ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

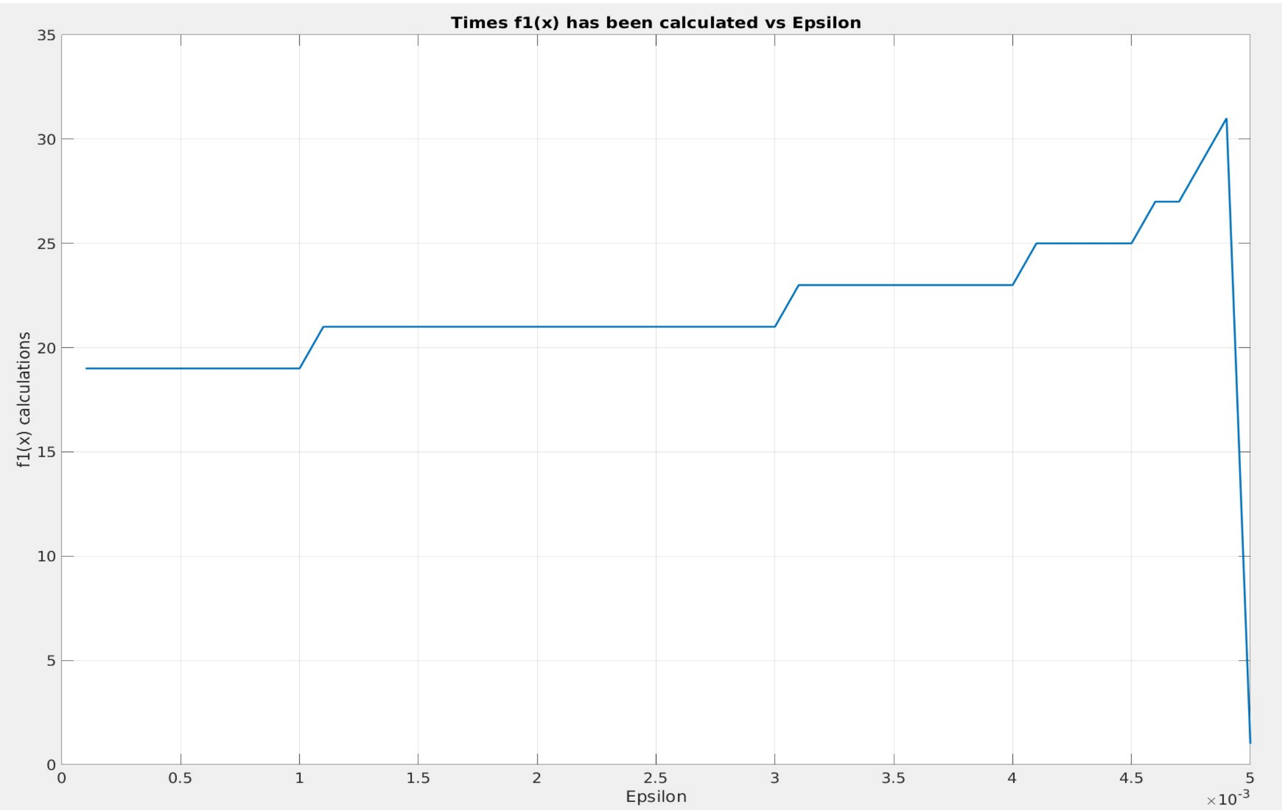
1η Εργαστηριακή Άσκηση

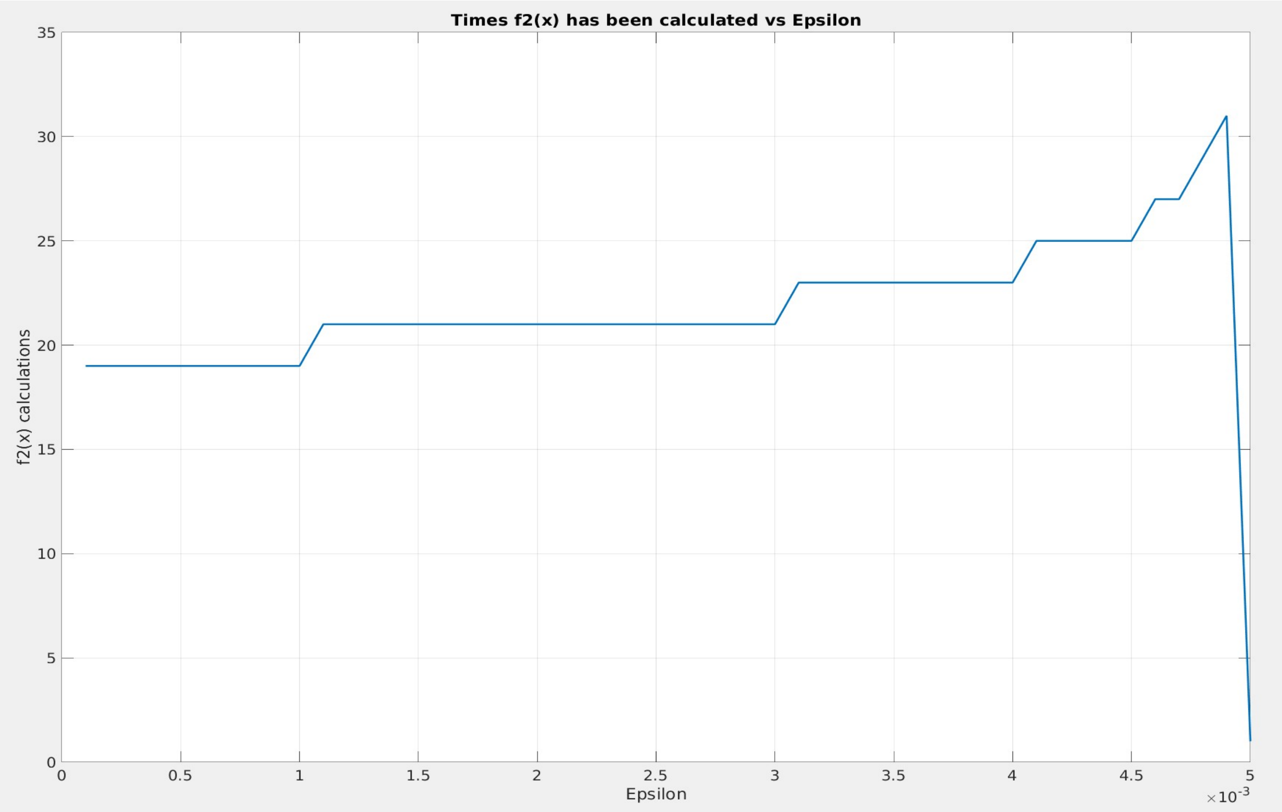
Καράτης Δημήτριος 10775

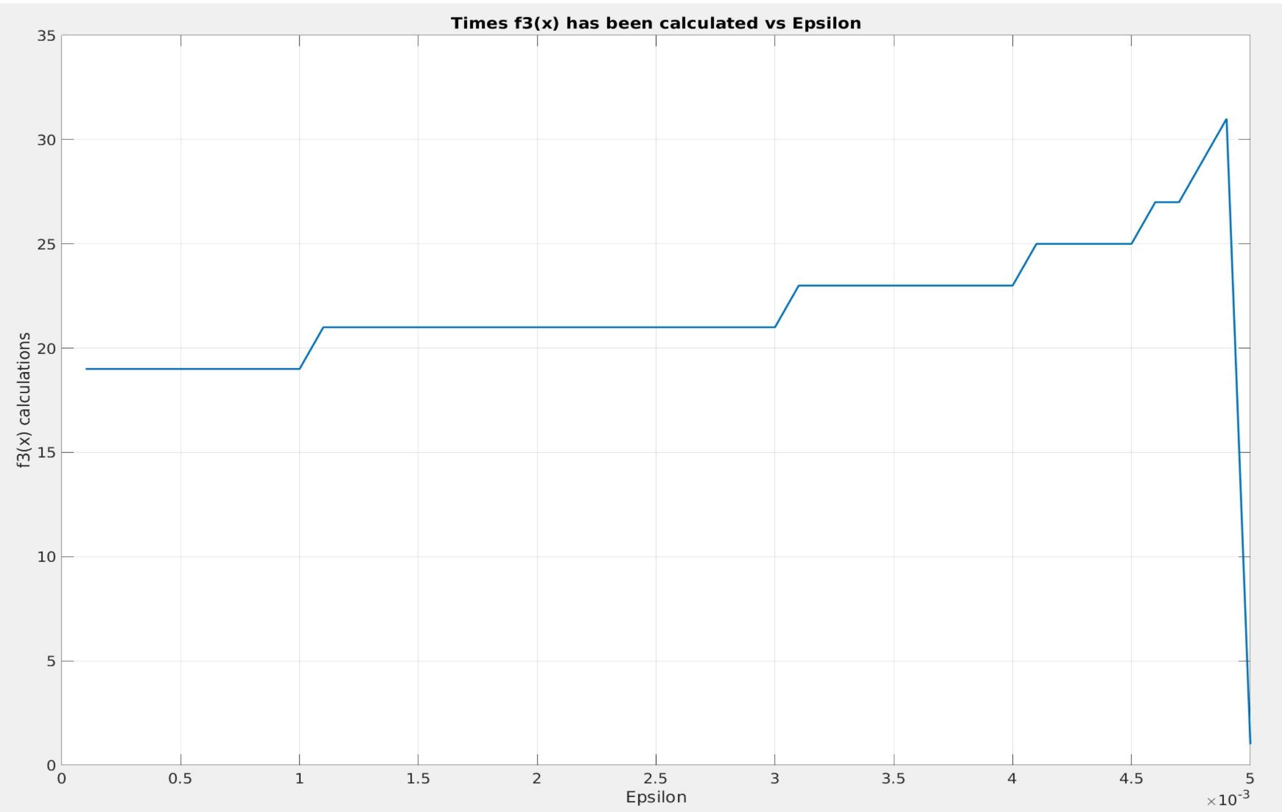
Θέμα 1:

1) Κρατώντας σταθερό το τελικό εύρος τιμών l = 0.01 προκύπτουν οι εξής γραφικές παραστάσεις που υποδηλώνουν τη σχέση μεταξύ του συνολικού αριθμού φορών που χρειάστηκε να υπολογιστεί η εκάστοτε f(x) και των διαφόρων τιμών ε (απόσταση από τη διχοτόμο).

(Σε κάθε κλήση της μεθόδου έχω 2κ + 1 υπολογισμούς της f. Σε κάθε επανάληψη αλλάζουν και το x1 και το x2, άρα έχω 2κ υπολογσιμούς και στο τέλος για την επιστροφή του $f(x_{min})$ έχω επίσης έναν.)



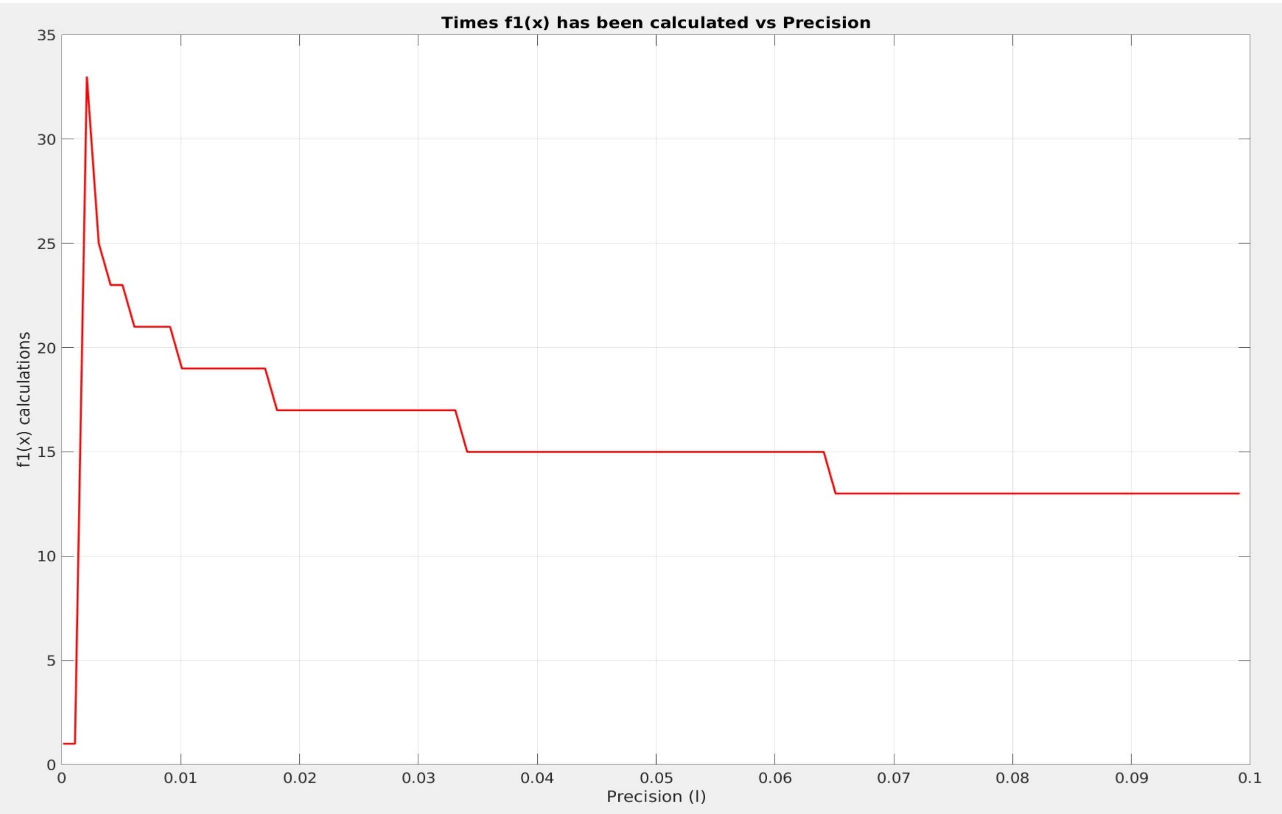

Times f2(x) has been calculated vs Epsilon
f2(x) calculations
Epsilon
35
30
25
20
15
10
5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
×10^-3

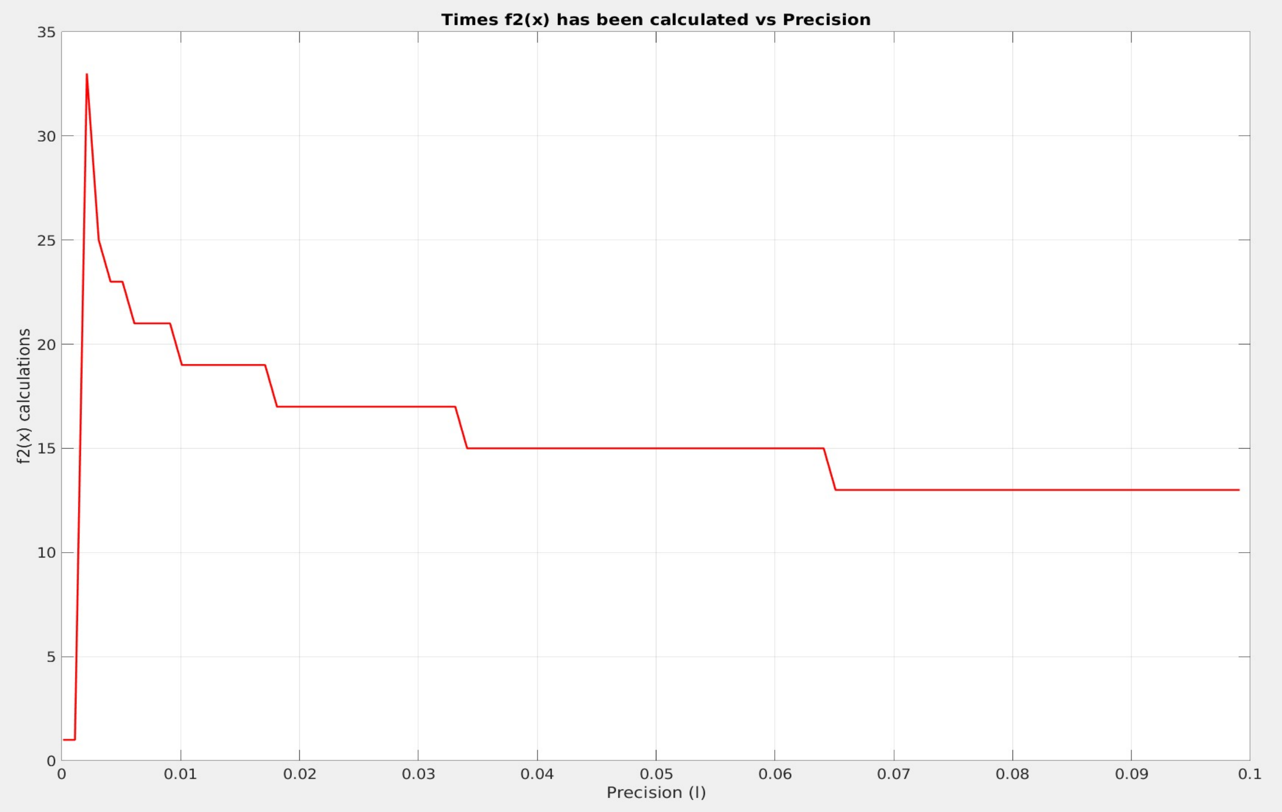

Times f3(x) has been calculated vs Epsilon
f3(x) calculations
Epsilon
35
30
25
20
15
10
5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
×10^-3

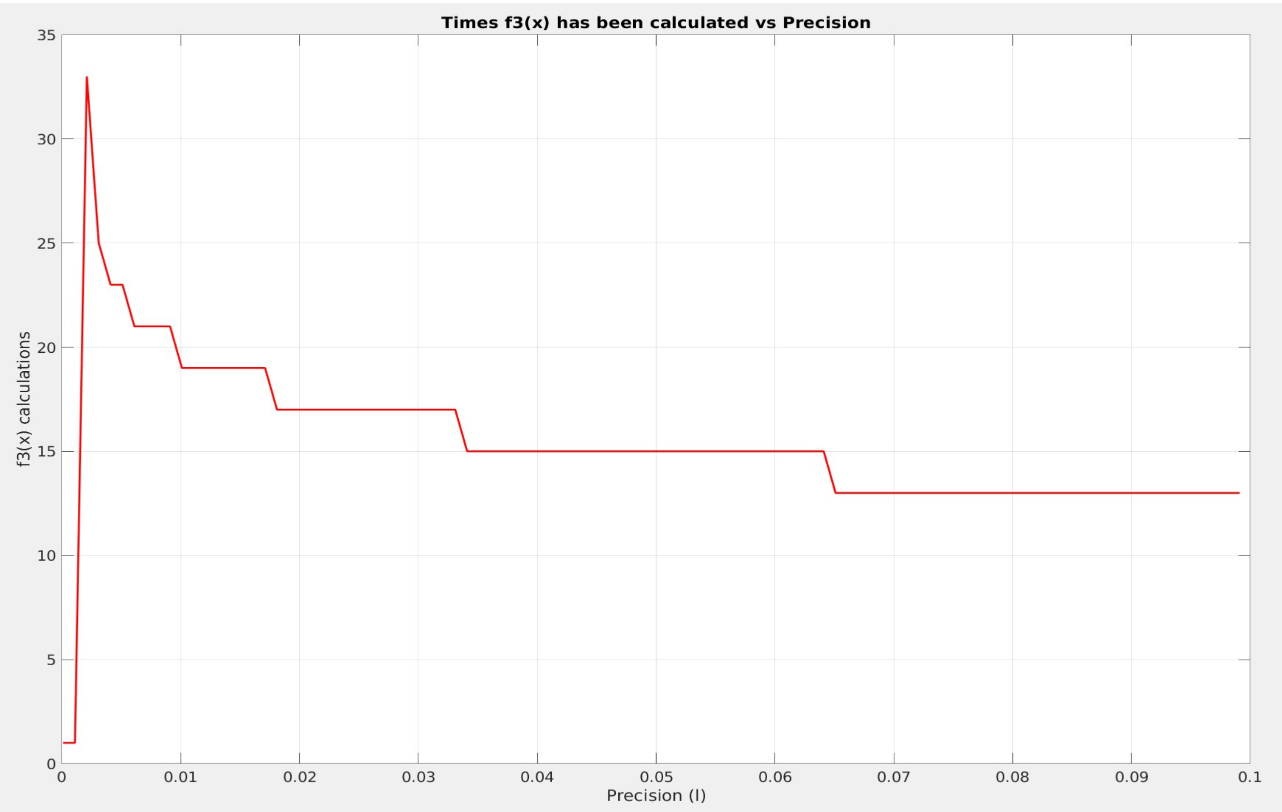
Με βάσει τα διαγράμματα μπορούμε να παρατηρήσουμε ότι και στις 3 περιπτώσεις (δηλαδή και για τις 3 συναρτήσεις), η **μέθοδος της Διχοτόμου** μας δίνει ίδια ή σχεδόν ίδια αποτελέσματα. Επιπλέον, παρατηρούμε ότι για τιμές ε > 0.005 η μέθοδος δεν υπολογίζει καθόλου τα εκάστοτε f(x) διότι ως γνωστόν, η μέθοδος συγκλίνει μονάχα για τιμές των ε και l τω 2 * ε < l => ε < l /2 ή ε < 0.01 / 2 = 0.005

2) Διατηρώντας σταθερή την απόσταση από τη διχοτόμο ε = 0.001 προκύπτουν οι εξής γραφικές παραστάσεις που υποδηλώνουν τη σχέση μεταξύ του συνολικού αριθμού φορών που χρειάστηκε να υπολογιστεί η εκάστοτε f(x) και των διαφόρων τιμών εύρους αναζήτησης l.

(Σε κάθε κλήση της μεθόδου έχω 2κ + 1 υπολογισμούς της f. Σε κάθε επανάληψη αλλάζουν και το x1 και το x2, άρα έχω 2κ υπολογσιμούς και στο τέλος για την επιστροφή του $f(x_{min})$ έχω επίσης έναν.)

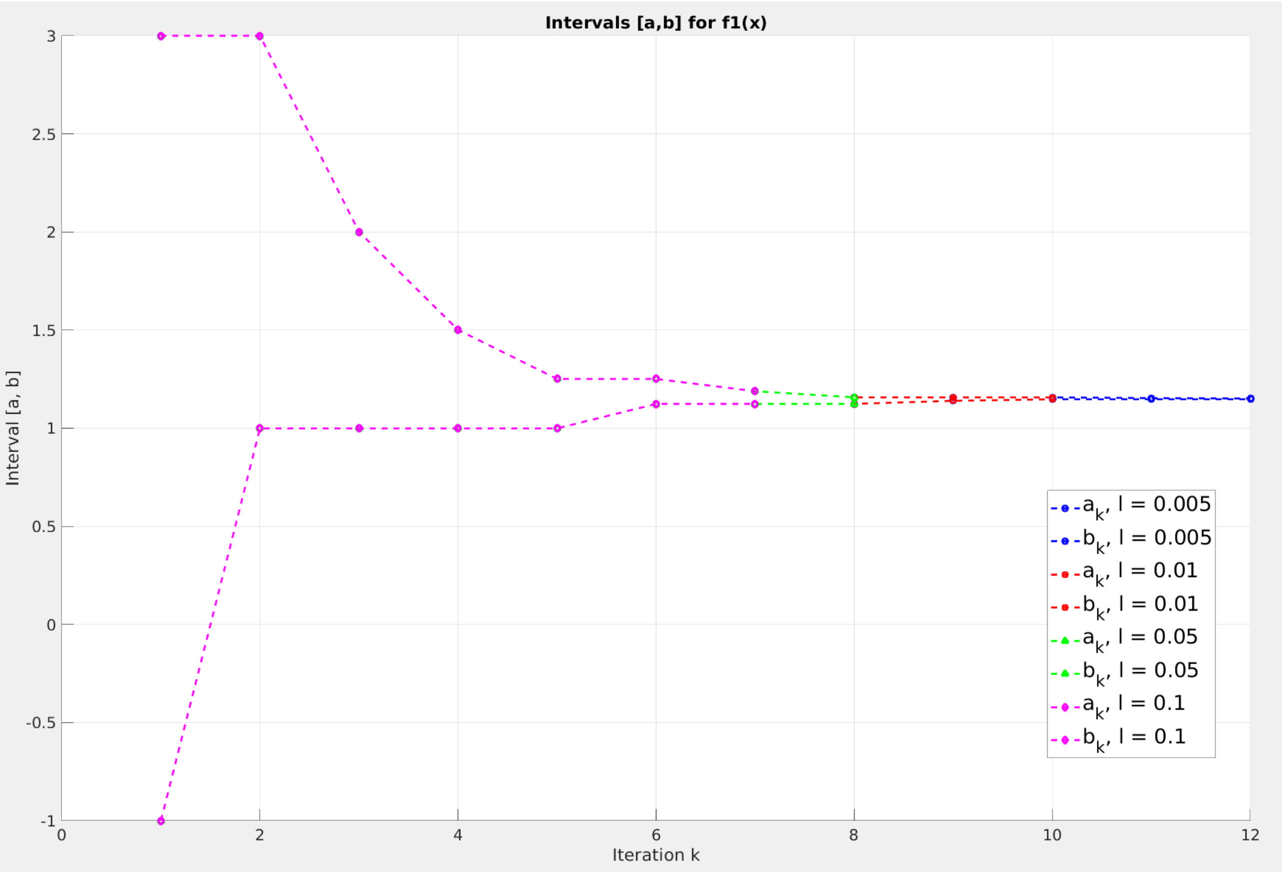


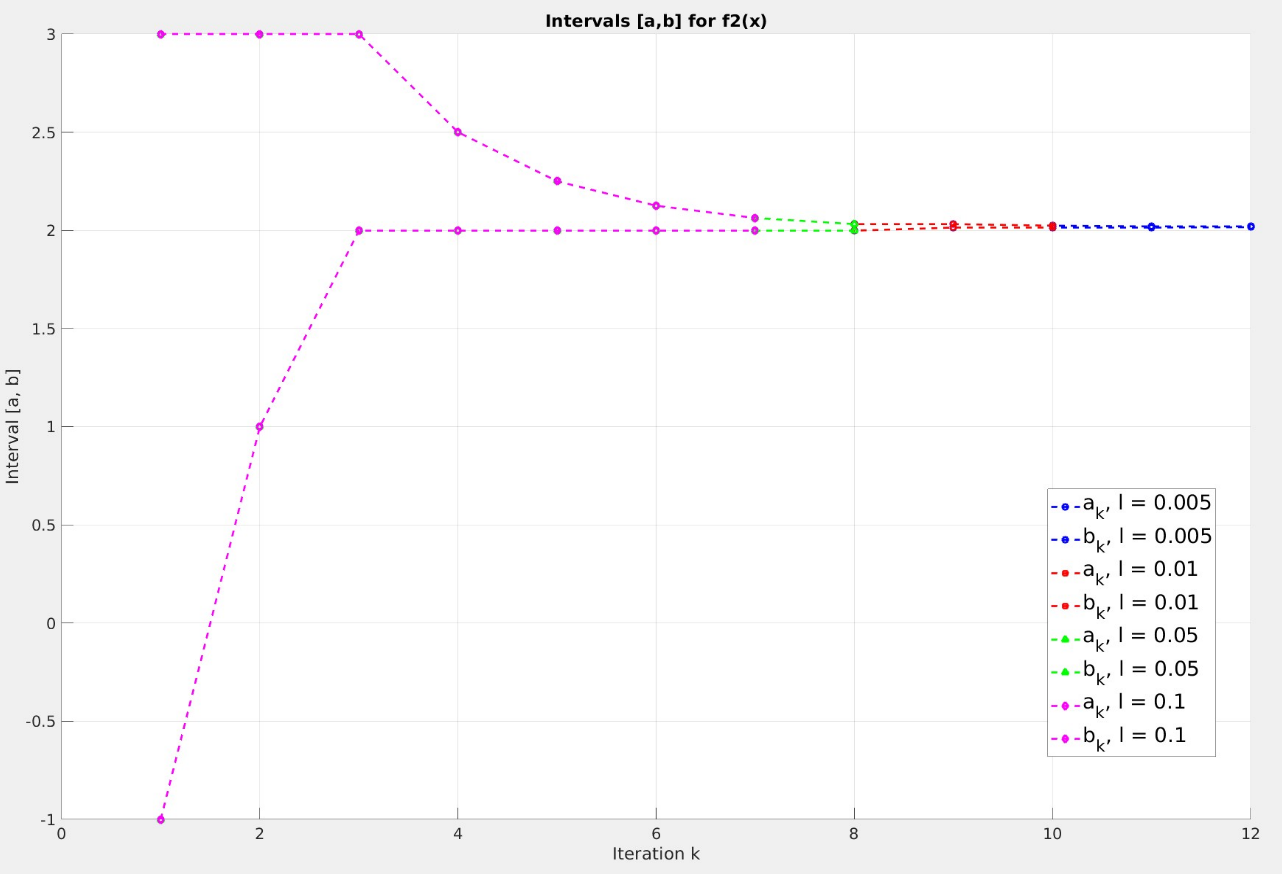

Times f2(x) has been calculated vs Precision
f2(x) calculations
Precision (l)
0
5
10
15
20
25
30
35
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

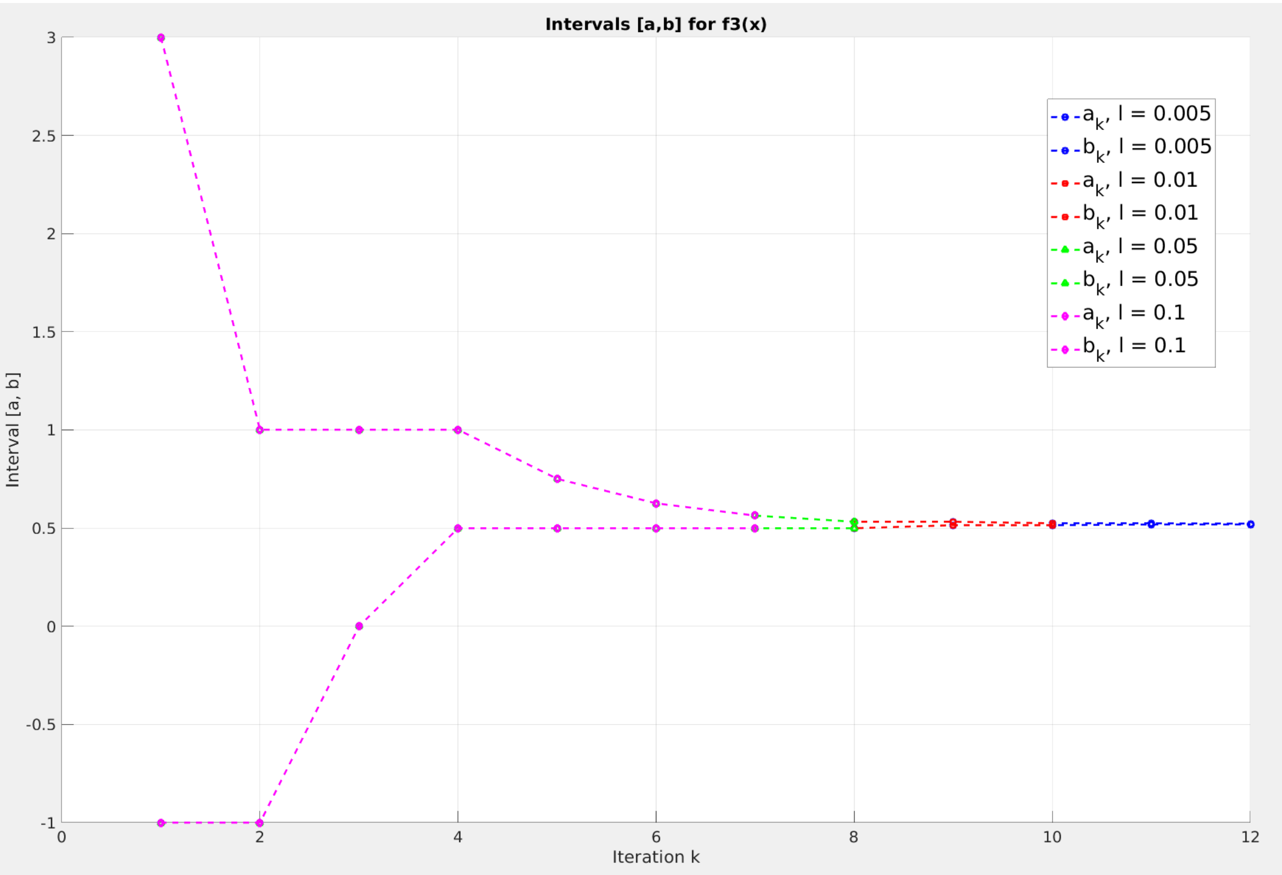

Times f3(x) has been calculated vs Precision
f3(x) calculations
Precision (l)
0
5
10
15
20
25
30
35
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

Μελετώντας τα διαγράμματα, ομοίως με πρίν παρατηρούμε ότι και στις 3 περιπτώσεις (δηλαδή και για τις 3 συναρτήσεις), η **μέθοδος της Διχοτόμου** μας δίνει ίδια ή σχεδόν ίδια αποτελέσματα.

3) Διατηρώντας και πάλι σταθερή την απόσταση από τη διχοτόμο ε = 0.001 θα βρούμε πως εξαρτώνται τα άκρα του διαστήματος $[\alpha_\kappa, b_\kappa]$ συναρτήσει του δείκτη επαναλήψεων κ για διάφορες τιμές του εύρους αναζήτησης l. Οι τιμές του l που επέλεξα να εξετάσω είναι οι εξής: l = 0.005, l = 0.01, l = 0.05, και l = 0.1. Έτσι, προκύπτουν οι γραφικές παραστάσεις:




Intervals [a,b] for f2(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1


Intervals [a,b] for f3(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1

Αυτή τη φορά υπάρχει ξεκάθαρη ανομοιομορφία μεταξύ των διαγραμμάτων διαφορετικών f για σταθερό l, όπως και περιμέναμε. Παρατηρώ, επίσης ότι ενώ μεταβάλλω την τιμή του l, τα διαστήματα $[\alpha_\kappa,\beta_\kappa]$, παραμένουν τα ίδια σε κάθε περίπτωση.

<u>Σύντομη επεξήγηση της μεθόδου:</u>
Η βασική ιδέα της μεθόδου είναι να υποδιαιρεί σταδιακά το διάστημα σε μικρότερα τμήματα, συγκρίνοντας τιμές της συνάρτησης κοντά στο κέντρο του διαστήματος, ώστε να εντοπιστεί το σημείο στο οποίο η συνάρτηση παίρνει την ελάχιστη τιμή της. Το ϵ επιτρέπει στην μέθοδο να συγκρίνει τιμές πολύ κοντά στο κέντρο, αλλά η επιλογή του πρέπει να είναι αρκετά μικρή για να εξασφαλιστεί ακρίβεια, χωρίς να επηρεάζει σημαντικά την ταχύτητα της μεθόδου. Σε κάθε επανάληψη:

1. Υπολογίζονται δύο σημεία, x1 και x2, συμμετρικά τοποθετημένα γύρω από το κέντρο του διαστήματος με μια πολύ μικρή απόσταση 2ϵ μεταξύ τους.

2. Ελέγχεται ποιο από τα δύο σημεία έχει μικρότερη τιμή της συνάρτησης:

- Αν f(x1) < f(x2), τότε το διάστημα μειώνεται σε [a,x2] γιατί το ελάχιστο βρίσκεται πιο αριστερά.
- Αν f(x2) < f(x1), τότε το διάστημα μειώνεται σε [x1,β], γιατί το ελάχιστο βρίσκεται πιο δεξιά.

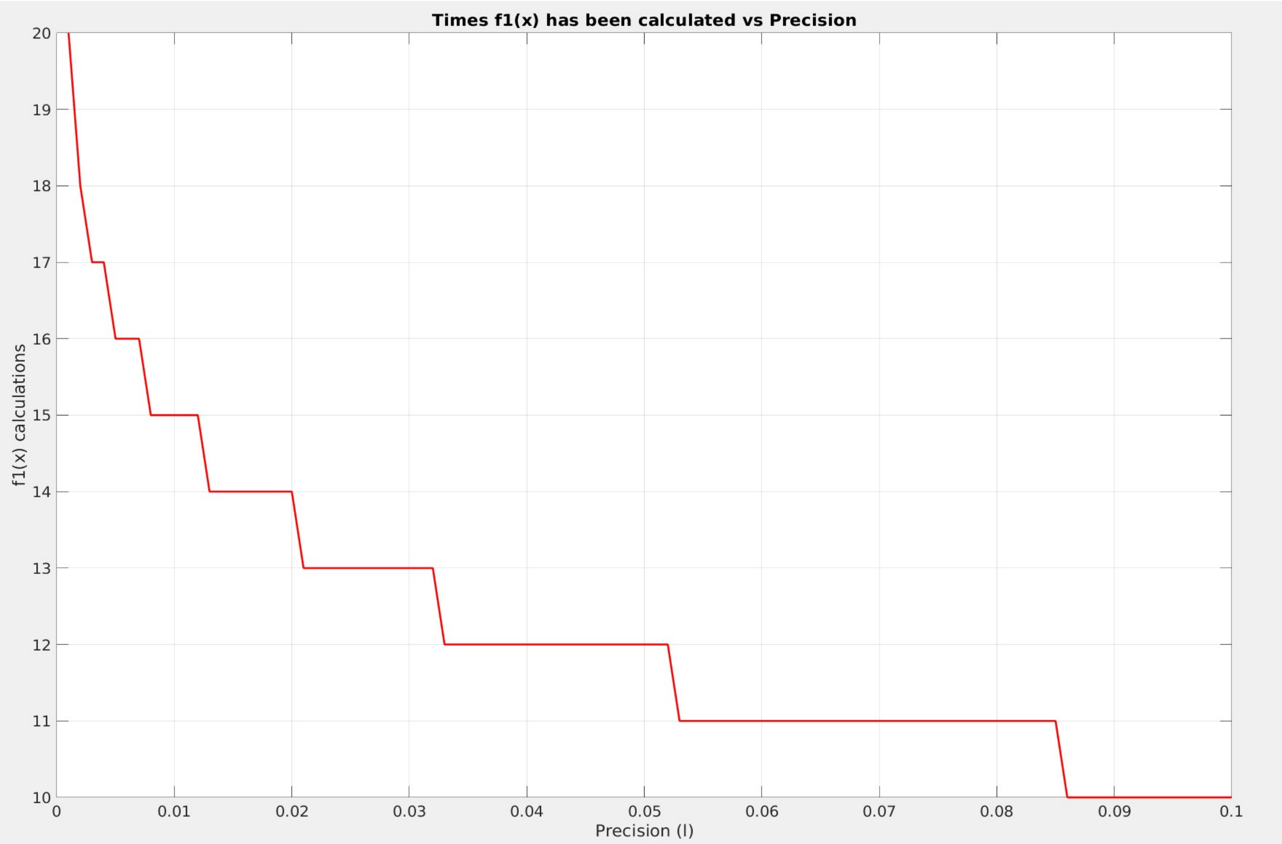
Αυτή η διαδικασία επαναλαμβάνεται έως ότου το διάστημα γίνει μικρότερο από μια προκαθορισμένη ακρίβεια l (δηλαδή, $|b-a| < l$), οπότε η μέθοδος σταματά.
Η μέθοδος είναι απλή και εγγυάται τη σύγκλιση στο ελάχιστο, αρκεί η συνάρτηση να είναι συνεχής και να ισχύει ότι $2*\varepsilon < l$. Ωστόσο, σε σύγκριση με άλλες μεθόδους, μπορεί να είναι αργή, ειδικά αν το διάστημα είναι μεγάλο. Τέλος, απαιτεί ένα αρχικό διάστημα όπου το ελάχιστο είναι γνωστό ή πιθανό να βρίσκεται.

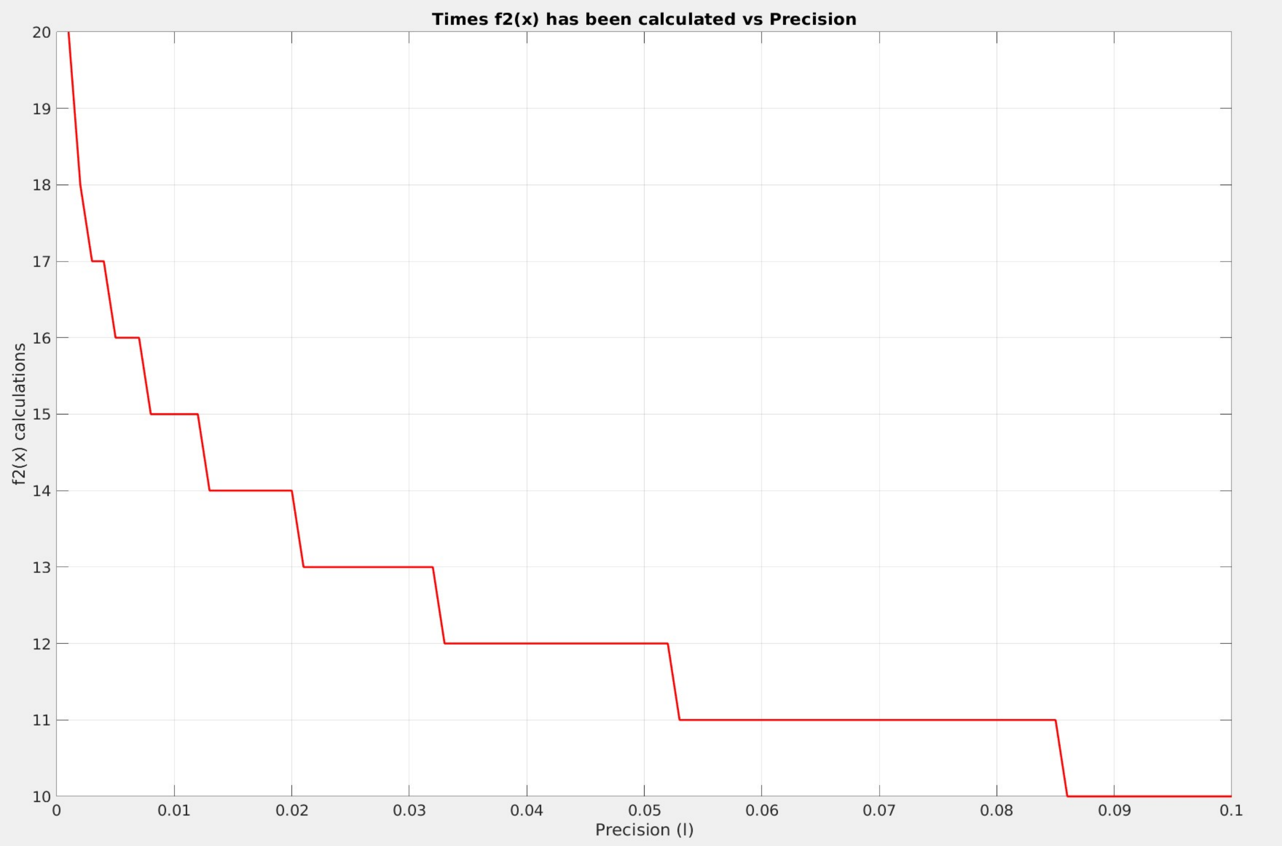
Αναλυτική επεξήγηση του κώδικα υπάρχει στο work1_ex1.m matlab αρχείο.

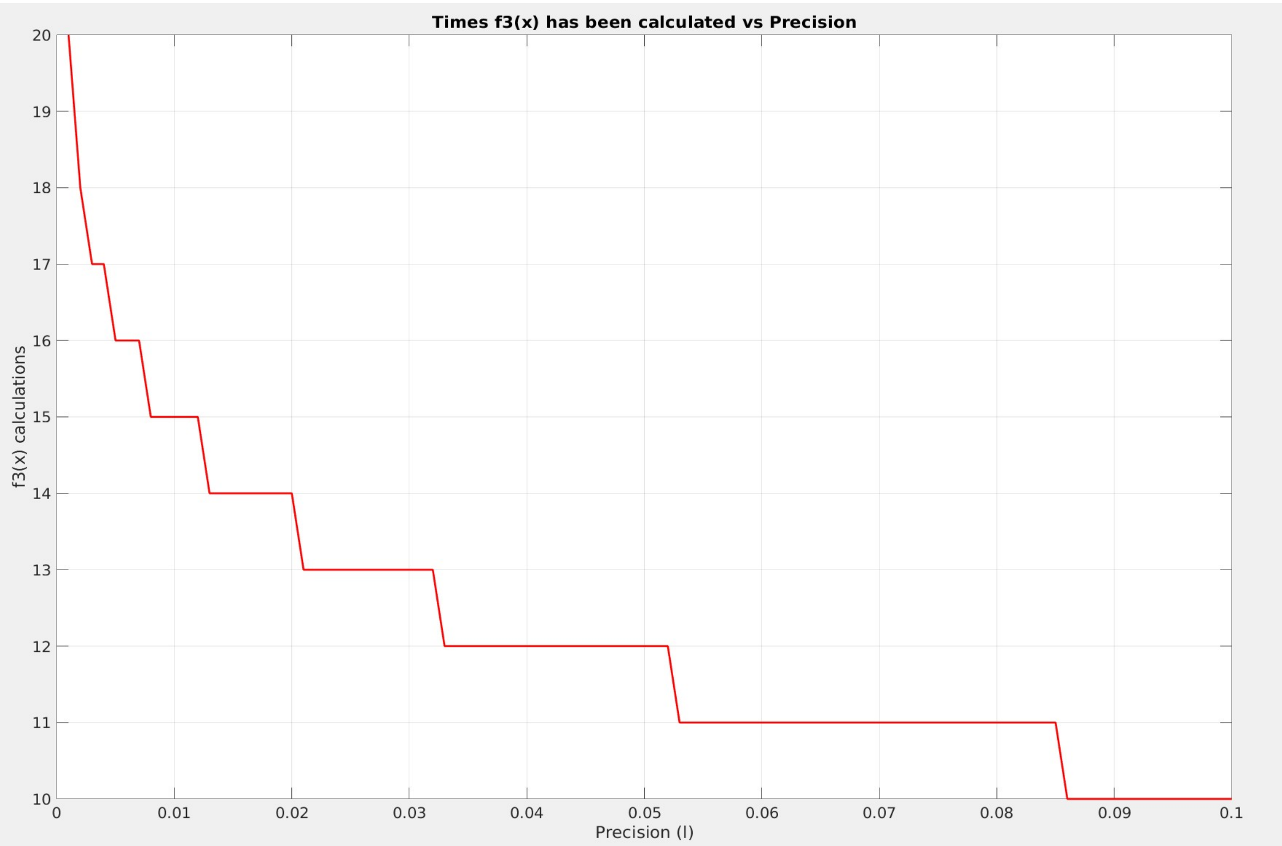
Θέμα 2:

1) Μεταβάλλοντας τις διάφορες τιμές του τελικού εύρους αναζήτησης l προκύπτουν τα εξής διαγράμματα που υποδεικνύουν τη σχέση μεταξύ του συνολικού αριθμού που πρέπει να υπολογιστεί η εκάστοτε f(x) και του εύρους αναζήτησης l.

(Σε κάθε κλήση της μεθόδου εχω κ + 2 υπολογισμούς της f. Στην πρώτη επανάληψη έχω δύο, σε όλες τις επόμενες έχω έναν, άρα κ, και στο τέλος για την επιστροφή του f(x_{min}) έχω επίσης έναν.)

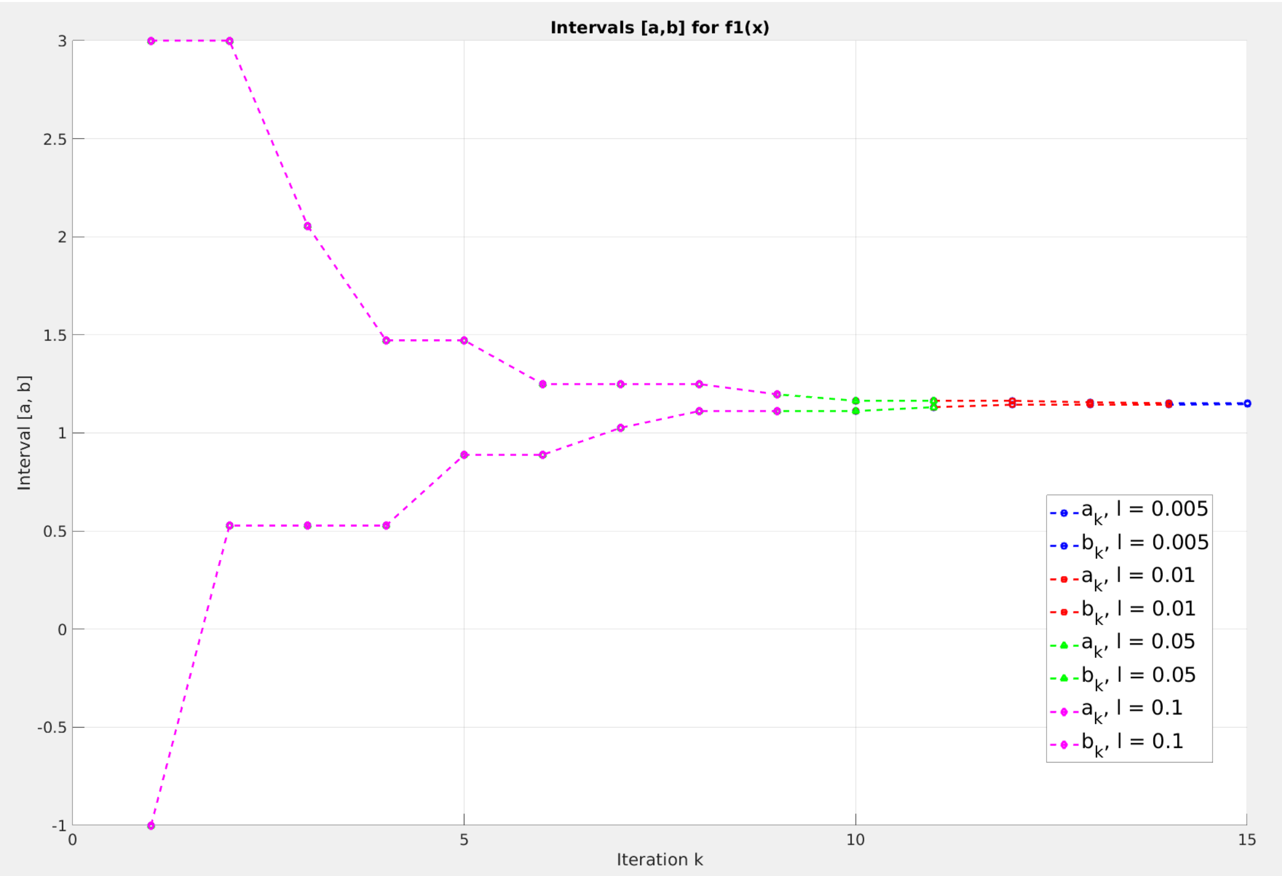


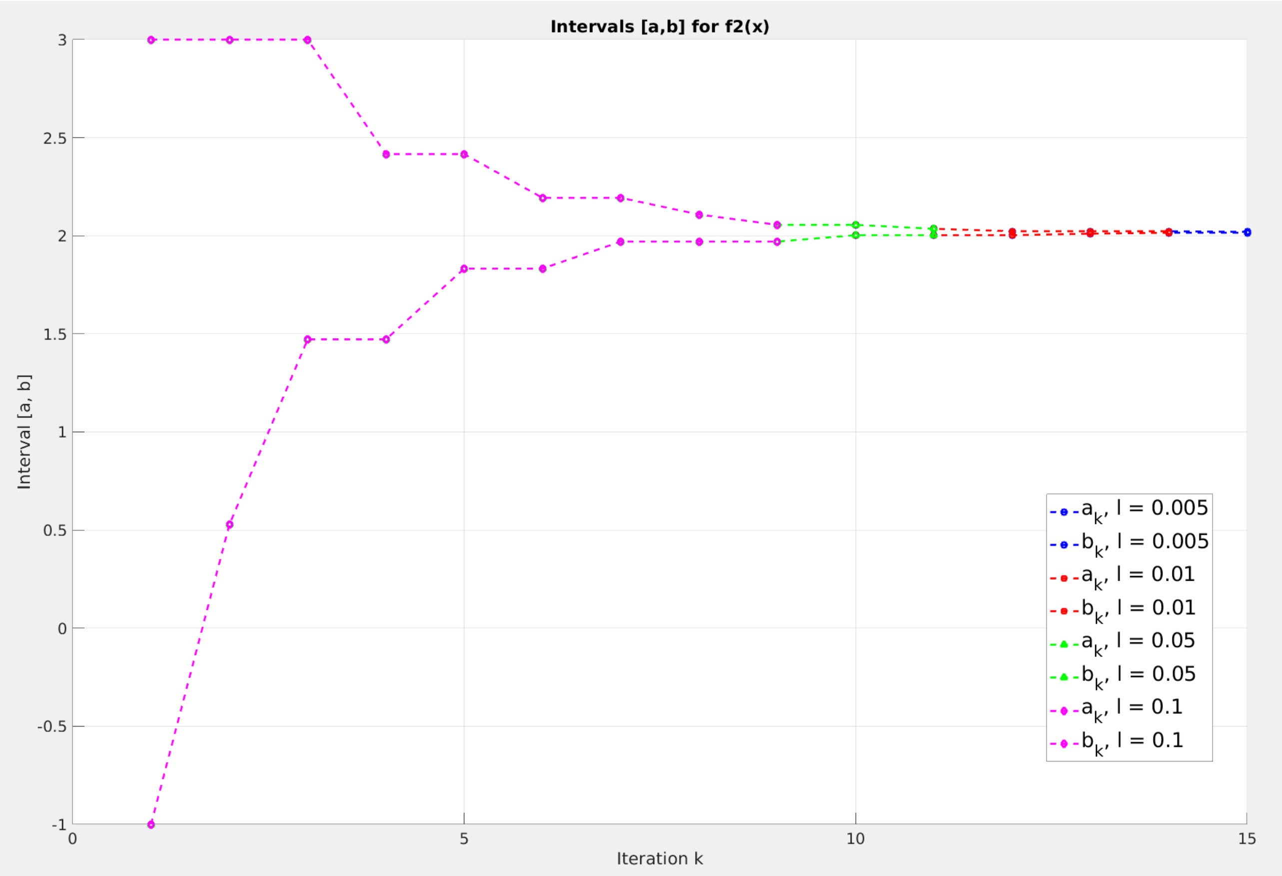

Times f2(x) has been calculated vs Precision
f2(x) calculations
Precision (l)
20
19
18
17
16
15
14
13
12
11
10
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1


Times f3(x) has been calculated vs Precision
f3(x) calculations
Precision (l)
20
19
18
17
16
15
14
13
12
11
10
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

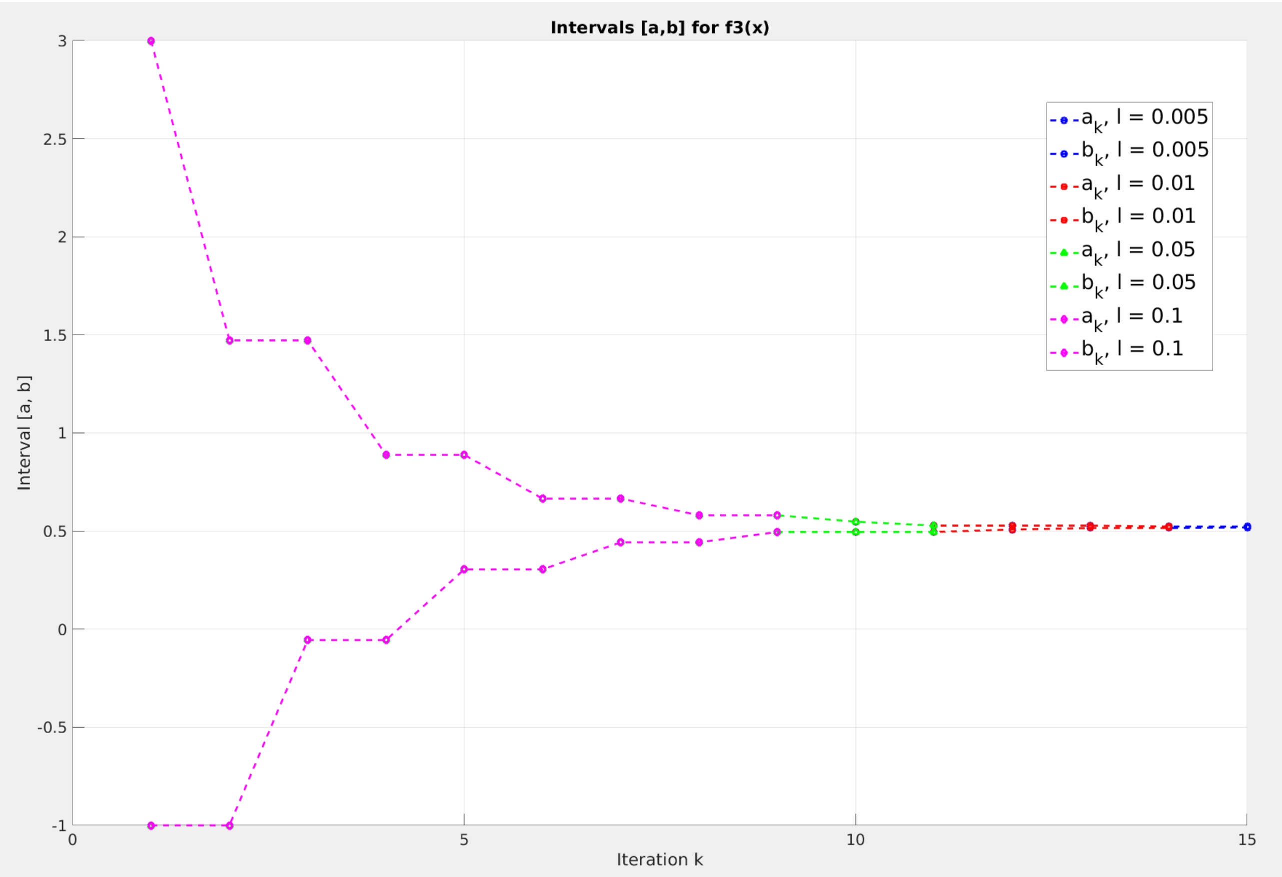
Με βάσει τα διαγράμματα μπορούμε να παρατηρήσουμε ότι και στις 3 περιπτώσεις (δηλαδή και για τις 3 συναρτήσεις), η **μέθοδος του Χρυσού Τομέα** μας δίνει ίδια ή σχεδόν ίδια αποτελέσματα.

2) Θα βρούμε πως εξαρτώνται τα άκρα του διαστήματος $[\alpha_\kappa, b_\kappa]$ συναρτήσει του δείκτη επαναλήψεων κ για διάφορες τιμές του εύρους αναζήτησης l. Οι τιμές του l που επέλεξα να εξετάσω είναι οι εξής: l = 0.005, l = 0.01, l = 0.05, και l = 0.1. Έτσι, προκύπτουν οι γραφικές παραστάσεις:





Intervals [a,b] for f2(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1



Intervals [a,b] for f3(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1

Αυτή τη φορά υπάρχει ξεκάθαρη ανομοιομορφία μεταξύ των διαγραμμάτων διαφορετικών f για σταθερό l, όπως και περιμέναμε. Παρατηρώ, επίσης ότι ενώ μεταβάλλω την τιμή του l, τα διαστήματα $[\alpha_\kappa,\beta_\kappa]$, παραμένουν τα ίδια σε κάθε περίπτωση.

Σύντομη επεξήγηση της μεθόδου:
Η βασική ιδέα της **μεθόδου του χρυσού τομέα** είναι να υποδιαιρεί σταδιακά το διάστημα [a,b] σε μικρότερα τμήματα, χρησιμοποιώντας την σταθερά αναλογίας γ για να εντοπίσει το σημείο όπου η συνάρτηση παίρνει την ελάχιστη τιμή της. Συγκεκριμένα, αντί να επιλέγονται δύο σημεία κοντά στο κέντρο, όπως στη μέθοδο διχοτόμου, η μέθοδος του χρυσού τομέα υπολογίζει δύο σημεία x1 και x2 εντός του διαστήματος με βάση τον λόγο γ=0.618, ώστε η σύγκριση να είναι πιο αποδοτική. Σε κάθε επανάληψη:

1. Υπολογίζονται δύο σημεία, x1 και x2, ως εξής:
   $x1=a+(1-\gamma)\cdot(b-a)$ και $x2=a+\gamma\cdot(b-a)$

   Αυτά τα σημεία διατηρούν την αναλογία του χρυσού τομέα και επιτρέπουν στη μέθοδο να επαναχρησιμοποιεί το ένα από τα δύο σε κάθε βήμα, χωρίς να χρειάζεται να τα υπολογίζει από την αρχή.

2. Ελέγχεται ποιο από τα δύο σημεία έχει μικρότερη τιμή της συνάρτησης:

   - Αν $f(x1)<f(x2)$, τότε το διάστημα μειώνεται σε [a,x2] γιατί το ελάχιστο βρίσκεται πιο αριστερά.
   - Αν $f(x2)<f(x1)$, τότε το διάστημα μειώνεται σε [x1,b], γιατί το ελάχιστο βρίσκεται πιο δεξιά.

Αυτή η διαδικασία επαναλαμβάνεται έως ότου το μήκος του διαστήματος γίνει μικρότερο από μια προκαθορισμένη ακρίβεια l (δηλαδή, $|b-a|<l$), οπότε η μέθοδος σταματά.

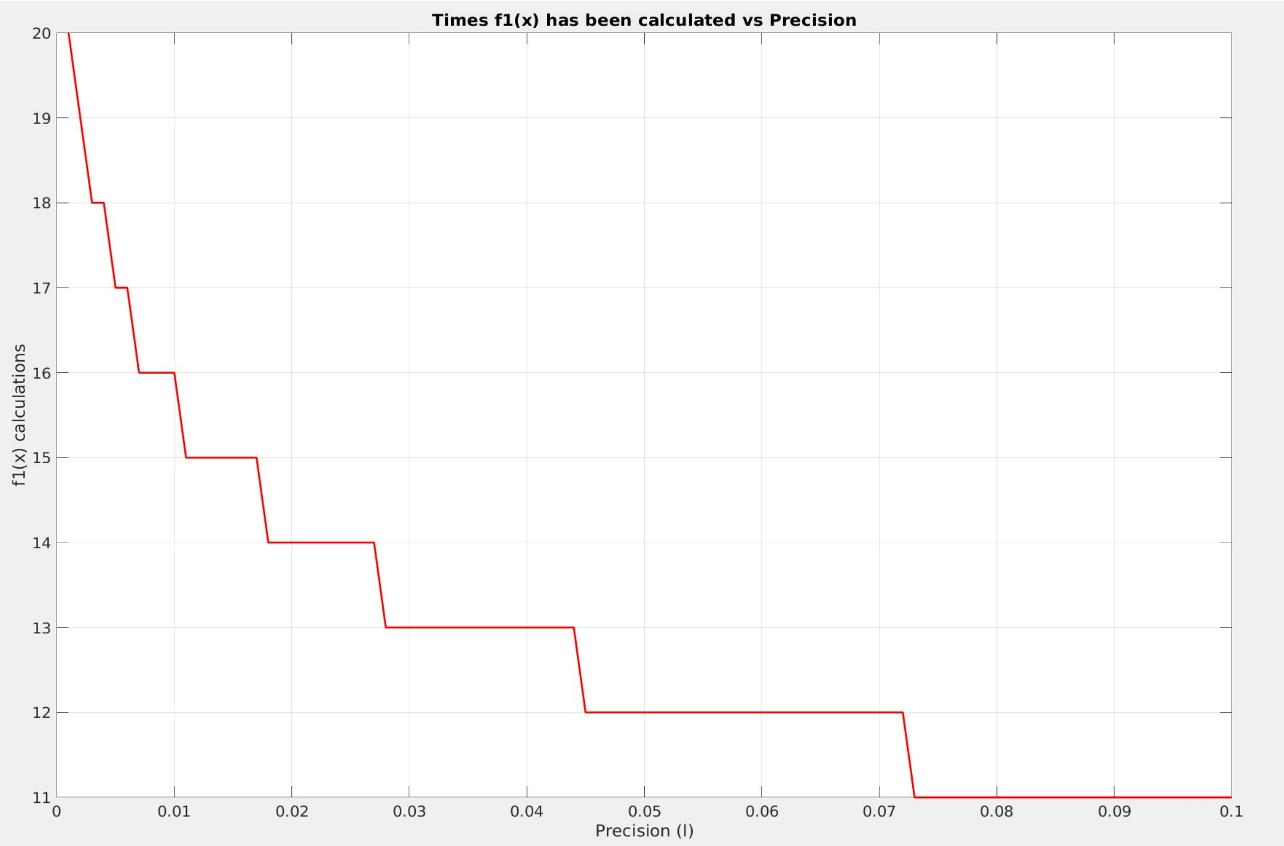
Η μέθοδος του χρυσού τομέα είναι αποδοτική και συγκλίνει γρήγορα στο ελάχιστο, καθώς χρησιμοποιεί τη σταθερά γ για να μειώνει το διάστημα αναζήτησης χωρίς επανυπολογισμούς, γεγονός που την καθιστά πιο αποδοτική από άλλες μεθόδους, όπως τη μέθοδο της διχοτόμου. Επιπλέον, είναι αξιόπιστη για συνεχείς και μονότονες συναρτήσεις. Ωστόσο, παρουσιάζει και μειονεκτήματα, όπως η απαίτηση να υπάρχει μοναδικό ελάχιστο στο αρχικό διάστημα, αλλιώς μπορεί να αποτύχει στον προσδιορισμό της ορθής λύσης. Επίσης, σε περιπτώσεις πολύπλοκων ή μη συνεχών συναρτήσεων, η απόδοσή της είναι περιορισμένη.

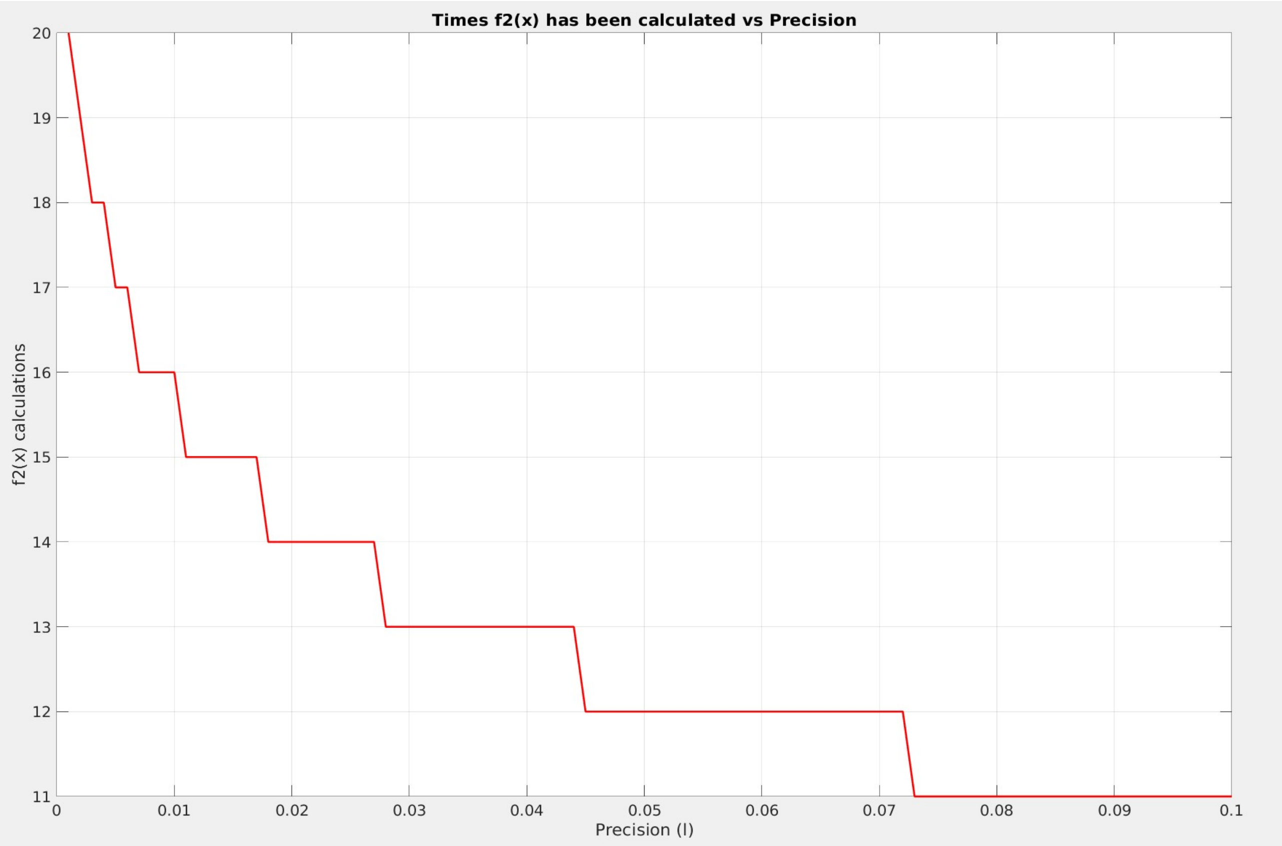
Αναλυτική επεξήγηση του κώδικα υπάρχει στο work1_ex2.m matlab αρχείο.

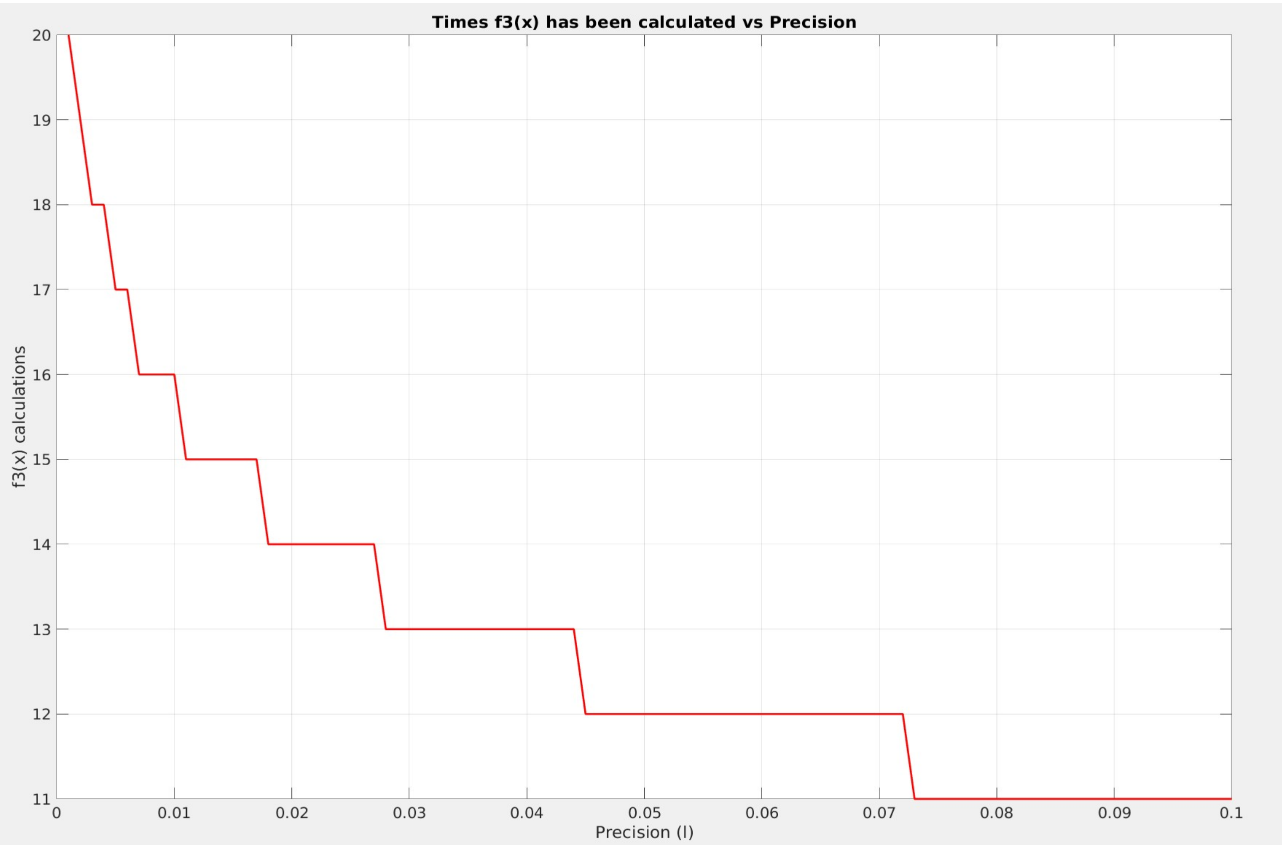
<u>Θέμα 3:</u>

1) Διατηρώντας σταθερή την απόσταση ε = 0.001 και μεταβάλλοντας τις διάφορες τιμές του τελικού εύρους αναζήτησης l προκύπτουν τα εξής διαγράμματα που υποδεικνύουν τη σχέση μεταξύ του συνολικού αριθμού που πρέπει να υπολογιστεί η εκάστοτε f(x) και του εύρους αναζήτησης l.

(Σε κάθε κλήση της μεθόδου εχω κ + 3 υπολογισμούς της f. Δύο στην αρχικοποίηση, έναν σε κάθε επανάληψη, άρα κ, και έναν στο τέλος για την επιστροφή του $f(x_{min})$)

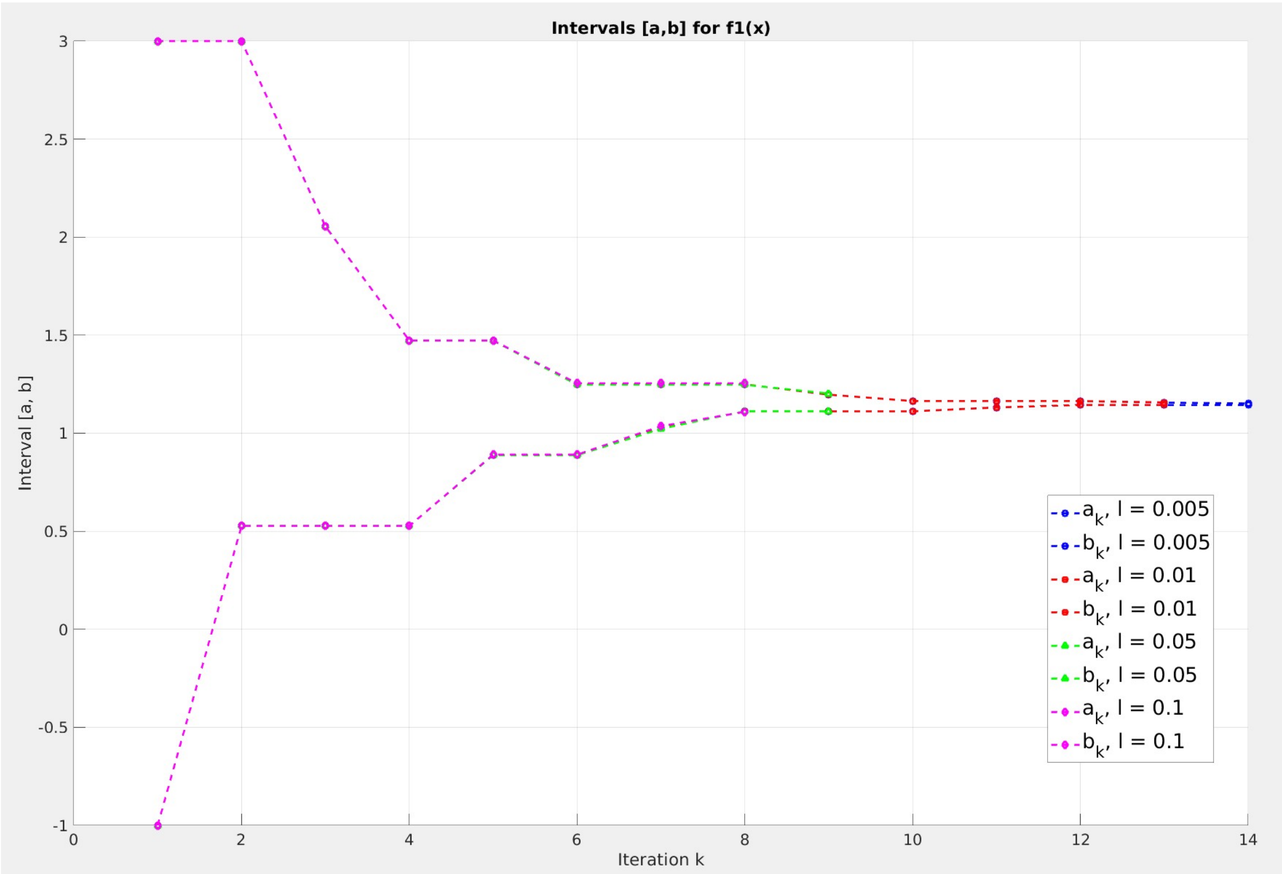


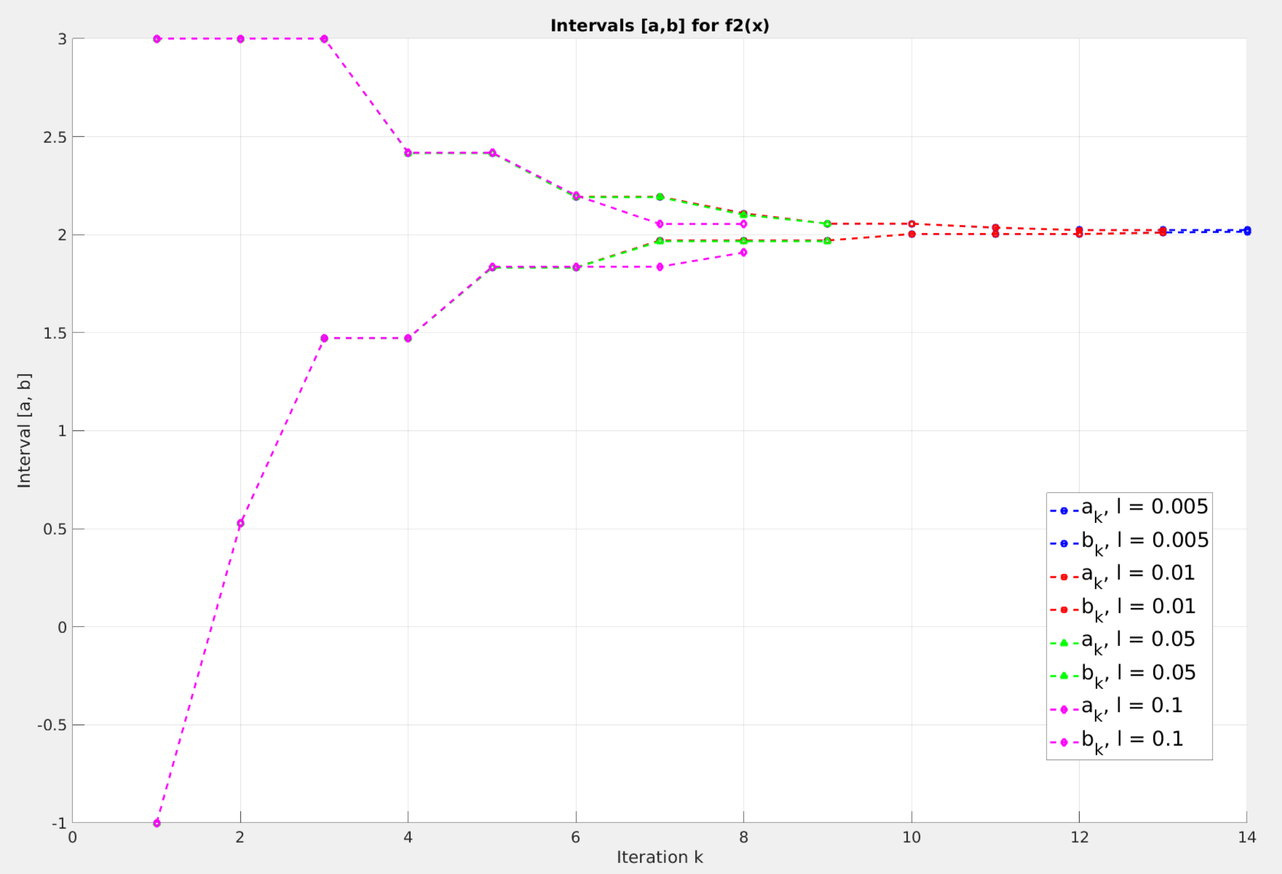

Times f2(x) has been calculated vs Precision
f2(x) calculations
Precision (l)
20
19
18
17
16
15
14
13
12
11
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

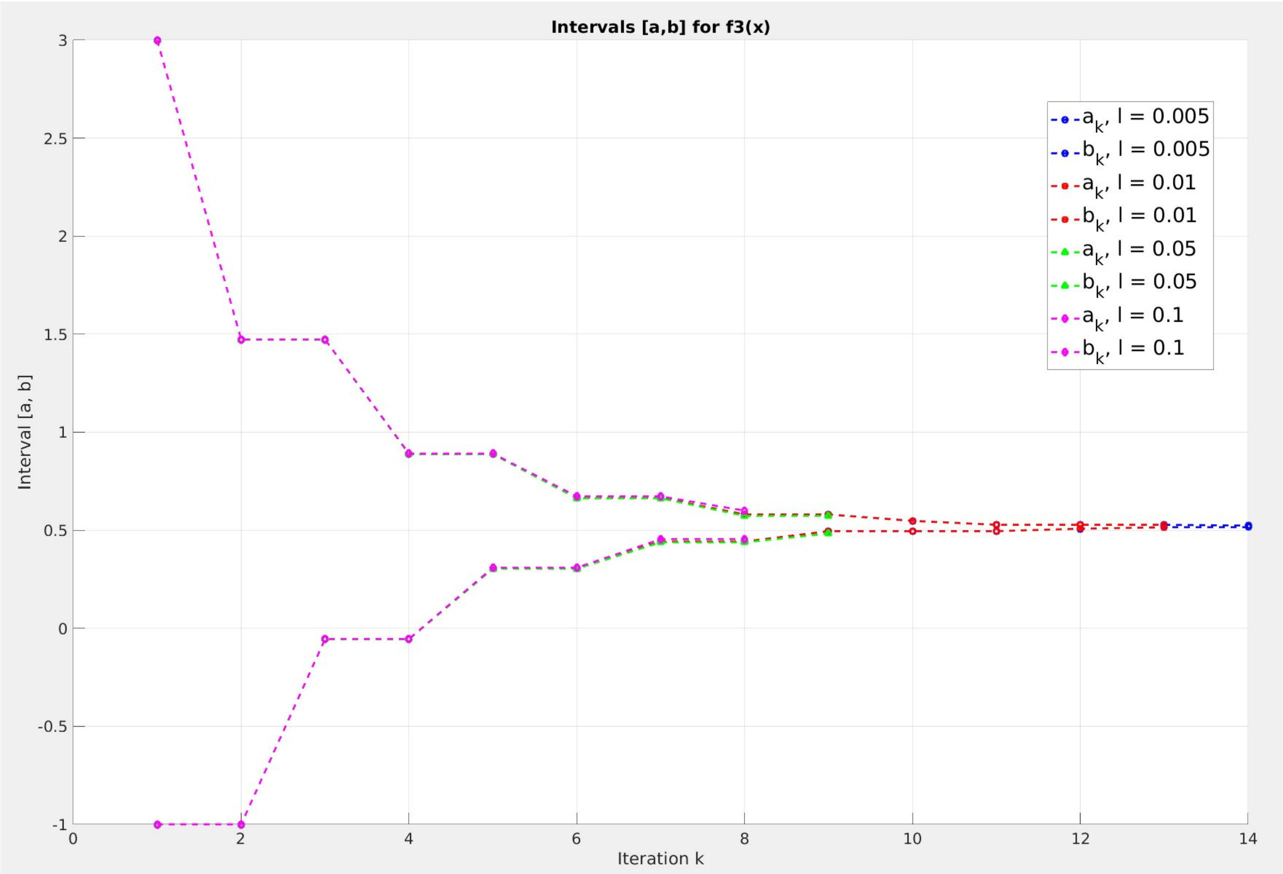

Times f3(x) has been calculated vs Precision
f3(x) calculations
Precision (l)
20
19
18
17
16
15
14
13
12
11
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

Με βάσει τα διαγράμματα μπορούμε να παρατηρήσουμε ότι και στις 3 περιπτώσεις (δηλαδή και για τις 3 συναρτήσεις), η **μέθοδος Fibonacci** μας δίνει ίδια ή σχεδόν ίδια αποτελέσματα.

2) Θα βρούμε πως εξαρτώνται τα άκρα του διαστήματος $[\alpha_\kappa, b_\kappa]$ συναρτήσει του δείκτη επαναλήψεων κ για διάφορες τιμές του εύρους αναζήτησης l. Οι τιμές του l που επέλεξα να εξετάσω είναι οι εξής: l = 0.005, l = 0.01, l = 0.05, και l = 0.1. Έτσι, προκύπτουν οι γραφικές παραστάσεις:




Intervals [a,b] for f2(x)
Interval [a, b]
Iteration k
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1


Intervals [a,b] for f3(x)
Interval [a, b]
Iteration k
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1

Αυτή τη φορά υπάρχει ξεκάθαρη ανομοιομορφία μεταξύ των διαγραμμάτων διαφορετικών f για σταθερό l, όπως και περιμέναμε. Παρατηρώ, επίσης ότι μεταβάλλοντας την τιμή του l, τα διαστήματα $[\alpha_\kappa,\beta_\kappa]$, αλλάζουν επίσης.

<u>Σύντομη επεξήγηση της μεθόδου:</u>
Η βασική ιδέα της **μεθόδου Fibonacci** είναι να υποδιαιρεί σταδιακά το διάστημα [a,b] σε μικρότερα τμήματα χρησιμοποιώντας αριθμούς Fibonacci, έτσι ώστε να εντοπιστεί το σημείο όπου η συνάρτηση παίρνει την ελάχιστη τιμή της. Η χρήση της ακολουθίας Fibonacci σε κάθε βήμα βοηθάει στη σύγκριση τιμών της συνάρτησης στα σημεία (x1,x2), τα οποία είναι προκαθορισμένα με βάση τη σχέση Fibonacci και διατηρούνται σε σταθερές αποστάσεις εντός του διαστήματος.

Σε κάθε επανάληψη:

1. Υπολογίζονται τα δύο σημεία x1 και x2, με τη βοήθεια των τιμών της ακολουθίας Fibonacci, ώστε να διατηρηθεί η κατάλληλη αναλογία στο διάστημα:

   x1=a+fib(N)fib(N−2)⋅(b−a)καιx2=a+fib(N)fib(N−1)⋅(b−a)
2. Ελέγχεται ποιο από τα δύο σημεία έχει μικρότερη τιμή της συνάρτησης:
   - Αν f(x1)<f(x2), τότε το διάστημα μειώνεται σε [a,x2], καθώς η ελάχιστη τιμή βρίσκεται αριστερότερα.
   - Αν f(x2)<f(x1), τότε το διάστημα περιορίζεται σε [x1,b], αφού το ελάχιστο εντοπίζεται πιο δεξιά.

Αυτή η διαδικασία επαναλαμβάνεται για N επαναλήψεις.

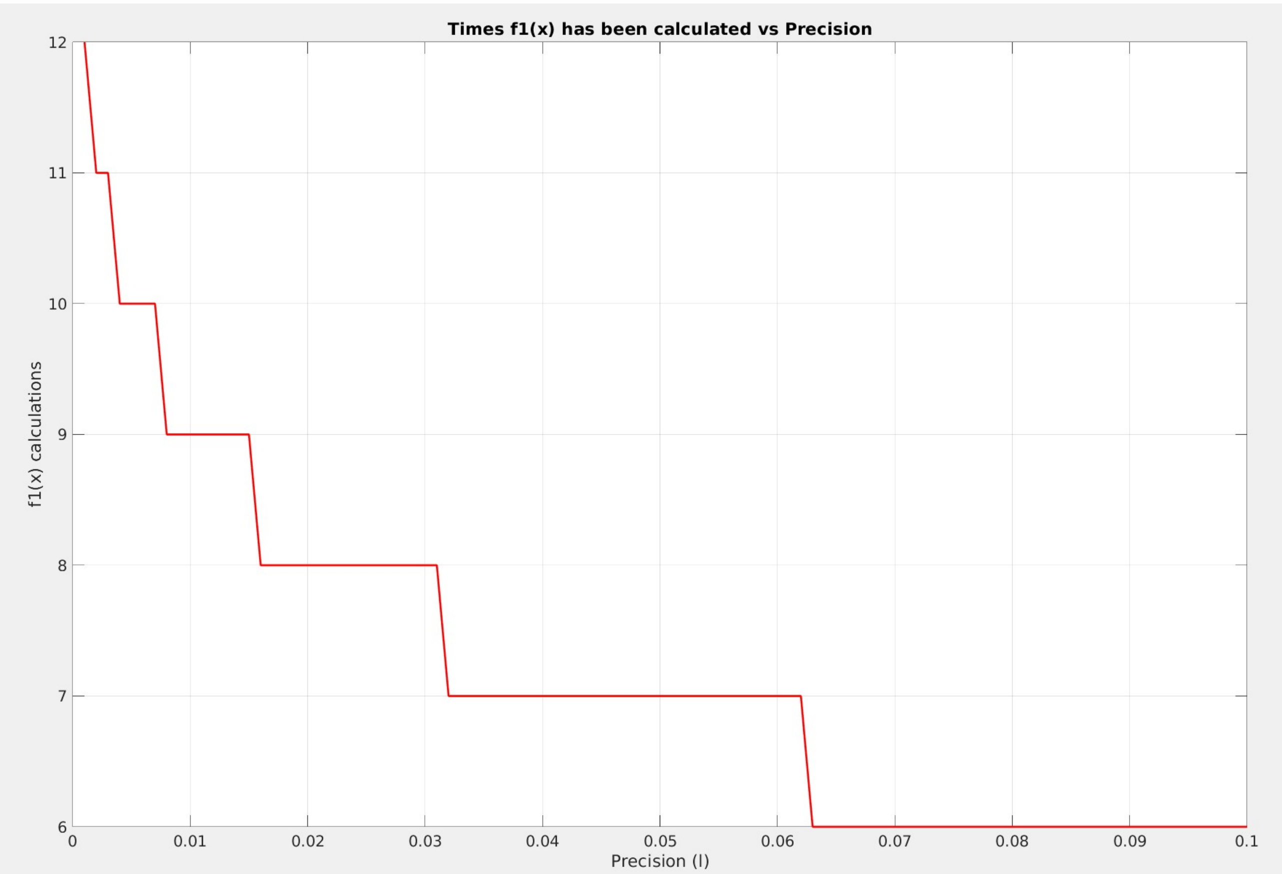
Η μέθοδος Fibonacci έχει αρκετά πλεονεκτήματα, καθώς αξιοποιεί τις αναλογίες της ακολουθίας Fibonacci για να μειώνει το διάστημα αναζήτησης πιο αποδοτικά, συγκριτικά με τη μέθοδο διχοτόμησης. Χάρη στη στρατηγική αυτή, προσφέρει υψηλότερη ταχύτητα σύγκλισης και απαιτεί συνήθως λιγότερους υπολογισμούς, ειδικά όταν το διάστημα είναι μεγάλο. Ωστόσο, παρουσιάζει και κάποια μειονεκτήματα: απαιτεί γνώση του αρχικού αριθμού επαναλήψεων (το N στην ακολουθία), γεγονός που μπορεί να δυσχεραίνει τη χρήση της σε άγνωστες συναρτήσεις. Επίσης, σε συναρτήσεις με πολλά τοπικά ελάχιστα ή μη συνεχείς συναρτήσεις, μπορεί να μη συγκλίνει στο πραγματικό ελάχιστο.

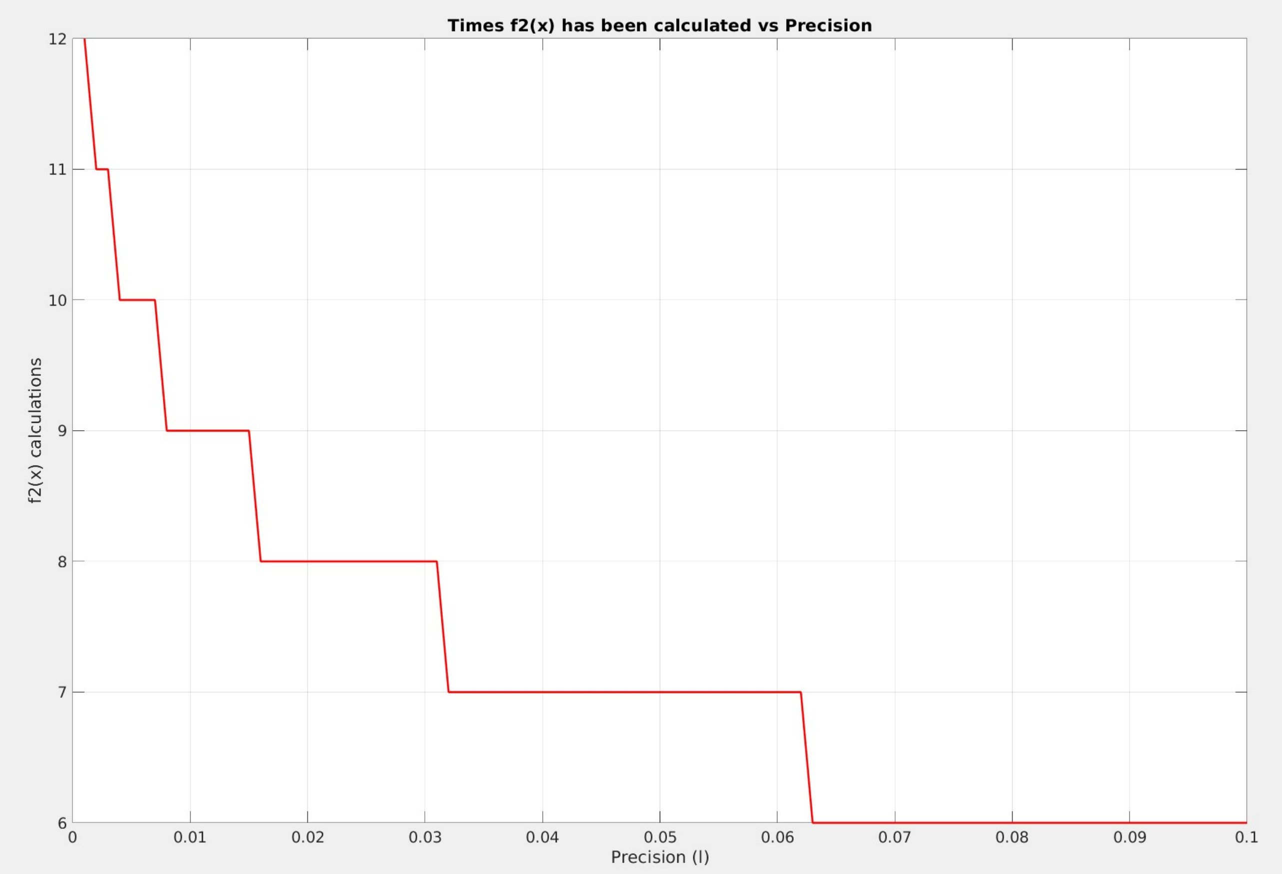
Αναλυτική επεξήγηση του κώδικα υπάρχει στο work1_ex3.m matlab αρχείο.

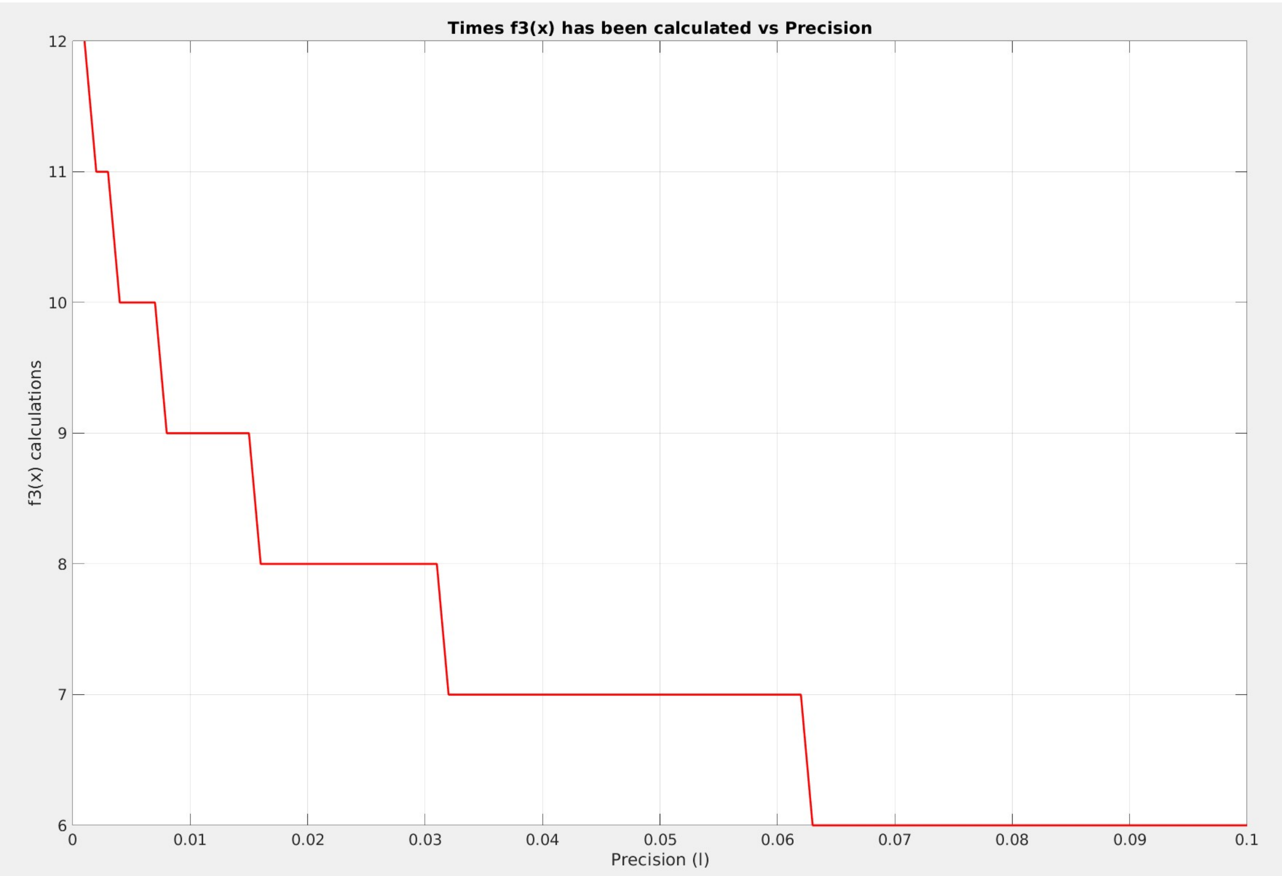
Θέμα 4:

1) Μεταβάλλοντας τις διάφορες τιμές του τελικού εύρους αναζήτησης l προκύπτουν τα εξής διαγράμματα που υποδεικνύουν τη σχέση μεταξύ του συνολικού αριθμού που πρέπει να υπολογιστεί η εκάστοτε f(x) και του εύρους αναζήτησης l.

(Σε κάθε κλήση της μεθόδου εχω έναν υπολογισμό της της f. Επομένως, για να βγουν τα διαγράμματα παρόμοια με τις πάνω περιπτώσεις θα δείξω τη σχέση μεταξύ του συνολικού αριθμού που πρέπει να υπολογιστεί η εκάστοτε df(x) και του εύρους αναζήτησης l. Σε αυτήν την περίπτωση έχω κ υπολογισμούς της df(x), μία σε κάθε επανάληψη, ενώ, στο τέλος για την επιστροφή του $f(x_{min})$ έχω επίσης άλλον έναν, σύνολο κ+1.)

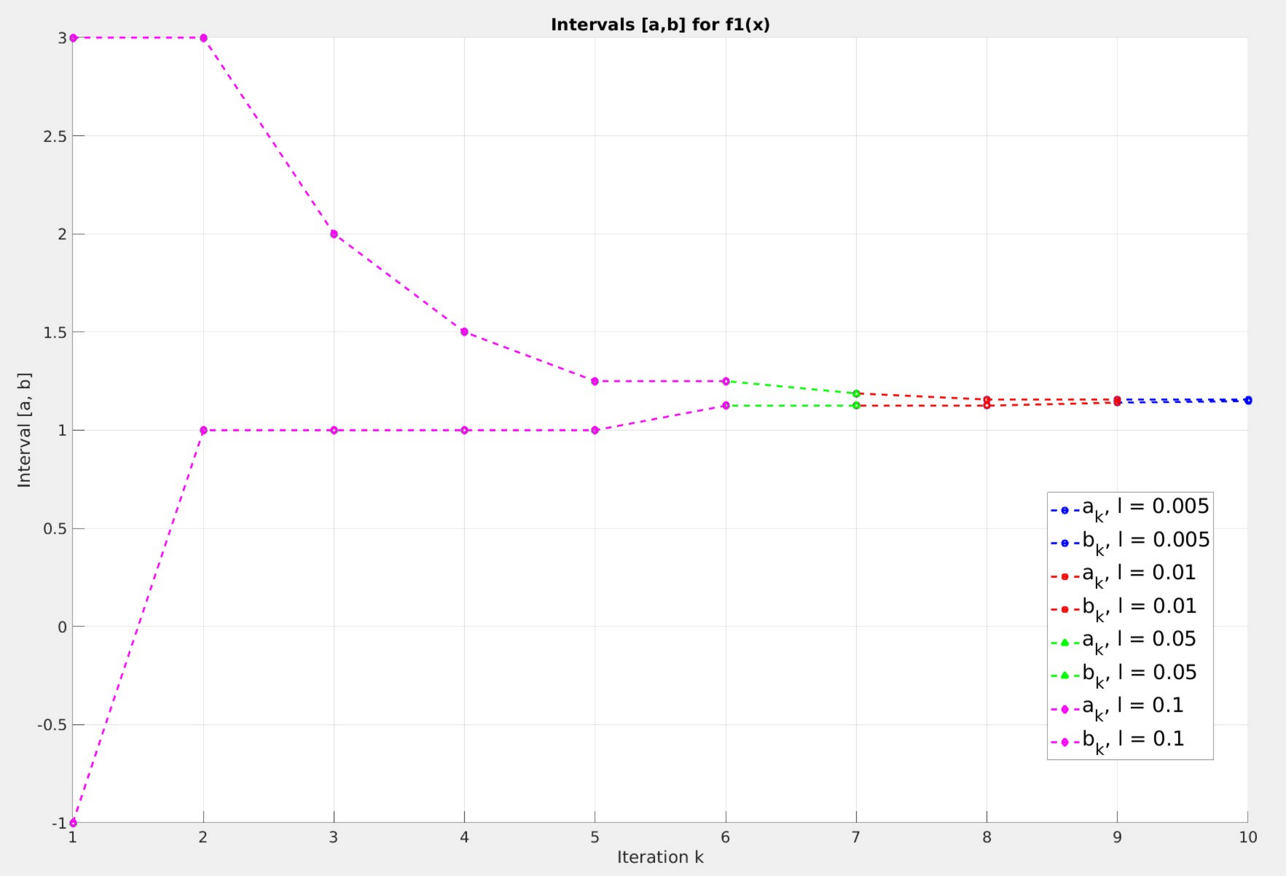


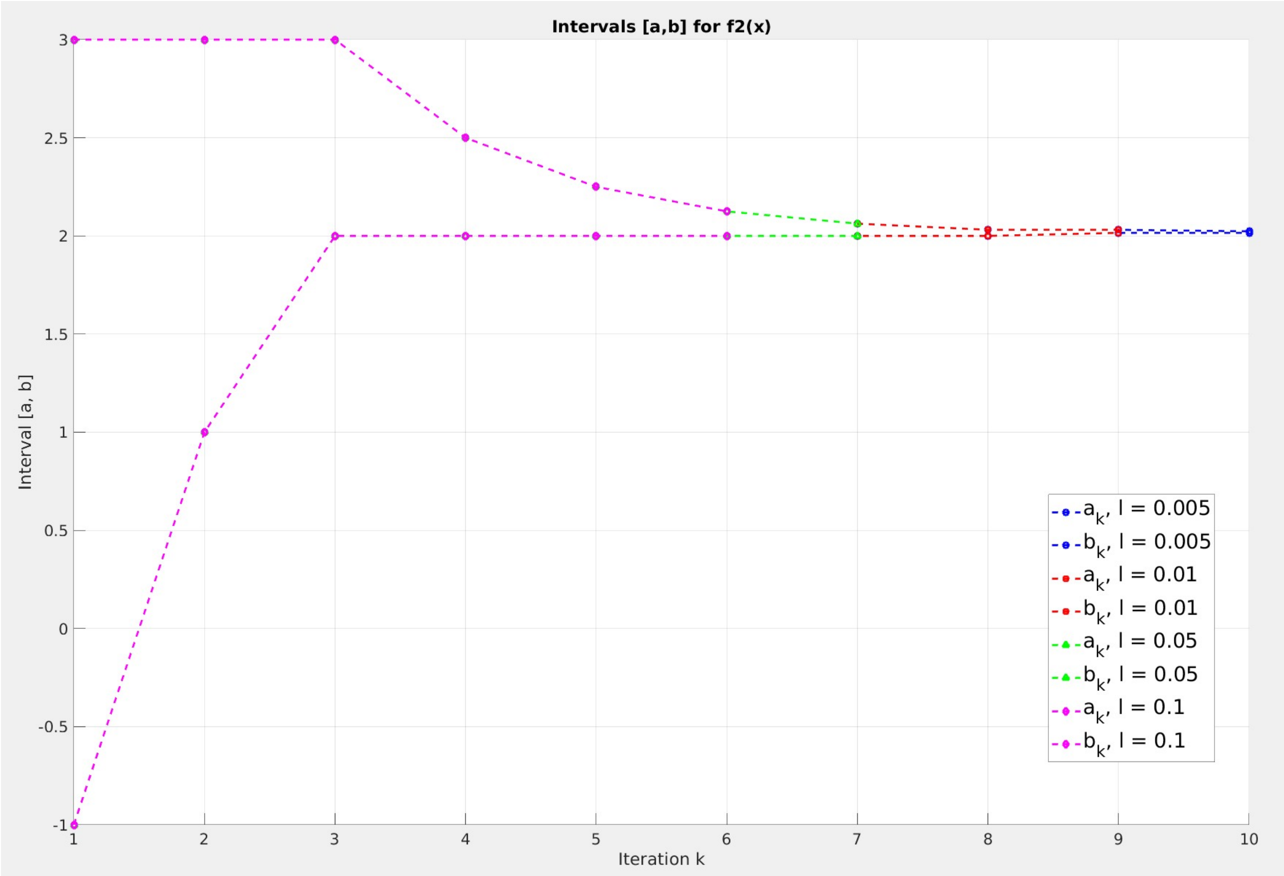

Times f2(x) has been calculated vs Precision
f2(x) calculations
Precision (l)
12
11
10
9
8
7
6
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

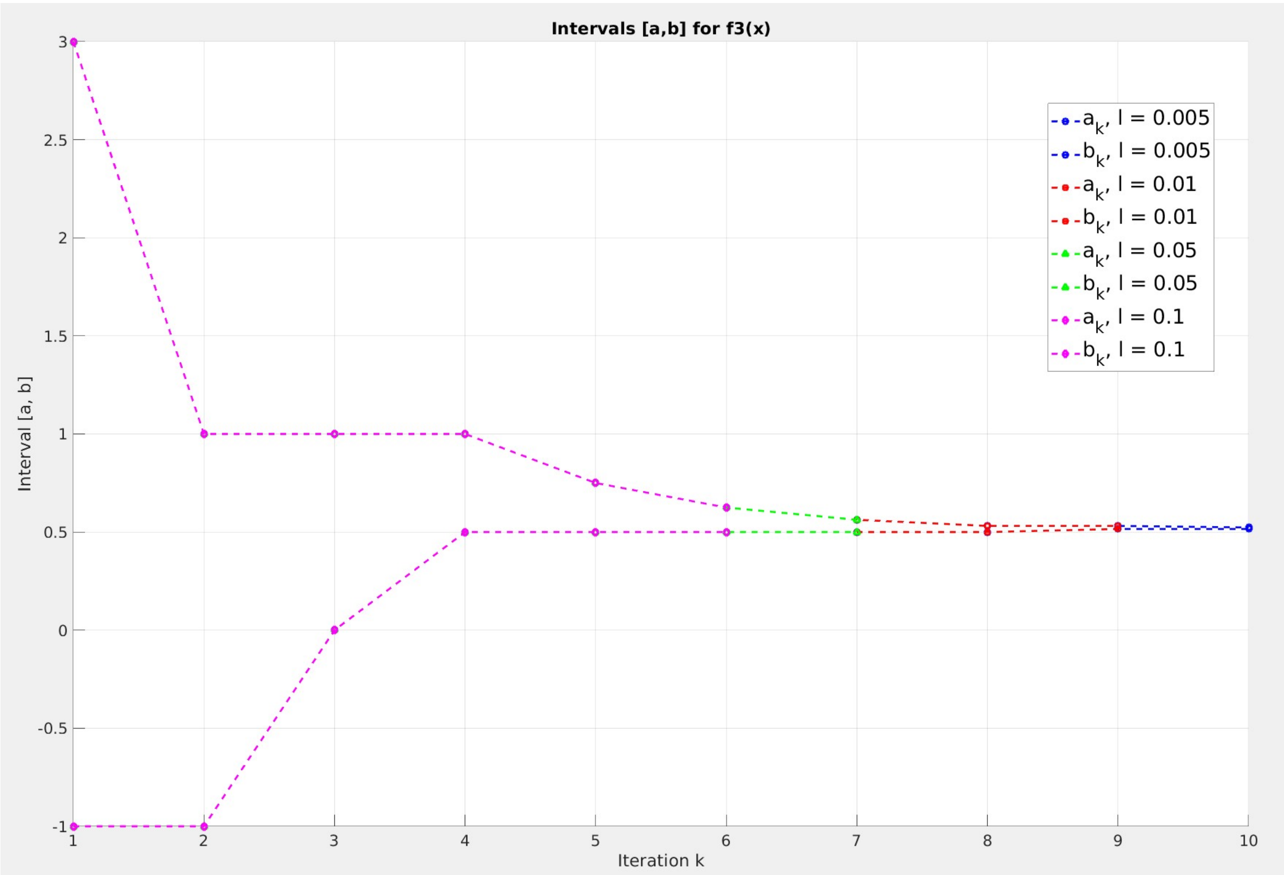

Times f3(x) has been calculated vs Precision
f3(x) calculations
Precision (l)
12
11
10
9
8
7
6
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

Με βάσει τα διαγράμματα μπορούμε να παρατηρήσουμε ότι και στις 3 περιπτώσεις (δηλαδή και για τις 3 συναρτήσεις), η **μέθοδος της Διχοτόμου με χρήση παραγώγου** μας δίνει ίδια ή σχεδόν ίδια αποτελέσματα.

2) Θα βρούμε πως εξαρτώνται τα άκρα του διαστήματος $[\alpha_\kappa, b_\kappa]$ συναρτήσει του δείκτη επαναλήψεων κ για διάφορες τιμές του εύρους αναζήτησης l. Οι τιμές του l που επέλεξα να εξετάσω είναι οι εξής: l = 0.005, l = 0.01, l = 0.05, και l = 0.1. Έτσι, προκύπτουν οι γραφικές παραστάσεις:




Intervals [a,b] for f2(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1


Intervals [a,b] for f3(x)
Interval [a, b]
Iteration k
a_k, l = 0.005
b_k, l = 0.005
a_k, l = 0.01
b_k, l = 0.01
a_k, l = 0.05
b_k, l = 0.05
a_k, l = 0.1
b_k, l = 0.1

Αυτή τη φορά υπάρχει ξεκάθαρη ανομοιομορφία μεταξύ των διαγραμμάτων διαφορετικών f για σταθερό l, όπως και περιμέναμε. Παρατηρώ, επίσης ότι ενώ μεταβάλλω την τιμή του l, τα διαστήματα [α_κ,β_κ], παραμένουν τα ίδια σε κάθε περίπτωση.

Σύντομη επεξήγηση της μεθόδου:

Η βασική ιδέα της **μεθόδου της διχοτόμου με χρήση της παραγώγου** είναι να υποδιαιρεί σταδιακά το διάστημα [a,b] σε μικρότερα τμήματα, χρησιμοποιώντας την παράγωγο της συνάρτησης ώστε να εντοπιστεί το σημείο όπου η συνάρτηση παίρνει την ελάχιστη τιμή της. Σε κάθε επανάληψη, υπολογίζεται το σημείο xmid, δηλαδή το κέντρο του διαστήματος, και στη συνέχεια γίνεται έλεγχος στην παράγωγο της συνάρτησης f′(xmid) στο σημείο αυτό.

Συγκεκριμένα:

1. **Έλεγχος της παραγώγου**: Αν η παράγωγος στο xmid είναι μηδέν, τότε το σημείο xmid είναι στάσιμο και πιθανότατα το ελάχιστο, οπότε το διάστημα γίνεται [xmid,xmid] και η διαδικασία σταματά.
2. **Σύγκριση παραγώγου και μείωση διαστήματος**: Αν η παράγωγος είναι αρνητική (f′(xmid)<0), σημαίνει ότι η συνάρτηση κατεβαίνει προς τα δεξιά, επομένως το ελάχιστο βρίσκεται στο δεξιότερο τμήμα του διαστήματος, οπότε μετακινούμε το αριστερό όριο του διαστήματος στο xmid (θέτουμε a=xmid). Αντίθετα, αν η παράγωγος είναι θετική (f′(xmid)>0), το ελάχιστο βρίσκεται αριστερά, και έτσι μειώνεται το διάστημα από τα δεξιά, με b=xmid.

Αυτή η διαδικασία συνεχίζεται έως ότου ο αριθμός των επαναλήψεων φτάσει στο προκαθορισμένο όριο N, εξασφαλίζοντας ότι το διάστημα [a,b] έχει περιοριστεί επαρκώς γύρω από το σημείο που προσεγγίζει το ελάχιστο της συνάρτησης.

Η μέθοδος της διχοτόμου με χρήση παραγώγου προσφέρει σημαντικά πλεονεκτήματα για την εύρεση του ελάχιστου σημείου μιας συνάρτησης. Χρησιμοποιώντας την παράγωγο, μπορεί να εντοπίσει τη φορά της ελάχιστης τιμής με μεγαλύτερη ακρίβεια και ταχύτητα, μειώνοντας το διάστημα αναζήτησης πιο αποδοτικά από τη συμβατική μέθοδο διχοτόμησης. Ειδικά για συνεχείς και ομαλές συναρτήσεις, η μέθοδος είναι αξιόπιστη και συγκλίνει γρήγορα, προσφέροντας ακρίβεια χωρίς να απαιτεί πολλές επαναλήψεις. Ωστόσο, έχει και ορισμένα μειονεκτήματα: απαιτεί τη γνώση της παραγώγου της συνάρτησης, κάτι που δεν είναι πάντα εύκολο ή πρακτικό για πολύπλοκες συναρτήσεις. Επιπλέον, μπορεί να μην είναι αποτελεσματική για συναρτήσεις που δεν είναι ομαλές ή παρουσιάζουν πολλά τοπικά ελάχιστα, καθώς μπορεί να εγκλωβιστεί σε αυτά και να μην βρει το πραγματικό ελάχιστο του διαστήματος. Τέλος, όμοια με την μέθοδο Fibonacci, απαιτεί γνώση του αρχικού αριθμού επαναλήψεων.

Αναλυτική επεξήγηση του κώδικα υπάρχει στο work1_ex4.m matlab αρχείο.

Σύντομος σχολιασμός πάνω στην αποδοτικότητα των μεθόδων για τις τρεις συναρτήσεις

Από τα διαγράμματα που απεικονίζουν τα εκάστοτε διαστήματα $[\alpha_\kappa,\beta_\kappa]$ συναρτήσει των επαναλήψεων κ, παρατηρώ ότι η μέθοδος με τις περισσότερες συνολικές επαναλήψεις είναι η **μέθοδος του Χρυσού τομέα** (για l = 0.005 έχω 15 επαναλήψεις). Μετά ακολουθεί η **μέθοδος Fibonacci** (για l = 0.005 και ε = 0.001έχω 14 επαναλήψεις). Έπειτα, σειρά έχει η **μέθοδος της Διχοτόμου** (για l = 0.005 και ε = 0.001 έχω 12 επαναλήψεις). Τέλος, η μέθοδος με τις λιγότερες επαναλήψεις είναι αυτή της **Διχοτόμου με χρήση παραγώγων** (για l = 0.005 έχω 10 επαναλήψεις). Επομένως, πιο αποδοτική (για δεδομένα ε, l και δεδομένων των τριών συναρτήσεων που μας δίνονται) κρίνεται η **μέθοδος της Διχοτόμου με χρήση παραγώγων**, καθώς ανάμεσα στις υπόλοιπες χρειάζεται κατά μέσο όρο τις λιγότερες επαναλήψεις για να προσεγγίσει το ελάχιστο με βάσει μια προκαθορισμένη ακρίβεια.