show Y values
void __fastcall TForm1::btnClearClick(TObject *Sender)
{
	for(int i=0;i<this->SgValuesGramEks->RowCount;i++)
		for(int j=0;j<this->SgValuesGramEks->ColCount;j++)
			this->SgValuesGramEks->Cells[j][i]="";
	for(int i=0;i<this->SgValuesOloklirosi->RowCount;i++)
		for(int j=0;j<this->SgValuesOloklirosi->ColCount;j++)
			this->SgValuesOloklirosi->Cells[j][i]="";
	for(int i=0;i<this->SgNewton->RowCount;i++)
		for(int j=0;j<this->SgNewton->ColCount;j++)
			this->SgNewton->Cells[j][i]="";
	this->SgValuesGramEks->Visible=false;
	this->SgValuesOloklirosi->Visible=false;
	this->SgNewton->Visible=false;
	this->SgValuesGramEks->RowCount=2;
	this->SgValuesOloklirosi->RowCount=2;
	this->SgNewton->RowCount=2;

	this->Label33->Visible=false;
	this->Label45->Visible=false;
	this->Label46->Visible=false;
	this->EditSinartisi->Visible=false;
	this->EditSinartisi->Clear();
```

```
this->EditGrafikiiA->Visible=false;
this->EditGrafikiiA->Clear();
this->EditGrafikiiB->Visible=false;
this->EditGrafikiiB->Clear();
this->ButtonGrafiki->Visible=false;
this->btnTetragonouSave->Enabled=false;
this->btnLaGrangeSave->Enabled=false;
this->GbDixotomisi->Visible=false;
this->EditDixotomisiA->Clear();
this->EditDixotomisiB->Clear();
this->EditDixotomisiE->Clear();
this->EditDixotomisiT->Clear();
this->LabelNewA->Caption=0;
this->LabelNewB->Caption=0;

this->GbNewtonRaphson->Visible=false;
this->EditNewtonXo->Clear();
this->EditNewtonE->Clear();
this->EditNewtonT->Clear();

this->GbRegulaFalsi->Visible=false;
this->EditRegulaFalsiA->Clear();
this->EditRegulaFalsiB->Clear();
this->EditRegulaFalsiE->Clear();
this->EditRegulaFalsiT->Clear();

this->GbSimspon->Visible=false;
this->EditSimpsonA->Clear();
this->EditSimpsonB->Clear();
this->EditSimpsonDiastimata->Clear();

this->GbTemnousa->Visible=false;
this->EditTemnousaA->Clear();
this->EditTemnousaB->Clear();
this->EditTemnousaE->Clear();
this->EditTemnousaT->Clear();

this->GbTrapexiou->Visible=false;
this->EdTrapeziouA->Clear();
this->EdTrapeziouB->Clear();
this->EdTrapeziouDiastima->Clear();

this->GbBimaGramEks->Visible=false;

this->GbTetrogonou->Visible=false;
this->EditTetragonou_X->Clear();
this->EditTetragonou_Y->Clear();

this->GbLaGrange->Visible=false;
this->GbFinalGramEks->Visible=false;
this->GbPoints->Visible=false;
```

```
	this->GbOloklirosi->Visible=false;
	this->GbTetragonouApot->Visible=false;
	this->GbPoints->Visible=false;
	this->btnUp->Visible=false;
	this->btnDown->Visible=false;
	this->btnLeft->Visible=false;
	this->btnRight->Visible=false;
	this->btnZoomIn->Visible=false;
	this->btnZoomOut->Visible=false;
	this->btnResetZoom->Visible=false;
	Form1->Labelx->Visible=false;
	Form1->Labely->Visible=false;
	Form1->Label_X->Visible=false;
	Form1->Label_Y->Visible=false;
	//Form1->series[i]->Clear();
	this->Series1->Clear();
	this->Series2->Clear();
	this->Series3->Clear();
	this->Series4->Clear();
	this->Series5->Clear();
	this->Series6->Clear();
	this->Series7->Clear();
	this->Series8->Clear();
	this->Series9->Clear();
	this->Series10->Clear();
	this->Series11->Clear();
	this->Series12->Clear();
	this->Series13->Clear();
	this->Series14->Clear();
	this->Series15->Clear();

	this->LabelNewA->Visible=false;
	this->LabelNewB->Visible=false;
	this->Label6->Visible=false;
	this->LabelRow->Caption=0;
	this->Chart1->Visible=false;
	this->Chart2->Visible=false;
	this->Chart3->Visible=false;
	this->ButtonBima->Visible=false;
	this->ButtonEnd->Visible=false;
	this->ButtonBima->Enabled=true;
	this->ButtonEnd->Enabled=true;
	this->ButtonOloklirosi->Visible=false;
	this->ButtonOloklirosi->Enabled=true;
	this->LabelWait->Visible=false;
	this->LabelSinartisi->Caption="";
	this->SGL->Visible=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::ButtonOloklirosiClick(TObject *Sender)
{
```

```
	//arxikes times
	if(this->GbTrapexiou->Visible)
	{
		this->GbOloklirosi->Visible=true;
		this->SgValuesOloklirosi->Visible=true;
		this->SgValuesOloklirosi->Cells[0][0]="A/A";
		this->SgValuesOloklirosi->Cells[1][0]="X";
		this->SgValuesOloklirosi->Cells[2][0]="Y_X";
		methodoi.trapeziou();
		this->ButtonOloklirosi->Enabled=false;
	}
	else if(this->GbSimspon->Visible)
	{
		this->GbOloklirosi->Visible=true;
		this->SgValuesOloklirosi->Visible=true;
		this->SgValuesOloklirosi->Cells[0][0]="A/A";
		this->SgValuesOloklirosi->Cells[1][0]="X";
		this->SgValuesOloklirosi->Cells[2][0]="Y_X";
		methodoi.simpson();
		this->ButtonOloklirosi->Enabled=false;
	}
	else if(this->GbLaGrange->Visible)
	{
		methodoi.laGrange();
	}
	else if(this->GbTetrogonou->Visible)
	{
		methodoi.tetragonou();
	}
	this->Label3->Visible=false;
	this->LabelPointM->Visible=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Chart1ClickSeries(TCustomChart *Sender,
	TChartSeries *Series, int ValueIndex, TMouseButton Button,
	TShiftState Shift, int X, int Y)
{
	if(Series==Form1->Series2)
	{
		Form1->Labelx->Visible=true;
		Form1->Labely->Visible=true;
		Form1->Label_X->Visible=true;
		Form1->Label_Y->Visible=true;
		Form1->Label_X->Caption=FloatToStr(Series->XValue[ValueIndex]);
		Form1->Label_Y->Caption=FloatToStr(Series->YValue[ValueIndex]);
	}
}
//---------------------------------------------------------------------------
void __fastcall TForm1::N6Click(TObject *Sender)
{
	this->btnClear->Click();
```

```
	this->GbLaGrange->Visible=true;
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος LaGrange";
	this->GbTetrogonou->Visible=false;
	this->GbDixotomisi->Visible=false;
	this->GbNewtonRaphson->Visible=false;
	this->GbRegulaFalsi->Visible=false;
	this->GbSimspon->Visible=false;
	this->GbTemnousa->Visible=false;
	this->GbTrapexiou->Visible=false;
	this->ButtonBima->Visible=false;
	this->ButtonEnd->Visible=false;
	this->ButtonOloklirosi->Visible=true;
	this->SgValuesOloklirosi->Visible=true;
	//this->SGlagrange->Visible=true;
	this->SgValuesOloklirosi->Cells[0][0]="A/A";
	this->SgValuesOloklirosi->Cells[1][0]="X";
	this->SgValuesOloklirosi->Cells[2][0]="Y";
}
//---------------------------------------------------------------------------
void __fastcall TForm1::btnLaGrangeSaveClick(TObject *Sender)
{
	this->Chart3->Visible=true;
	//this->Series15->AddXY(this->EditLaGrange_X->Text.ToDouble(),this-
>EditLaGrange_Y->Text.ToDouble());
	int points=this->LabelRow->Caption.ToInt();
	points++;
	this->LabelRow->Caption=points;
	//this->Series15->AddXY(this->EditLaGrange_X->Text.ToDouble(),this-
>EditLaGrange_Y->Text.ToDouble());
	this->SgValuesOloklirosi->RowCount++;
	this->Series15->AddXY(this->EditLaGrange_X->Text.ToDouble(),this-
>EditLaGrange_Y->Text.ToDouble());
	this->SgValuesOloklirosi->Cells[0][points]=points;
	this->SgValuesOloklirosi->Cells[1][points]=this->EditLaGrange_X->Text;
	this->SgValuesOloklirosi->Cells[2][points]=this->EditLaGrange_Y->Text;
	this->EditLaGrange_X->Text="";
	this->EditLaGrange_Y->Text="";
	this->EditLaGrange_X->SetFocus();
	//Form1->Series2->AddXY(this->EditLaGrange_X,this->EditLaGrange_Y);
	//this->Chart3->Visible=true;
}
//---------------------------------------------------------------------------

void __fastcall TForm1::N10Click(TObject *Sender)
{
	this->btnClear->Click();
	this->GbLaGrange->Visible=false;
	this->GbTetrogonou->Visible=true;
	this->Caption=" Αριθμητικές Μέθοδοι - Μέθοδος Ευθείας Ελαχίστων Τετραγώνων";
	this->GbDixotomisi->Visible=false;
	this->GbNewtonRaphson->Visible=false;
```

```
	this->GbRegulaFalsi->Visible=false;
	this->GbSimspon->Visible=false;
	this->GbTemnousa->Visible=false;
	this->GbTrapexiou->Visible=false;
	this->ButtonBima->Visible=false;
	this->ButtonEnd->Visible=false;
	this->ButtonOloklirosi->Visible=true;
	this->SgValuesOloklirosi->Visible=true;
	this->SgValuesOloklirosi->Cells[0][0]="A/A";
	this->SgValuesOloklirosi->Cells[1][0]="X";
	this->SgValuesOloklirosi->Cells[2][0]="Y";
}
//---------------------------------------------------------------------------

void __fastcall TForm1::btnTetragonouSaveClick(TObject *Sender)
{
	this->Chart3->Visible=true;
	int points=this->LabelRow->Caption.ToInt();
	points++;
	this->LabelRow->Caption=points;
	this->SgValuesOloklirosi->RowCount++;
	this->Series15->AddXY(this->EditTetragonou_X->Text.ToDouble(),this-
>EditTetragonou_Y->Text.ToDouble());
	this->SgValuesOloklirosi->Cells[0][points]=points;
	this->SgValuesOloklirosi->Cells[1][points]=this->EditTetragonou_X->Text;
	this->SgValuesOloklirosi->Cells[2][points]=this->EditTetragonou_Y->Text;
	this->EditTetragonou_X->Text="";
	this->EditTetragonou_Y->Text="";
	this->EditTetragonou_X->SetFocus();


}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditSinartisiExit(TObject *Sender)
{
	this->kommaTelia(this->EditSinartisi);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditGrafikiiAExit(TObject *Sender)
{
	this->kommaTelia(this->EditGrafikiiA);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditGrafikiiBExit(TObject *Sender)
{
	this->kommaTelia(this->EditGrafikiiB);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditGrafikiiAChange(TObject *Sender)
{
	if(this->EditGrafikiiA->Text!=""&&
```

```
        this->EditGrafikiiB->Text!="")
            this->ButtonGrafiki->Enabled=true;
    else
            this->ButtonGrafiki->Enabled=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTemnousaAExit(TObject *Sender)
{
    this->kommaTelia(this->EditTemnousaA);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditLaGrange_XExit(TObject *Sender)
{
    this->kommaTelia(this->EditLaGrange_X);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditLaGrange_YExit(TObject *Sender)
{
    this->kommaTelia(this->EditLaGrange_Y);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EdTrapeziouAExit(TObject *Sender)
{
    this->kommaTelia(this->EdTrapeziouA);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EdTrapeziouBExit(TObject *Sender)
{
    this->kommaTelia(this->EdTrapeziouB);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EdTrapeziouDiastimaExit(TObject *Sender)
{
    this->kommaTelia(this->EdTrapeziouDiastima);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTetragonou_XExit(TObject *Sender)
{
    this->kommaTelia(this->EditTetragonou_X);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTetragonou_YExit(TObject *Sender)
{
    this->kommaTelia(this->EditTetragonou_Y);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditRegulaFalsiAExit(TObject *Sender)
{
    this->kommaTelia(this->EditRegulaFalsiA);
}
//---------------------------------------------------------------------------
```

```
void __fastcall TForm1::EditRegulaFalsiBExit(TObject *Sender)
{
	this->kommaTelia(this->EditRegulaFalsiB);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditRegulaFalsiEExit(TObject *Sender)
{
	this->kommaTelia(this->EditRegulaFalsiE);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditRegulaFalsiTExit(TObject *Sender)
{
	this->kommaTelia(this->EditRegulaFalsiT);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditDixotomisiAExit(TObject *Sender)
{
	this->kommaTelia(this->EditDixotomisiA);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditDixotomisiBExit(TObject *Sender)
{
	this->kommaTelia(this->EditDixotomisiB);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditDixotomisiEExit(TObject *Sender)
{
	this->kommaTelia(this->EditDixotomisiE);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditDixotomisiTExit(TObject *Sender)
{
	this->kommaTelia(this->EditDixotomisiT);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditNewtonXoExit(TObject *Sender)
{
	this->kommaTelia(this->EditNewtonXo);
}
//---------------------------------------------------------------------------

void __fastcall TForm1::EditNewtonEExit(TObject *Sender)
{
	this->kommaTelia(this->EditNewtonE);
}
```

```
//---------------------------------------------------------------------------

void __fastcall TForm1::EditNewtonTExit(TObject *Sender)
{
	this->kommaTelia(this->EditNewtonT);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditSimpsonAExit(TObject *Sender)
{
	this->kommaTelia(this->EditSimpsonA);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditSimpsonBExit(TObject *Sender)
{
	this->kommaTelia(this->EditSimpsonB);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditSimpsonDiastimataExit(TObject *Sender)
{
	this->kommaTelia(this->EditSimpsonDiastimata);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTemnousaBExit(TObject *Sender)
{
	this->kommaTelia(this->EditTemnousaB);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTemnousaEExit(TObject *Sender)
{
	this->kommaTelia(this->EditTemnousaE);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTemnousaTExit(TObject *Sender)
{
	this->kommaTelia(this->EditTemnousaT);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::FormCreate(TObject *Sender)
{
	m_pEngine = engOpen(NULL);
	if(m_pEngine == NULL)
	{
		exit(1);
	}
	engSetVisible(m_pEngine,false);
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTetragonou_XKeyDown(TObject *Sender, WORD &Key,
	TShiftState Shift)
{
	if(Key==VK_RETURN&&this->btnTetragonouSave->Enabled)
```

```
            this->btnTetragonouSave->Click();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditLaGrange_XKeyDown(TObject *Sender, WORD &Key,
    TShiftState Shift)
{
      if(Key==VK_RETURN&&this->btnLaGrangeSave->Enabled)
            this->btnLaGrangeSave->Click();
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditLaGrange_XChange(TObject *Sender)
{
      if(this->EditLaGrange_X->Text!=""&&
        this->EditLaGrange_Y->Text!="")
            this->btnLaGrangeSave->Enabled=true;
      else
            this->btnLaGrangeSave->Enabled=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::EditTetragonou_XChange(TObject *Sender)
{
      if(this->EditTetragonou_X->Text!=""&&
        this->EditTetragonou_Y->Text!="")
            this->btnTetragonouSave->Enabled=true;
      else
            this->btnTetragonouSave->Enabled=false;
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Chart3ClickSeries(TCustomChart *Sender,
    TChartSeries *Series, int ValueIndex, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
      if(Series==Form1->Series15)
      {
            Form1->Labelx->Visible=true;
            Form1->Labely->Visible=true;
            Form1->Label_X->Visible=true;
            Form1->Label_Y->Visible=true;
            Form1->Label_X->Caption=FloatToStr(Series->XValue[ValueIndex]);
            Form1->Label_Y->Caption=FloatToStr(Series->YValue[ValueIndex]);
      }
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Chart2ClickSeries(TCustomChart *Sender,
    TChartSeries *Series, int ValueIndex, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
      if(Series==Form1->Series8)
      {
            Form1->Labelx->Visible=true;
            Form1->Labely->Visible=true;
```

```
            Form1->Label_X->Visible=true;
            Form1->Label_Y->Visible=true;
            Form1->Label_X->Caption=FloatToStr(Series->XValue[ValueIndex]);
            Form1->Label_Y->Caption=FloatToStr(Series->YValue[ValueIndex]);
        }
}
//---------------------------------------------------------------------------
void __fastcall TForm1::BtnLaGrangegrafClick(TObject *Sender)
{
            this->Series13->Clear();
            this->Series14->Clear();
            this->Chart3->Visible=true;
            double min=0.0,max=0.0;
            double a=0.0,b=0.0;
            a=this->EditLaGrangeA->Text.ToDouble();
            b=this->EditLaGrangeB->Text.ToDouble();
            //ShowMessage(b);
            double bima=(b-a)/100;  //100 simeia
            for(double i=a;i<=b;i+=bima)
            {
                this->LabelSinartisi->Caption=this->LabelLaGrange->Caption;
                double y=this->f(i,this->LabelLaGrange->Caption);
                this->Series13->AddXY(i,y);
                this->Series14->AddXY(i,y);
                Application->ProcessMessages();
            }
}
//---------------------------------------------------------------------------
```

**Μέθοδοι**

```
#pragma hdrstop
#include "methodoi.h"
#include "Unit1.h"
#pragma package(smart_init)
#include <vcl.h>
//---------------------------------------------------------------------------
Methodoi::Methodoi()
{}
//---------------------------------------------------------------------------
Methodoi::~Methodoi()
{}
//---------------------------------------------------------------------------
//methodos dixotomisis
//epistrefei false an bima-bima
bool Methodoi::dixotomisi(void)
{
	//an OXI bima-bima
	if(Form1->Label6->Caption=="true")
	{
		int a=Form1->LabelRow->Caption.ToInt();
		a++;
		Form1->LabelRow->Caption=a;
		//put a and b to table
		Form1->SgValuesGramEks->Cells[1][a]=Form1->LabelNewA->Caption;
		Form1->SgValuesGramEks->Cells[2][a]=Form1->LabelNewB->Caption;
	}
	//select last row
	TGridRect myRect;
	myRect.Top=Form1->SgValuesGramEks->RowCount-1;
	myRect.Bottom=Form1->SgValuesGramEks->RowCount-1;
	Form1->SgValuesGramEks->Selection=myRect;
	//+ 1 grammi sto table
	Form1->SgValuesGramEks->RowCount++;
	//metablites dixotomisis
	double a=0.0,b=0.0,ya=0.0,yb=0.0,x=0.0,y=0.0;
	//arithmos epanalipsi
	int z=Form1->LabelRow->Caption.ToInt();
	Form1->SgValuesGramEks->Cells[0][z]=z;
	Form1->LabelBimaT->Caption=z;
```

```
//oria a kai b
a=Form1->LabelNewA->Caption.ToDouble();
b=Form1->LabelNewB->Caption.ToDouble();
//topothetisi neon simeion a kai b
Form1->Series2->Clear();
Form1->Series2->AddXY(a,0);
Form1->Series2->AddXY(b,0);
//euresi ton f(a) kai f(b)
ya=Form1->f(a,Form1->LabelSinartisi->Caption);
yb=Form1->f(b,Form1->LabelSinartisi->Caption);
//topothetisi neon simeion f(a) kai f(b)
Form1->Series2->AddXY(a,ya);
Form1->Series2->AddXY(b,yb);
Form1->LabelPointA->Caption="( "+FloatToStr(a)+" , "+FloatToStr(ya)+" )";
Form1->LabelPointB->Caption="( "+FloatToStr(b)+" , "+FloatToStr(yb)+" )";
//methodos gia ta nea simeia an yparxei riza (if=true)
bool end=false;
if((ya*yb)<0)
{
    x=(a+b)/2;
    //topothetisi simeiou x
    Form1->SgValuesGramEks->Cells[3][z]=x;
    Form1->Series2->AddXY(x,0);
    //topothetisi toy f(x)
    y=Form1->f(x,Form1->LabelSinartisi->Caption);
    Form1->SgValuesGramEks->Cells[4][z]=y;
    Form1->LabelBimaRiza->Caption=x;
    Form1->Series2->AddXY(x,y);
    Form1->LabelPointM->Caption="( "+FloatToStr(x)+" , "+FloatToStr(y)+" )";
    //eleghos gia telos
    double e=10.0;
    if(z>1)
    {
        double xprev=Form1->SgValuesGramEks->Cells[3][z-1].ToDouble();
        double xx=0.0;
        if(x<0)  xx=x*(-1);
        else     xx=x;
        if(xprev<0)  xprev=xprev*(-1);
        e=(xx-xprev);  //akribeia
        if(e<0)  e=e*(-1);
        Form1->LabelBimaE->Caption=e;
        Form1->LabelFinalE->Caption=e;
    }
    if(Form1->EditDixotomisiT->Text.ToInt()==Form1->SgValuesGramEks-
>Cells[0][z].ToInt()||e<=Form1->EditDixotomisiE->Text.ToDouble())
    {
        Form1->Label6->Caption="false";
        end=true;
        Form1->LabelFinalT->Caption=z;
        Form1->LabelFinalRiza->Caption=x;
        Form1->GbFinalGramEks->Visible=true;
```

```
                Form1->GbBimaGramEks->Visible=false;
                Form1->ButtonBima->Enabled=false;
                Form1->ButtonEnd->Enabled=false;
            }
            else  //nea simeia
            {
                if((ya*y)<0)
                {
                    a=a;
                    b=x;
                }
                else if((ya*y)>0)
                {
                    a=x;
                    b=b;
                }
                else if((ya*y)==0)
                {
                    Form1->Label6->Caption="false";
                    end=true;
                    Form1->LabelFinalT->Caption=z;
                    Form1->LabelFinalRiza->Caption=x;
                    Form1->GbFinalGramEks->Visible=true;
                    Form1->GbBimaGramEks->Visible=false;
                    Form1->ButtonBima->Enabled=false;
                    Form1->ButtonEnd->Enabled=false;
                }
                Form1->LabelNewA->Caption=a;
                Form1->LabelNewB->Caption=b;
            }
        }
        else
        {
            //ShowMessage("den yparxei riza");
        }
        //dimioyrgia grafikis
        Form1->makeBordersGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),Form1->Series2->MinYValue(),Form1->Series2->MaxYValue(),Form1-
>Chart1);
        Form1->makeOutBorders(Form1->SgValuesGramEks->Cells[1][z].ToDouble(),Form1-
>SgValuesGramEks->Cells[2][z].ToDouble(),Form1->Chart1->LeftAxis->Minimum,Form1-
>Chart1->LeftAxis->Maximum,Form1->Chart1->BottomAxis->Minimum,Form1->Chart1-
>BottomAxis->Maximum);
        Form1->makeGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),100,Form1->Series1);
        //energopoiisi btn mexri telos
        if(Form1->Label6->Caption=="false"&&!end)
        {
            Form1->ButtonEnd->Enabled=true;
            Form1->ButtonBima->Enabled=true;
        }
```

```
        else
        {
                Form1->ButtonBima->Enabled=false;
                Form1->ButtonEnd->Enabled=false;
        }
        if(Form1->Label6->Caption=="false")
                return false;
        else
                return true;
}
//---------------------------------------------------------------------------
//methodos regula falsi
//epistrefei false an bima-bima
bool Methodoi::regulafalsi(void)
{
        //an OXI bima-bima
        if(Form1->Label6->Caption=="true")
        {
                int a=Form1->LabelRow->Caption.ToInt();
                a++;
                Form1->LabelRow->Caption=a;
                //put a and b to table
                Form1->SgValuesGramEks->Cells[1][a]=Form1->LabelNewA->Caption;
                Form1->SgValuesGramEks->Cells[2][a]=Form1->LabelNewB->Caption;
        }
        //select last row
        TGridRect myRect;
        myRect.Top=Form1->SgValuesGramEks->RowCount-1;
        myRect.Bottom=Form1->SgValuesGramEks->RowCount-1;
        Form1->SgValuesGramEks->Selection=myRect;
        //+ 1 grammi sto table
        Form1->SgValuesGramEks->RowCount++;
        //metablites dixotomisis
        double a=0.0,b=0.0,ya=0.0,yb=0.0,x=0.0,y=0.0;
        //arithmos epanalipsi
        int z=Form1->LabelRow->Caption.ToInt();
        Form1->SgValuesGramEks->Cells[0][z]=z;
        Form1->LabelBimaT->Caption=z;
        //oria a kai b
        a=Form1->LabelNewA->Caption.ToDouble();
        b=Form1->LabelNewB->Caption.ToDouble();
        //topothetisi neon simeion a kai b
        Form1->Series2->Clear();
        Form1->Series2->AddXY(a,0);
        Form1->Series2->AddXY(b,0);
        //euresi ton f(a) kai f(b)
        ya=Form1->f(a,Form1->LabelSinartisi->Caption);
        yb=Form1->f(b,Form1->LabelSinartisi->Caption);
        //topothetisi neon simeion f(a) kai f(b)
        Form1->Series2->AddXY(a,ya);
        Form1->Series2->AddXY(b,yb);
```

```
//eutheia simion a, b
Form1->Series6->Clear();
Form1->Series6->AddXY(a,ya);
Form1->Series6->AddXY(b,yb);
Form1->LabelPointA->Caption="( "+FloatToStr(a)+" , "+FloatToStr(ya)+" )";
Form1->LabelPointB->Caption="( "+FloatToStr(b)+" , "+FloatToStr(yb)+" )";
//methodos gia ta nea simeia an yparxei riza (if=true)
bool end=false;
if((ya*yb)<0)
{
    x=b-((b-a)*yb)/(yb-ya);
    //topothetisi simeiou x
    Form1->SgValuesGramEks->Cells[3][z]=x;
    Form1->Series2->AddXY(x,0);
    //topothetisi toy f(x)
    y=Form1->f(x,Form1->LabelSinartisi->Caption);
    Form1->SgValuesGramEks->Cells[4][z]=y;
    Form1->LabelBimaRiza->Caption=x;
    Form1->Series2->AddXY(x,y);
    Form1->LabelPointM->Caption="( "+FloatToStr(x)+" , "+FloatToStr(y)+" )";
    //eleghos gia telos
    double e=10.0;
    if(z>1)
    {
        double xprev=Form1->SgValuesGramEks->Cells[3][z-1].ToDouble();
        double xx=0.0;
        if(x<0)  xx=x*(-1);
        else     xx=x;
        if(xprev<0)  xprev=xprev*(-1);
        e=(xx-xprev);  //akribeia
        if(e<0)  e=e*(-1);
        Form1->LabelBimaE->Caption=e;
        Form1->LabelFinalE->Caption=e;
    }
    if(Form1->EditRegulaFalsiT->Text.ToInt()==Form1->SgValuesGramEks-
>Cells[0][z].ToInt()||e<=Form1->EditRegulaFalsiE->Text.ToDouble())
    {
        Form1->Label6->Caption="false";
        end=true;
        Form1->LabelFinalT->Caption=z;
        Form1->LabelFinalRiza->Caption=x;
        Form1->GbFinalGramEks->Visible=true;
        Form1->GbBimaGramEks->Visible=false;
        Form1->ButtonBima->Enabled=false;
        Form1->ButtonEnd->Enabled=false;
    }
    else  //nea simeia
    {
        if((ya*y)<0)
        {
            a=a;
```

```
                b=x;
            }
            else
            {
                a=x;
                b=b;
            }
            Form1->LabelNewA->Caption=a;
            Form1->LabelNewB->Caption=b;
        }
    }
    else
    {
        //ShowMessage("den yparxei riza");
    }
    //dimioyrgia grafikis
    Form1->makeBordersGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),Form1->Series2->MinYValue(),Form1->Series2->MaxYValue(),Form1-
>Chart1);
    Form1->makeOutBorders(Form1->SgValuesGramEks->Cells[1][z].ToDouble(),Form1-
>SgValuesGramEks->Cells[2][z].ToDouble(),Form1->Chart1->LeftAxis->Minimum,Form1-
>Chart1->LeftAxis->Maximum,Form1->Chart1->BottomAxis->Minimum,Form1->Chart1-
>BottomAxis->Maximum);
    Form1->makeGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),100,Form1->Series1);
    //energopoiisi btn mexri telos
    if(Form1->Label6->Caption=="false"&&!end)
    {
        Form1->ButtonEnd->Enabled=true;
        Form1->ButtonBima->Enabled=true;
    }
    else
    {
        Form1->ButtonBima->Enabled=false;
        Form1->ButtonEnd->Enabled=false;
    }
    if(Form1->Label6->Caption=="false")
        return false;
    else
        return true;
}
//---------------------------------------------------------------------------
//methodos regula falsi
//epistrefei false an bima-bima
bool Methodoi::temnousa(void)
{
    //an OXI bima-bima
    if(Form1->Label6->Caption=="true")
    {
        int a=Form1->LabelRow->Caption.ToInt();
        a++;
```

```
        Form1->LabelRow->Caption=a;
        //put a and b to table
        Form1->SgValuesGramEks->Cells[1][a]=Form1->LabelNewA->Caption;
        Form1->SgValuesGramEks->Cells[2][a]=Form1->LabelNewB->Caption;
}
bool end=false;
//select last row
TGridRect myRect;
myRect.Top=Form1->SgValuesGramEks->RowCount-1;
myRect.Bottom=Form1->SgValuesGramEks->RowCount-1;
Form1->SgValuesGramEks->Selection=myRect;
//+ 1 grammi sto table
Form1->SgValuesGramEks->RowCount++;
//metablites dixotomisis
double a=0.0,b=0.0,ya=0.0,yb=0.0,x=0.0,y=0.0;
//arithmos epanalipsi
int z=Form1->LabelRow->Caption.ToInt();
Form1->SgValuesGramEks->Cells[0][z]=z;
Form1->LabelBimaT->Caption=z;
//oria a kai b
a=Form1->LabelNewA->Caption.ToDouble();
b=Form1->LabelNewB->Caption.ToDouble();
//ston pinaka ta a kai b
Form1->SgValuesGramEks->Cells[1][z]=a;
Form1->SgValuesGramEks->Cells[2][z]=b;
//euresi ton f(a) kai f(b)
ya=Form1->f(a,Form1->LabelSinartisi->Caption);
yb=Form1->f(b,Form1->LabelSinartisi->Caption);
//euresi rizas
x=b-((b-a)/(yb-ya))*yb;
Form1->SgValuesGramEks->Cells[3][z]=x;
//euresi toy f(x)
y=Form1->f(x,Form1->LabelSinartisi->Caption);
Form1->SgValuesGramEks->Cells[4][z]=y;
//topothetisi neon simeion a , b kai x
Form1->Series2->Clear();
Form1->Series2->AddXY(a,0);
Form1->Series2->AddXY(b,0);
Form1->Series2->AddXY(x,0);
Form1->Series2->AddXY(a,ya);
Form1->Series2->AddXY(b,yb);
Form1->Series2->AddXY(x,y);
//eutheia simion a, b
Form1->Series6->Clear();
Form1->Series6->AddXY(a,ya);
Form1->Series6->AddXY(b,yb);
Form1->Series6->AddXY(x,0);
//emfanisi simeion
Form1->LabelPointA->Caption="( "+FloatToStr(a)+" , "+FloatToStr(ya)+" )";
Form1->LabelPointB->Caption="( "+FloatToStr(b)+" , "+FloatToStr(yb)+" )";
Form1->LabelPointM->Caption="( "+FloatToStr(x)+" , "+FloatToStr(y)+" )";
```

```
	Form1->LabelBimaRiza->Caption=x;
	//eleghos gia telos
	double e=10.0;
	if(z>1)
	{
		double xprev=Form1->SgValuesGramEks->Cells[3][z-1].ToDouble();
		double xx=0.0;
		if(x<0)  xx=x*(-1);
		else     xx=x;
		if(xprev<0)  xprev=xprev*(-1);
		e=(xx-xprev);  //akribeia
		if(e<0)  e=e*(-1);
		Form1->LabelBimaE->Caption=e;
		Form1->LabelFinalE->Caption=e;
	}
	if(Form1->EditTemnousaT->Text.ToInt()==Form1->SgValuesGramEks-
>Cells[0][z].ToInt()||e<=Form1->EditTemnousaE->Text.ToDouble())
	{
		Form1->Label6->Caption="false";
		end=true;
		Form1->LabelFinalT->Caption=z;
		Form1->LabelFinalRiza->Caption=x;
		Form1->GbFinalGramEks->Visible=true;
		Form1->GbBimaGramEks->Visible=false;
		Form1->ButtonBima->Enabled=false;
		Form1->ButtonEnd->Enabled=false;
	}
	else  //nea simeia
	{
		if(z>0)
		{
			Form1->SgValuesGramEks->Cells[1][z+1]=Form1->SgValuesGramEks-
>Cells[2][z];
			Form1->SgValuesGramEks->Cells[2][z+1]=Form1->SgValuesGramEks-
>Cells[3][z];
			a=Form1->SgValuesGramEks->Cells[1][z+1].ToDouble();
			b=Form1->SgValuesGramEks->Cells[2][z+1].ToDouble();
			Form1->LabelNewA->Caption=a;
			Form1->LabelNewB->Caption=b;
		}
	}
	//dimioyrgia grafikis
	Form1->makeBordersGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),Form1->Series2->MinYValue(),Form1->Series2->MaxYValue(),Form1-
>Chart1);
	Form1->makeOutBorders(Form1->SgValuesGramEks->Cells[1][z].ToDouble(),Form1-
>SgValuesGramEks->Cells[2][z].ToDouble(),Form1->Chart1->LeftAxis->Minimum,Form1-
>Chart1->LeftAxis->Maximum,Form1->Chart1->BottomAxis->Minimum,Form1->Chart1-
>BottomAxis->Maximum);
	Form1->makeGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),100,Form1->Series1);
```

```
        //energopoiisi btn mexri telos
        if(Form1->Label6->Caption=="false"&&!end)
        {
                Form1->ButtonEnd->Enabled=true;
                Form1->ButtonBima->Enabled=true;
        }
        else
        {
                Form1->ButtonBima->Enabled=false;
                Form1->ButtonEnd->Enabled=false;
        }
        if(Form1->Label6->Caption=="false")
                return false;
        else
                return true;
}

//methodos muller
bool Methodoi::muller(void)
{
  //an OXI bima-bima
        if(Form1->Label6->Caption=="true")
        {
                int a=Form1->LabelRow->Caption.ToInt();
                a++;
                Form1->LabelRow->Caption=a;
                //put a, b and c to table
                Form1->SgMuller->Cells[1][a]=Form1->LabelNewA->Caption;
                Form1->SgMuller->Cells[2][a]=Form1->LabelNewB->Caption;
                Form1->SgMuller->Cells[3][a]=Form1->LabelNewC->Caption;
        }
         bool end=false;
        //select last row
        TGridRect myRect;
        myRect.Top=Form1->SgMuller->RowCount-1;
        myRect.Bottom=Form1->SgMuller->RowCount-1;
        Form1->SgMuller->Selection=myRect;
        //+ 1 grammi sto table
        Form1->SgMuller->RowCount++;

        double
a=0.0,b=0.0,c=0.0,ya=0.0,yb=0.0,yc=0.0,x=0.0,y=0.0,s=0.0,sign=0.0,c1=0.0,c2=0.0,d1=0.0;
        //arithmos epanalipsi
        int z=Form1->LabelRow->Caption.ToInt();
        Form1->SgMuller->Cells[0][z]=z;
        //Form1->SgValuesGramEks->Cells[0][z]=z;
        Form1->LabelBimaT->Caption=z;
        //oria a kai c
        a=Form1->LabelNewA->Caption.ToDouble();
        b=Form1->LabelNewB->Caption.ToDouble();
        c=Form1->LabelNewC->Caption.ToDouble();
```

```
//ston pinaka ta a , b kai c
Form1->SgMuller->Cells[1][z]=a;
Form1->SgMuller->Cells[2][z]=b;
Form1->SgMuller->Cells[3][z]=c;
//euresi ton f(a) , f(b) kai f(c)
ya=Form1->f(a,Form1->LabelSinartisi->Caption);
yb=Form1->f(b,Form1->LabelSinartisi->Caption);
yc=Form1->f(c,Form1->LabelSinartisi->Caption);
Form1->SgMuller->Visible=true;
//euresi rizas
c1=(yb-ya)/(b-a);
c2=(yc-yb)/(c-b);
d1=(c2-c1)/(c-a);
s=c2+d1*(c-b);

if(s>0)
   sign=1.0;
else if(s<0)
   sign=-1.0;
else
   sign=0.0;

x=c-(2*yc)/(s+sign*sqrt(abs(s*s-4*yc*d1)));
Form1->SgMuller->Cells[4][z]=x;


//euresi toy f(x)
y=Form1->f(x,Form1->LabelSinartisi->Caption);
Form1->SgMuller->Cells[5][z]=y;
//y=Form1->f(x,Form1->LabelSinartisi->Caption);
//Form1->SgValuesGramEks->Cells[4][z]=y;

//topothetisi neon simeion a , b , c kai x
Form1->Series2->Clear();
Form1->Series2->AddXY(a,0);
Form1->Series2->AddXY(b,0);
Form1->Series2->AddXY(c,0);
Form1->Series2->AddXY(x,0);
Form1->Series2->AddXY(a,ya);
Form1->Series2->AddXY(b,yb);
Form1->Series2->AddXY(c,yc);
Form1->Series2->AddXY(x,y);
//eutheia simion a, b, c
Form1->Series6->Clear();
Form1->Series6->AddXY(a,ya);
Form1->Series6->AddXY(b,yb);
Form1->Series6->AddXY(c,yc);
Form1->Series6->AddXY(x,0);

  //emfanisi simeion
```

```
//Form1->GbPoints->Visible=false;
Form1->LabelPointMullerA->Caption="( "+FloatToStr(a)+" , "+FloatToStr(ya)+" )";
Form1->LabelPointMullerB->Caption="( "+FloatToStr(b)+" , "+FloatToStr(yb)+" )";
Form1->LabelPointMullerC->Caption="( "+FloatToStr(c)+" , "+FloatToStr(yc)+" )";

Form1->LabelPointMullerM->Caption="( "+FloatToStr(x)+" , "+FloatToStr(y)+" )";
Form1->LabelBimaRiza->Caption=x;


//eleghos gia telos
double e=10.0;
if(z>1)
{
        double xprev=Form1->SgMuller->Cells[4][z-1].ToDouble();
        double xx=0.0;
        if(x<0)  xx=x*(-1);
        else     xx=x;
        if(xprev<0)  xprev=xprev*(-1);
        e=(xx-xprev);  //akribeia
        if(e<0)  e=e*(-1);
        Form1->LabelBimaE->Caption=e;
        Form1->LabelFinalE->Caption=e;
}

 if(Form1->EditMullerT->Text.ToInt()==Form1->SgMuller-
>Cells[0][z].ToInt()||e<=Form1->EditMullerE->Text.ToDouble())
//if(Form1->EditTemnousaT->Text.ToInt()==Form1->SgValuesGramEks-
>Cells[0][z].ToInt()||e<=Form1->EditTemnousaE->Text.ToDouble())
{
        Form1->Label6->Caption="false";
        end=true;
        Form1->LabelFinalT->Caption=z;
        Form1->LabelFinalRiza->Caption=x;
        Form1->GbFinalGramEks->Visible=true;
        Form1->GbBimaGramEks->Visible=false;
        Form1->ButtonBima->Enabled=false;
        Form1->ButtonEnd->Enabled=false;
}
else  //nea simeia
      {
        if(z>0)
        {     Form1->SgMuller->Cells[1][z+1]=Form1->SgMuller->Cells[2][z];
              Form1->SgMuller->Cells[2][z+1]=Form1->SgMuller->Cells[3][z];
              Form1->SgMuller->Cells[3][z+1]=Form1->SgMuller->Cells[4][z];
              // Form1->SgValuesGramEks->Cells[1][z+1]=Form1->SgValuesGramEks-
>Cells[2][z];
              //Form1->SgValuesGramEks->Cells[2][z+1]=Form1->SgValuesGramEks-
>Cells[3][z];
              a=Form1->SgMuller->Cells[1][z+1].ToDouble();
              b=Form1->SgMuller->Cells[2][z+1].ToDouble();
              c=Form1->SgMuller->Cells[3][z+1].ToDouble();
```

```
                //a=Form1->SgValuesGramEks->Cells[1][z+1].ToDouble();
                //b=Form1->SgValuesGramEks->Cells[2][z+1].ToDouble();
                 Form1->LabelNewA->Caption=a;
                 Form1->LabelNewB->Caption=b;
                 Form1->LabelNewC->Caption=c;
            }
        }
        //dimioyrgia grafikis
        Form1->makeBordersGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),Form1->Series2->MinYValue(),Form1->Series2->MaxYValue(),Form1-
>Chart1);
        Form1->makeOutBorders(Form1->SgMuller->Cells[1][z].ToDouble(),Form1->SgMuller-
>Cells[3][z].ToDouble(),Form1->Chart1->LeftAxis->Minimum,Form1->Chart1->LeftAxis-
>Maximum,Form1->Chart1->BottomAxis->Minimum,Form1->Chart1->BottomAxis-
>Maximum);
        Form1->makeGraf(Form1->Series2->MinXValue(),Form1->Series2-
>MaxXValue(),100,Form1->Series1);
        //energopoiisi btn mexri telos
         if(Form1->Label6->Caption=="false"&&!end)
        {
            Form1->ButtonEnd->Enabled=true;
            Form1->ButtonBima->Enabled=true;
        }
        else
        {
            Form1->ButtonBima->Enabled=false;
            Form1->ButtonEnd->Enabled=false;
        }
        if(Form1->Label6->Caption=="false")
            return false;
        else
            return true;


}
//---------------------------------------------------------------------------
//methodos newton
bool Methodoi::newton(void)
{
        //an OXI bima-bima
        if(Form1->Label6->Caption=="true")
        {
            int a=Form1->LabelRow->Caption.ToInt();
            a++;
            Form1->LabelRow->Caption=a;
            //put a and b to table
            Form1->SgNewton->Cells[1][1]=Form1->EditNewtonXo->Text;
        }
        bool end=false;
        //select last row
        TGridRect myRect;
```

```
myRect.Top=Form1->SgNewton->RowCount-1;
myRect.Bottom=Form1->SgNewton->RowCount-1;
Form1->SgNewton->Selection=myRect;
//+ 1 grammi sto table
Form1->SgNewton->RowCount++;
//metablites newton
double Xo=0.0,Xn=0.0,Yxn=0.0,YYxn=0.0;
//arithmos epanalipsi
int z=Form1->LabelRow->Caption.ToInt();
Form1->SgNewton->Cells[0][z]=z;
Form1->LabelBimaT->Caption=z;
//euresi metavliton
Xn=Form1->SgNewton->Cells[1][z].ToDouble();
Yxn=Form1->f(Xn,Form1->LabelSinartisi->Caption);
//metatropi sinartisis se 'x^2' gia euresi paragogoy
AnsiString par="";
par+="'";
par+=Form1->LabelSinartisi->Caption;
par+="'";
YYxn=Form1->diff(Xn,par);
Xo=Xn-(Yxn/YYxn);
//topothetisi sto tamplo
Form1->SgNewton->Cells[1][z+1]=Xo;
Form1->SgNewton->Cells[2][z]=Yxn;
Form1->LabelBimaRiza->Caption=Xo;
//topothetisi neon simeion Xn-1 , Xo kai Y(Xn-1)
Form1->Series2->Clear();
Form1->Series2->AddXY(Xn,0);
Form1->Series2->AddXY(Xo,0);
Form1->Series2->AddXY(Xn,Yxn);
Form1->Series8->Clear();
Form1->Series8->AddXY(Xn,0);
Form1->Series8->AddXY(Xo,0);
Form1->Series8->AddXY(Xn,Yxn);
//katheti eythia
Form1->Series12->Clear();
Form1->Series12->AddXY(Xn,0);
Form1->Series12->AddXY(Xn,Yxn);
Form1->Series10->Clear();
Form1->Series10->AddXY(Xn,0);
Form1->Series10->AddXY(Xn,Yxn);
//eutheia simion Xo kai Y(Xn-1)
Form1->Series6->Clear();
Form1->Series6->AddXY(Xn,Yxn);
Form1->Series6->AddXY(Xo,0);
Form1->Series9->Clear();
Form1->Series9->AddXY(Xn,Yxn);
Form1->Series9->AddXY(Xo,0);
//eythia sto aksona x
Form1->Series5->Clear();//αρχικά σβήνεται η ευθεία πάνω στον οριζόντιο άξονα
```

```
	Form1->Series5->AddXY(Form1->Chart1->BottomAxis-
>Minimum,0);//+++++++++++++++++++
	Form1->Series5->AddXY(Form1->Chart1->BottomAxis-
>Maximum,0);//+++++++++++++++++++
	Form1->Series11->Clear();//αρχικά σβήνεται η ευθεία πάνω στον οριζόντιο άξονα
	Form1->Series11->AddXY(Form1->Chart1->BottomAxis-
>Minimum,0);//+++++++++++++++++++
	Form1->Series11->AddXY(Form1->Chart1->BottomAxis-
>Maximum,0);//+++++++++++++++++++
	//dimioyrgia neas mikris grafikis
	Form1->makeBordersGraf(Form1->Series8->MinXValue(),Form1->Series8-
>MaxXValue(),Form1->Series8->MinYValue(),Form1->Series8->MaxYValue(),Form1-
>Chart2);
	Form1->makeGraf(Form1->Series8->MinXValue(),Form1->Series8-
>MaxXValue(),150,Form1->Series7);

	double minX,minY,maxX,maxY;
	if(Form1->Series2->MinXValue()<Form1->Series1->MinXValue())
		minX=Form1->Series2->MinXValue();
	else
		minX=Form1->Series1->MinXValue();
	if(Form1->Series2->MinYValue()<Form1->Series1->MinYValue())
		minY=Form1->Series2->MinYValue();
	else
		minY=Form1->Series1->MinYValue();
	if(Form1->Series2->MaxXValue()>Form1->Series1->MaxXValue())
		maxX=Form1->Series2->MaxXValue();
	else
		maxX=Form1->Series1->MaxXValue();
	if(Form1->Series2->MaxYValue()>Form1->Series1->MaxYValue())
		maxY=Form1->Series2->MaxYValue();
	else
		maxY=Form1->Series1->MaxYValue();


	Form1->makeBordersGraf(minX,maxX,minY,maxY,Form1->Chart1);
	Form1->makeGraf(minX,maxX,100,Form1->Series1);

	//eleghos gia telos
	double e=10.0;
	if(z>1)
	{
		double xprev=Form1->SgNewton->Cells[1][z].ToDouble();
		if(xprev==Xo)
		{
			e=0;
		}
		else
		{
			double xx=0.0;
			if(Xo<0)  xx=Xo*(-1);
```

```
                else    xx=Xo;
                if(xprev<0)  xprev=xprev*(-1);
                e=(xx-xprev);  //akribeia
                if(e<0)  e=e*(-1);
            }
            Form1->LabelBimaE->Caption=e;
            Form1->LabelFinalE->Caption=e;
        }
        if(Form1->EditNewtonT->Text.ToInt()==Form1->SgNewton-
>Cells[0][z].ToInt()||e<=Form1->EditNewtonE->Text.ToDouble())
        {
            Form1->Label6->Caption="false";
            end=true;
            Form1->LabelFinalT->Caption=z;
            Form1->LabelFinalRiza->Caption=Xo;
            Form1->GbFinalGramEks->Visible=true;
            Form1->GbBimaGramEks->Visible=false;
            Form1->ButtonBima->Enabled=false;
            Form1->ButtonEnd->Enabled=false;
        }
        //energopoiisi btn mexri telos
        if(Form1->Label6->Caption=="false"&&!end)
        {
            Form1->ButtonEnd->Enabled=true;
            Form1->ButtonBima->Enabled=true;
        }
        else
        {
            Form1->ButtonBima->Enabled=false;
            Form1->ButtonEnd->Enabled=false;
        }
        if(Form1->Label6->Caption=="false")
            return false;
        else
            return true;
}
//---------------------------------------------------------------------------
//methodos trapeziou
void Methodoi::trapeziou(void)
{
        double embadon=0.0,max_Y=0.0,min_Y=0.0;
        int m=0;
        //arxikes times
        double a=Form1->EdTrapeziouA->Text.ToDouble();
        double b=Form1->EdTrapeziouB->Text.ToDouble();
        int d=Form1->EdTrapeziouDiastima->Text.ToInt();
        //grafiki
        double ya=Form1->f(a,Form1->LabelSinartisi->Caption);
        double yb=Form1->f(b,Form1->LabelSinartisi->Caption);
        Form1->Series2->Clear();
        Form1->Series2->AddXY(a,ya);
```

```
Form1->Series2->AddXY(b,yb);
Form1->Series3->Clear();
Form1->Series3->AddXY(a,0);
Form1->Series3->AddXY(a,ya);
Form1->Series4->Clear();
Form1->Series4->AddXY(b,0);
Form1->Series4->AddXY(b,yb);
Form1->Series5->AddXY(Form1->Chart1->BottomAxis->Minimum,0);
Form1->Series5->AddXY(Form1->Chart1->BottomAxis->Maximum,0);
//apostasi
double h=(b-a)/d;
embadon=(ya+yb)/2;
double athrisma=0.0;
//ypodiastimata
for(int i=1;i<d;i++)
{
	//table
	double x_k=a+(i*h);
	double Y_x_k=Form1->f(x_k,Form1->LabelSinartisi->Caption);
	athrisma+=Y_x_k;
	//table
	Form1->SgValuesOloklirosi->Cells[0][i]=i;  //a/a
	Form1->SgValuesOloklirosi->Cells[1][i]=x_k;  //a
	Form1->SgValuesOloklirosi->Cells[2][i]=Y_x_k;  //ya
	//embadon
	//grammi +1
	Form1->SgValuesOloklirosi->RowCount++;
	//graf
	//Form1->series[i]->Clear();
	Form1->series[i]=new TLineSeries(Form1);
	Form1->series[i]->ParentChart=Form1->Chart1;
	Form1->series[i]->SeriesColor=clBlue;
	Form1->series[i]->AddXY(x_k,0);
	Form1->series[i]->AddXY(x_k,Y_x_k);
	Form1->Series2->AddXY(x_k,Y_x_k);

	//up line
	if(i>1)
	{
		Form1->series[d+i]=new TLineSeries(Form1);
		Form1->series[d+i]->ParentChart=Form1->Chart1;
		Form1->series[d+i]->SeriesColor=clBlue;
		Form1->series[d+i]->AddXY(Form1->SgValuesOloklirosi-
>Cells[1][i].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][i].ToDouble());
		Form1->series[d+i]->AddXY(Form1->SgValuesOloklirosi->Cells[1][i-
1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][i-1].ToDouble());
	}
	m=i;
}
Form1->series[d+m+1]=new TLineSeries(Form1);
Form1->series[d+m+1]->ParentChart=Form1->Chart1;
```

```
	Form1->series[d+m+1]->SeriesColor=clBlue;
	Form1->series[d+m+1]->AddXY(a,ya);
	Form1->series[d+m+1]->AddXY(Form1->SgValuesOloklirosi-
>Cells[1][1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][1].ToDouble());
	Form1->series[d+m+2]=new TLineSeries(Form1);
	Form1->series[d+m+2]->ParentChart=Form1->Chart1;
	Form1->series[d+m+2]->SeriesColor=clBlue;
	Form1->series[d+m+2]->AddXY(b,yb);
	Form1->series[d+m+2]->AddXY(Form1->SgValuesOloklirosi->Cells[1][d-
1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][d-1].ToDouble());
	//Form1->Series2->AddXY(a,ya);
	//Form1->Series2->AddXY(b,yb);
	//Form1->series[i]->Clear();
	embadon+=athrisma;
	embadon*=h;
	//find min and max
	for(int i=1;i<d;i++)
	{
		if(Form1->SgValuesOloklirosi->Cells[2][i].ToDouble()>max_Y)
		{
			max_Y=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
		}
		else if(Form1->SgValuesOloklirosi->Cells[2][i].ToDouble()<min_Y)
		{
			min_Y=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
		}
	}

	if(max_Y>0&&min_Y>0)
		min_Y=0;
	else if(max_Y<0&&min_Y<0)
		max_Y=0;
	//grafiki me 100 simeia
	Form1->makeBordersGraf(a,b,min_Y,max_Y,Form1->Chart1);
	Form1->makeGraf(a-h,b+h,100,Form1->Series1);

	//embadon
	if(embadon>=0)
	Form1->LabelResulEmbadon->Caption=embadon;
	else if(embadon<0)
	{
	 embadon=embadon*(-1);
	 Form1->LabelResulEmbadon->Caption=embadon;
	 }
	//Form1->series[d]->Clear();
}
//---------------------------------------------------------------------------
//methodos simpson
void Methodoi::simpson(void)
{
	int m=0,m1=0;
```

```
	double embadon=0.0,max_Y=0.0,min_Y=0.0;
	//arxikes times
	double a=Form1->EditSimpsonA->Text.ToDouble();
	double b=Form1->EditSimpsonB->Text.ToDouble();
	int d=Form1->EditSimpsonDiastimata->Text.ToInt();
	//grafiki
	double ya=Form1->f(a,Form1->LabelSinartisi->Caption);
	double yb=Form1->f(b,Form1->LabelSinartisi->Caption);
	Form1->Series2->Clear();
	Form1->Series2->AddXY(a,ya);
	Form1->Series2->AddXY(b,yb);
	Form1->Series3->Clear();
	Form1->Series3->AddXY(a,0);
	Form1->Series3->AddXY(a,ya);
	Form1->Series4->Clear();
	Form1->Series4->AddXY(b,0);
	Form1->Series4->AddXY(b,yb);
	Form1->Series5->AddXY(Form1->Chart1->BottomAxis->Minimum,0);
	Form1->Series5->AddXY(Form1->Chart1->BottomAxis->Maximum,0);
	//apostasi
	double h=(b-a)/d;
	//typos simpson
	double sinolo_A=0.0;
	double sinolo_B=0.0;
	double sinolo_C=0.0;
	//m
	m=d/2;
	//X_k
	for(int i=1;i<d;i++)
	{
		//grammi +1
		Form1->SgValuesOloklirosi->RowCount++;
		Form1->SgValuesOloklirosi->Cells[0][i]=i;
		Form1->SgValuesOloklirosi->Cells[1][i]=a+i*h;
		double y_k=Form1->SgValuesOloklirosi->Cells[1][i].ToDouble();
		Form1->SgValuesOloklirosi->Cells[2][i]=Form1->f(y_k,Form1->LabelSinartisi-
>Caption);
		//graf
		Form1->series[i]=new TLineSeries(Form1);
		Form1->series[i]->ParentChart=Form1->Chart1;
		Form1->series[i]->SeriesColor=clBlue;
		Form1->series[i]->AddXY(Form1->SgValuesOloklirosi->Cells[1][i].ToDouble(),0);
		Form1->series[i]->AddXY(Form1->SgValuesOloklirosi-
>Cells[1][i].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][i].ToDouble());
		Form1->Series2->AddXY(a,ya);
		Form1->Series2->AddXY(b,yb);
		//up line
		if(i>1)
		{
			Form1->series[d+i]=new TLineSeries(Form1);
			Form1->series[d+i]->ParentChart=Form1->Chart1;
```

```
                Form1->series[d+i]->SeriesColor=clBlue;
                Form1->series[d+i]->AddXY(Form1->SgValuesOloklirosi-
>Cells[1][i].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][i].ToDouble());
                Form1->series[d+i]->AddXY(Form1->SgValuesOloklirosi->Cells[1][i-
1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][i-1].ToDouble());
            }
            m1=i;

        }
        Form1->series[d+m1+1]=new TLineSeries(Form1);
        Form1->series[d+m1+1]->ParentChart=Form1->Chart1;
        Form1->series[d+m1+1]->SeriesColor=clBlue;
        Form1->series[d+m1+1]->AddXY(a,ya);
        Form1->series[d+m1+1]->AddXY(Form1->SgValuesOloklirosi-
>Cells[1][1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][1].ToDouble());

        Form1->series[d+m1+2]=new TLineSeries(Form1);
        Form1->series[d+m1+2]->ParentChart=Form1->Chart1;
        Form1->series[d+m1+2]->SeriesColor=clBlue;
        Form1->series[d+m1+2]->AddXY(b,yb);
        Form1->series[d+m1+2]->AddXY(Form1->SgValuesOloklirosi->Cells[1][d-
1].ToDouble(),Form1->SgValuesOloklirosi->Cells[2][d-1].ToDouble());

        //Form1->series[i]->Clear();
        //embadon
        sinolo_A=(h/3)*(ya+yb);
        //
        sinolo_B=(2*h/3);
        double athrisma1=0.0;
        double athrisma2=0.0;
        if(m>1)
        {
            for(int k=1;k<=m-1;k++)
            {
                athrisma1+=Form1->SgValuesOloklirosi->Cells[2][2*k].ToDouble();
            }
        }
        sinolo_B*=athrisma1;
        //
        sinolo_C=(4*h/3);
        for(int k=1;k<=m;k++)
        {
            athrisma2+=Form1->SgValuesOloklirosi->Cells[2][2*k-1].ToDouble();
        }
        sinolo_C*=athrisma2;
        //embadon
        embadon=sinolo_A+sinolo_B+sinolo_C;
        //Form1->LabelResulEmbadon->Caption=embadon;
        if(embadon>=0)
        Form1->LabelResulEmbadon->Caption=embadon;
        else if(embadon<0)
```

```
        {
         embadon=embadon*(-1);
         Form1->LabelResulEmbadon->Caption=embadon;
         }
        //find min and max
        for(int i=1;i<d;i++)
        {
                if(Form1->SgValuesOloklirosi->Cells[2][i].ToDouble()>max_Y)
                {
                        max_Y=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
                }
                else if(Form1->SgValuesOloklirosi->Cells[2][i].ToDouble()<min_Y)
                {
                        min_Y=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
                }
        }
        if(max_Y>0&&min_Y>0)
                min_Y=0;
        else if(max_Y<0&&min_Y<0)
                max_Y=0;
        //grafiki me 100 simeia
        Form1->makeBordersGraf(a,b,min_Y,max_Y,Form1->Chart1);
        Form1->makeGraf(a-h,b+h,100,Form1->Series1);
        //Form1->series[i]->Clear();

}
//---------------------------------------------------------------------------
void Methodoi::laGrange(void)
{
        AnsiString polionimo="(";  //polionimo lagrange
        AnsiString l[100];  //sintelestes L(x)
        AnsiString arithmitis="",paronomastis="";  //gia kathe L(x)
        //arithmos simeion
        int points=Form1->SgValuesOloklirosi->Cells[0][Form1->SgValuesOloklirosi-
>RowCount-2].ToInt();
        //L(x)
        for(int i=0;i<points;i++)
        {
                arithmitis="(";
                paronomastis="(";
                //arithmitis
                for(int j=0;j<points;j++)
                {
                        if(i!=j)
                        {
                                arithmitis+="(x-"+Form1->SgValuesOloklirosi->Cells[1][j+1]+")*";
                        }
                }
                //paronomastis
                AnsiString x=Form1->SgValuesOloklirosi->Cells[1][i+1];
                Form1->SGL->Cells[0][0]="L";
```

```
            for(int k=0;k<points;k++)
            {
                if(i!=k)
                {
                    paronomastis+="("+x+"-"+Form1->SgValuesOloklirosi-
>Cells[1][k+1]+")*";
                }
            }
            //sbisimo teleytaioy *
            arithmitis.SetLength(arithmitis.Length()-1);
            arithmitis+=")";
            paronomastis.SetLength(paronomastis.Length()-1);
            paronomastis+=")";
            //dieresi
            l[i]=Form1->dieresi(arithmitis,paronomastis);

            Form1->SGL->Visible=true;
            Form1->SGL->Cells[0][i+1]=i+1;
            Form1->SGL->Cells[1][i+1]=l[i];
        }
        //polionimo
        for(int z=0;z<points;z++)
        {
            polionimo+="("+l[z]+"*"+Form1->SgValuesOloklirosi->Cells[2][z+1]+")+";
        }
        //sbisimo teleytaioy +
        polionimo.SetLength(polionimo.Length()-1);
        polionimo+=")";
        Form1->LabelLaGrange->Caption=Form1->aplopiisi(polionimo);
        Form1->BtnLaGrangegraf->Enabled=true;
}
//---------------------------------------------------------------------------
void Methodoi::tetragonou(void)
{
        double n=0.0,xy=0.0,x=0.0,y=0.0,xx=0.0,a=0.0,b=0.0;
        //arithmos simeion
        n=Form1->SgValuesOloklirosi->Cells[0][Form1->SgValuesOloklirosi->RowCount-
2].ToInt();
        Form1->Chart3->Visible=true;
        for(int i=1;i<=n;i++)
        {
            x+=Form1->SgValuesOloklirosi->Cells[1][i].ToDouble();
            y+=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
            xx+=Form1->SgValuesOloklirosi->Cells[1][i].ToDouble()*Form1-
>SgValuesOloklirosi->Cells[1][i].ToDouble();
            xy+=Form1->SgValuesOloklirosi->Cells[1][i].ToDouble()*Form1-
>SgValuesOloklirosi->Cells[2][i].ToDouble();
        }
        //Form1->Chart3->Visible=true;
        a=((n*xy)-(x*y))/((n*xx)-(x*x));
        b=((xx*y)-(xy*x))/((n*xx)-(x*x));
```

```
	AnsiString f_y="("+FloatToStr(a)+")*x+("+FloatToStr(b)+")";
	Form1->LabelTetragonouPoints->Caption=n;
	Form1->LabelTetragonouX->Caption=x;
	Form1->LabelTetragonouY->Caption=y;
	Form1->LabelTetragonouXX->Caption=xx;
	Form1->LabelTetragonouXY->Caption=xy;
	Form1->LabelTetragonouA->Caption=a;
	Form1->LabelTetragonouB->Caption=b;
	Form1->LabelTetragonouEksisosi->Caption="Y="+f_y;
	Form1->Series13->Clear();
	Form1->Series14->Clear();
	for(int i=1;i<=n;i++)
	{
		x=Form1->SgValuesOloklirosi->Cells[1][i].ToDouble();
		y=Form1->SgValuesOloklirosi->Cells[2][i].ToDouble();
		Form1->Series13->AddXY(x,y);
	}
	double minX=Form1->Series13->MinXValue();
	double maxX=Form1->Series13->MaxXValue();

	double Yminx=Form1->f(minX,f_y);
	double Ymaxx=Form1->f(maxX,f_y);
	Form1->Series14->AddXY(minX,Yminx);
	Form1->Series14->AddXY(maxX,Ymaxx);
	Form1->GbTetragonouApot->Visible=true;
	Form1->Chart3->Visible=true;
}
//---------------------------------------------------------------------------
```

## Βιβλιογραφία

Κυβεντίδη Θωμά, Ολοκληρωτικός λογισμός συναρτήσεων μιας πραγματικής μεταβλητής, Εκδόσεις ΖΗΤΗ, Θεσσαλονίκη, 2005

Γεωργίου, Α.Κ. – Βασιλείου Χ.Γ. – Π., Μη γραμμικές μέθοδοι βελτιστοποίησης, Εκδόσεις ΖΗΤΗ, Θεσσαλονίκη, 1993

Γουσίδου – Κουτίτα Μαρία, Ανώτερα εφαρμοσμένα μαθηματικά και αριθμητικές μέθοδοι, Εκδόσεις Χριστοδουλίδη, Θεσσαλονίκη, 2001

Ακρίβη Γ.Δ – Δουγαλή Β.Α., Εισαγωγή στην Αριθμητική Ανάλυση, Πανεπιστημιακές εκδόσεις Κρήτης, Ηράκλειο, 2004

## Δικτυακοί τόποι

http://mathworld.wolfram.com/SecantMethod.html

http://mathworld.wolfram.com/SimpsonsRule.html

http://www.astro.auth.gr/~kokkotas/lesson/na_book/node103.html#10418

http://www.astro.auth.gr/~kokkotas/lesson/na_book/node142.html

http://www.mathcs.emory.edu/ccs/ccs215/integral/node4.html

http://www.actuar.aegean.gr/notes/sim_arithm_anal.pdf