Τεχνικές Βελτιστοποίησης - Εργασία 1

Ονοματεπώνυμο: Τσαμήτρος Νικόλαος

ΑΕΜ: 10781

Η εργασία αφορά την υλοποίηση μεθόδων μέσω των οποίων μπορούμε να ελαχιστοποιήσουμε μια κυρτή συνάρτηση $f(x)$ όταν $x \in [a,b]$. Συγκεκριμένα οι αλγόριθμοι που υλοποιούνται είναι:

1) Μέθοδοι αναζήτησης ελαχίστου χωρίς την χρήση παραγώγων:

- Μέθοδος της Διχοτόμου,
- Μέθοδος του Χρυσού Τομέα,
- Μέθοδος Fibonacci.

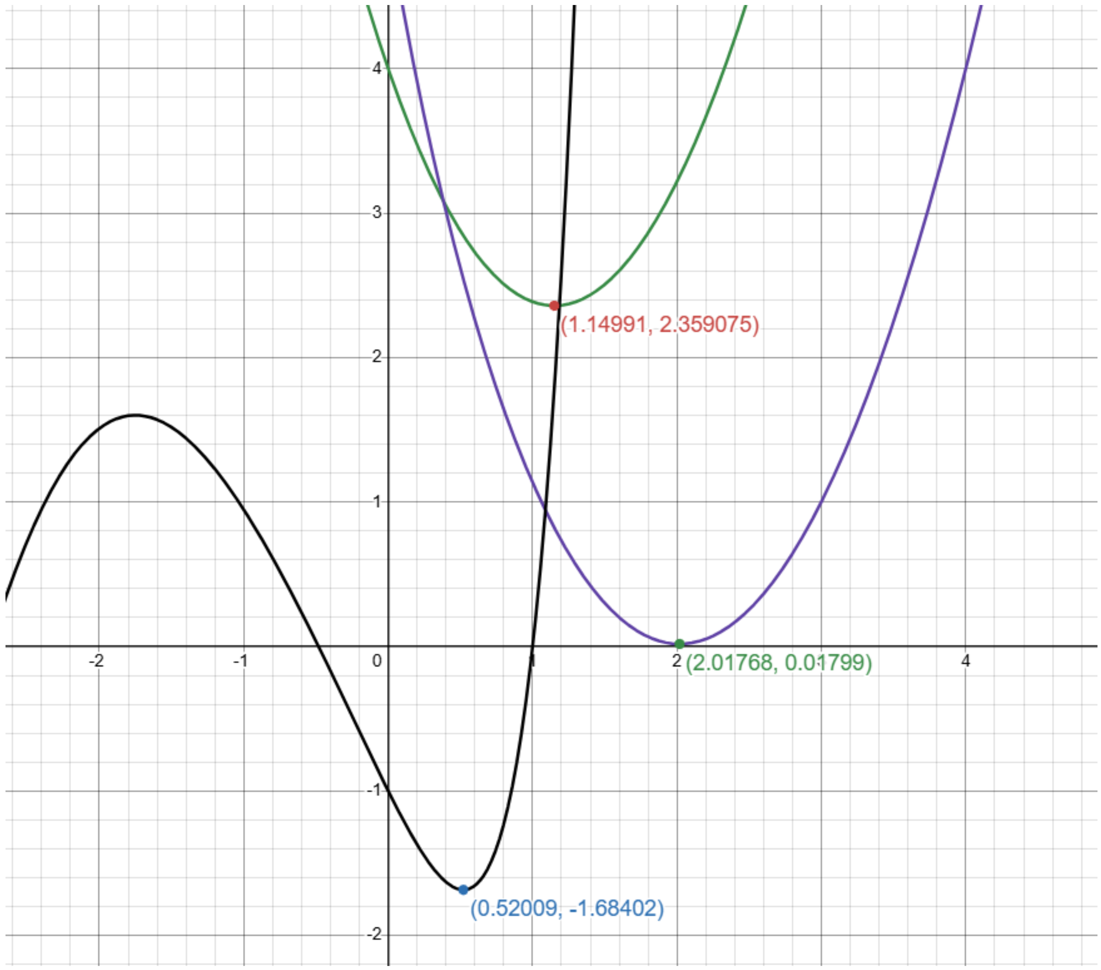
2) Μέθοδοι αναζήτησης με χρήση παραγώγων:

- Μέθοδος της Διχοτόμου με χρήση παραγώγου.

Αυτές οι μέθοδοι πρόκειται να εφαρμοστούν σε 3 συνολικά συναρτήσεις, στο διάστημα $[-1,3]$:

- $f_1(x) = (x-2)^2 + x\ln(x+3)$
- $f_2(x) = e^{-2x} + (x-2)^2$
- $f_3(x) = e^x(x^3-1) + (x-1)\sin x$

Τα διαγράμματα των συναρτήσεων φαίνονται παρακάτω, όπως σχεδιάστηκαν στο desmos.

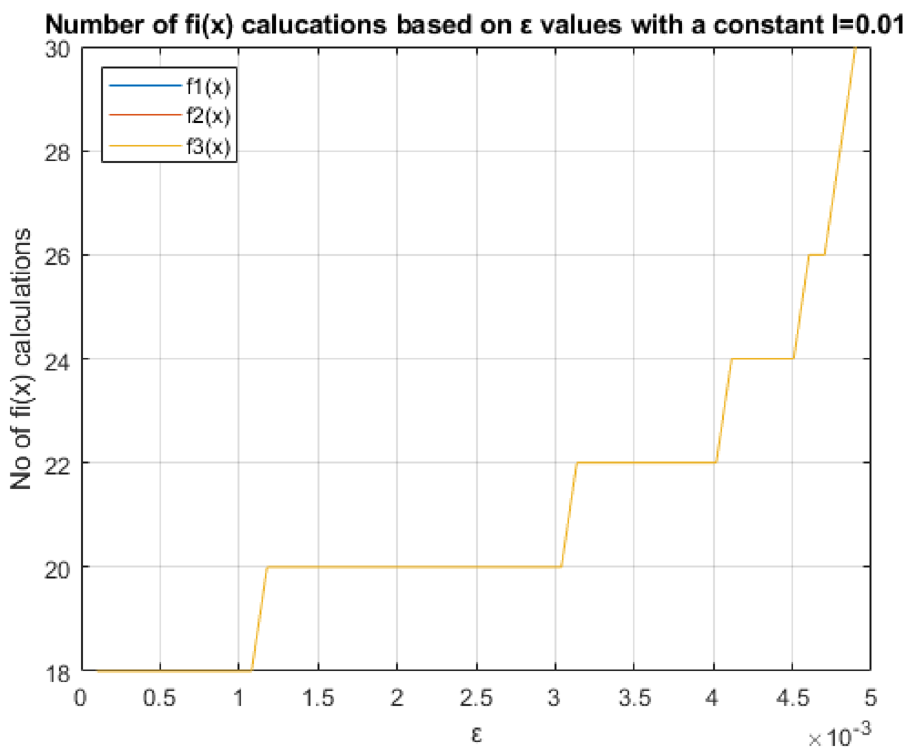


4
3
2
1
0
-1
-2
-2
-1
1
2
4
(1.14991, 2.359075)
(2.01768, 0.01799)
(0.52009, -1.68402)

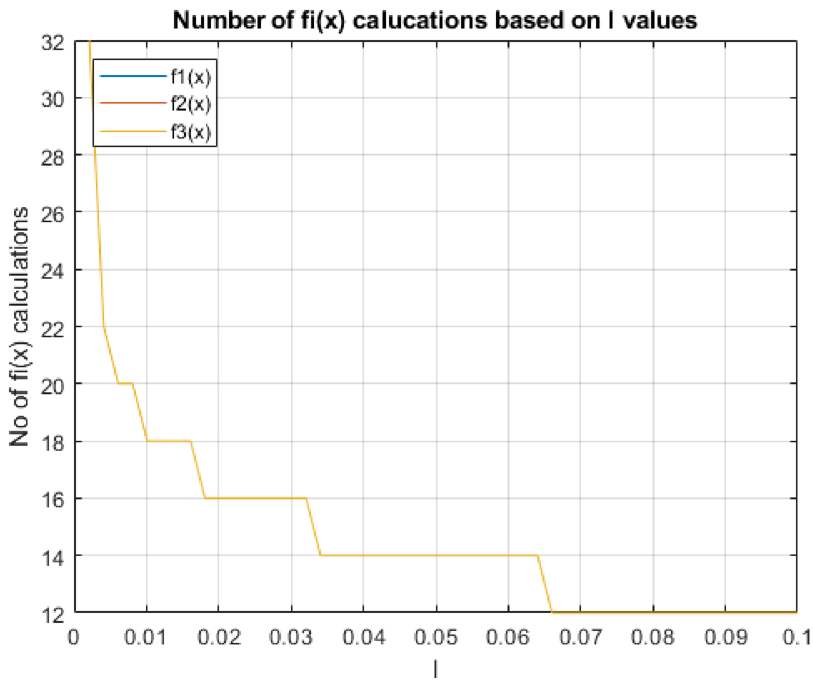
Θέμα 1

Στο πρώτο θέμα ζητείται η υλοποίηση της μεθόδου της διχοτόμου χωρίς την χρήση παραγώγου. Ο αλγόριθμος υπολογίζει το μέσο του διαστήματος $[\alpha, \beta]$ και προσθαφαιρεί σε αυτό την σταθερά ε . Υπολογίζοντας τις τιμές της $f(x)$ στα σημεία που προκύπτουν και συγκρίνοντας τα, ο αλγόριθμος προσδιαρίζει ένα καινούριο διάστημα αναζήτησης $[\alpha', \beta']$. Εφαρμόζοντας τον αλγόριθμο όσο $\beta - \alpha > l$, καταλήγουμε στο τελικό διάστημα όπου ανήκει το ελάχιστο της συνάρτησης. Σε κάθε επανάληψη του αλγορίθμου, υπολογίζουμε την αντικειμενική συνάρτηση 2 φορές, το οποίο δεν χρειάζεται σε άλλους αλγορίθμους. Όμως η μέθοδος είναι γρήγορη και απλή.

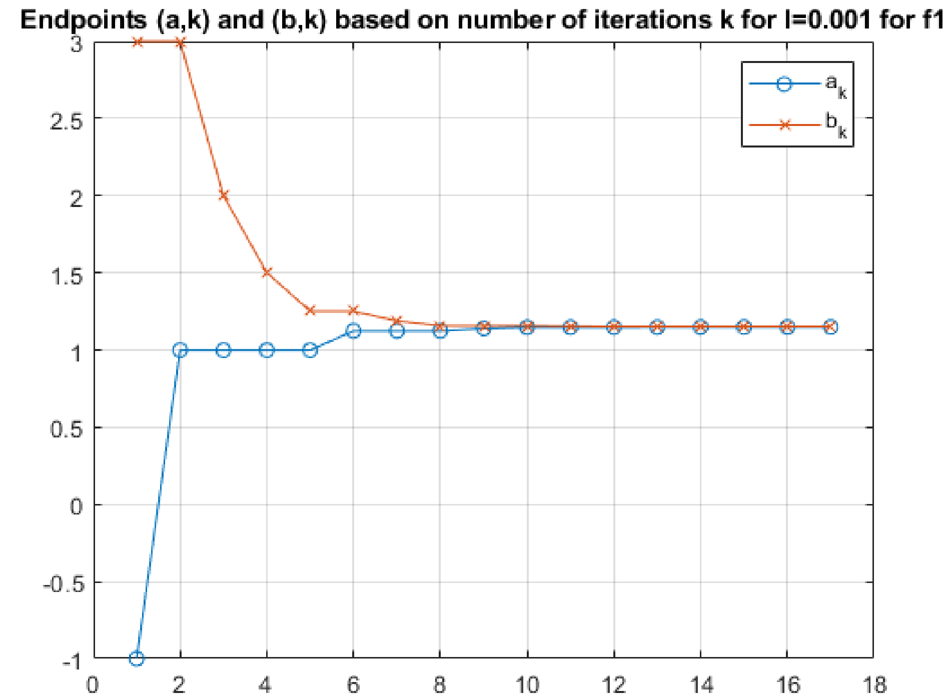
- Στο 1ο υποερώτημα, ζητείται η μεταβολή των υπολογισμών της αντικειμενικής συνάρτησης για σταθερό $l = 0.01$ (τελικό εύρος αναζήτησης) καθώς μεταβάλλουμε την σταθερά ε (απόσταση από την διχοτόμο). Το διάγραμμα που προκύπτει εμφανίζεται παρακάτω. Αφορά και τις 3 δοσμένες συναρτήσεις, τα διαγράμματα των οποίων είναι ίδια. Το διάγραμμα παρουσιάζει άνοδο όσο αυξάνεται το ε , δηλαδή για να τερματίσει ο αλγόριθμος χρειάζεται περισσότερες επαναλήψεις. Αυτό είναι λογικό αφού όσο μικρότερο είναι το ε , τόσο πιο επιθετικά μειώνεται το διάστημα $[\alpha_{\kappa,} \beta_\kappa]$.

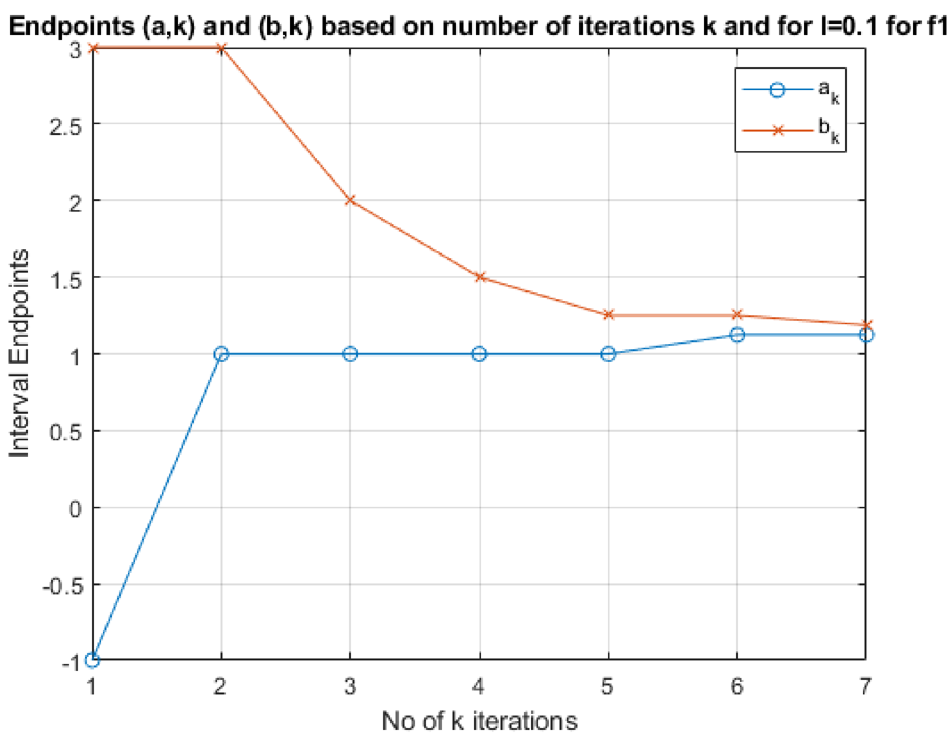


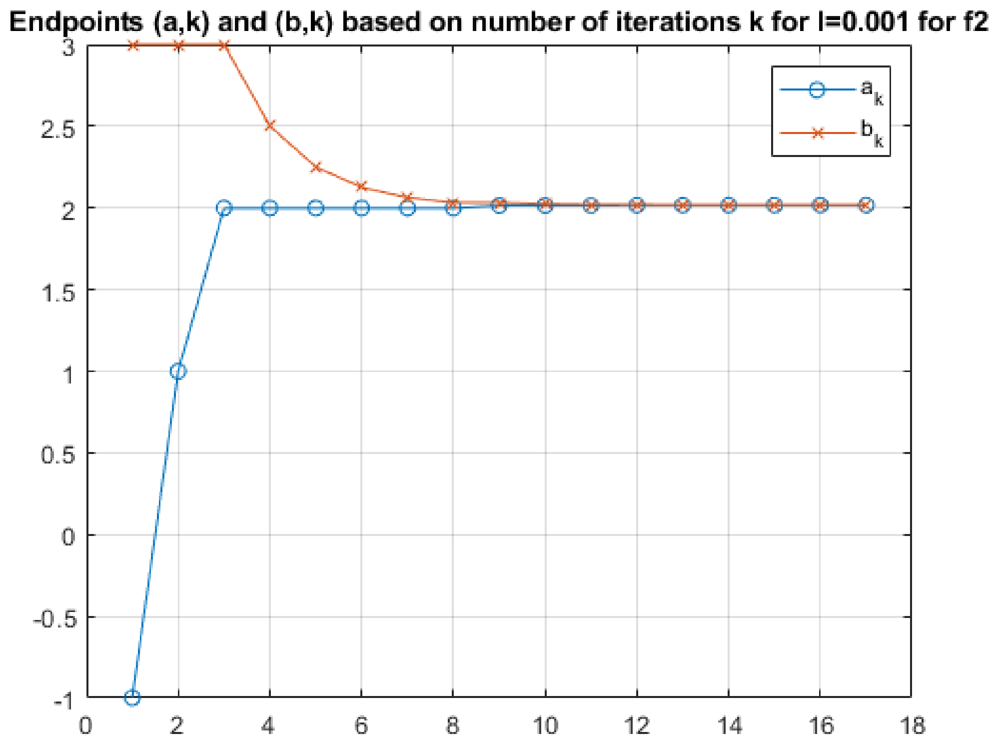
- Στο δεύτερο υποερώτημα ζητείται ο αριθμός των υπολογισμών της αντικειμενικής συνάρτησης με μεταβαλλόμενο l , κρατώντας σταθερό $\varepsilon = 0.001$. Το διάγραμμα που προκύπτει παρουσιάζει κάθοδο, δηλαδή για μεγαλύτερο l ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να τερματίσει. Αυτό είναι λογικό αφού η συνθήκη τερματισμού $\beta - \alpha \leq l$ ικανοποιείται πιο γρήγορα για μεγάλο l .

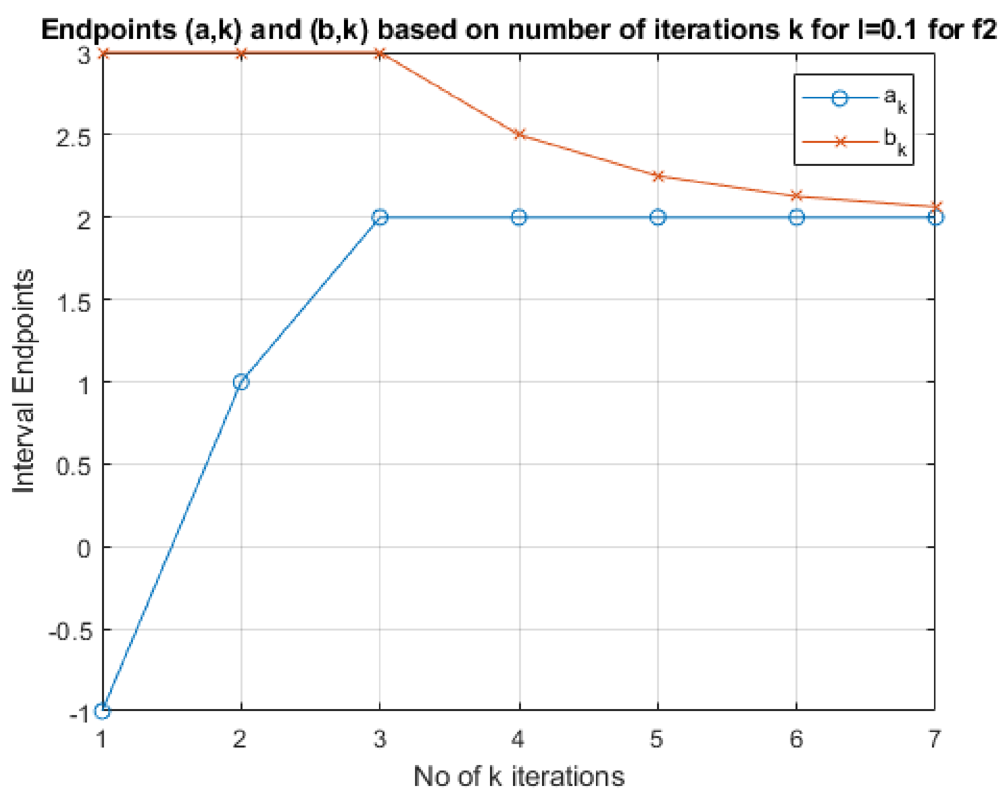


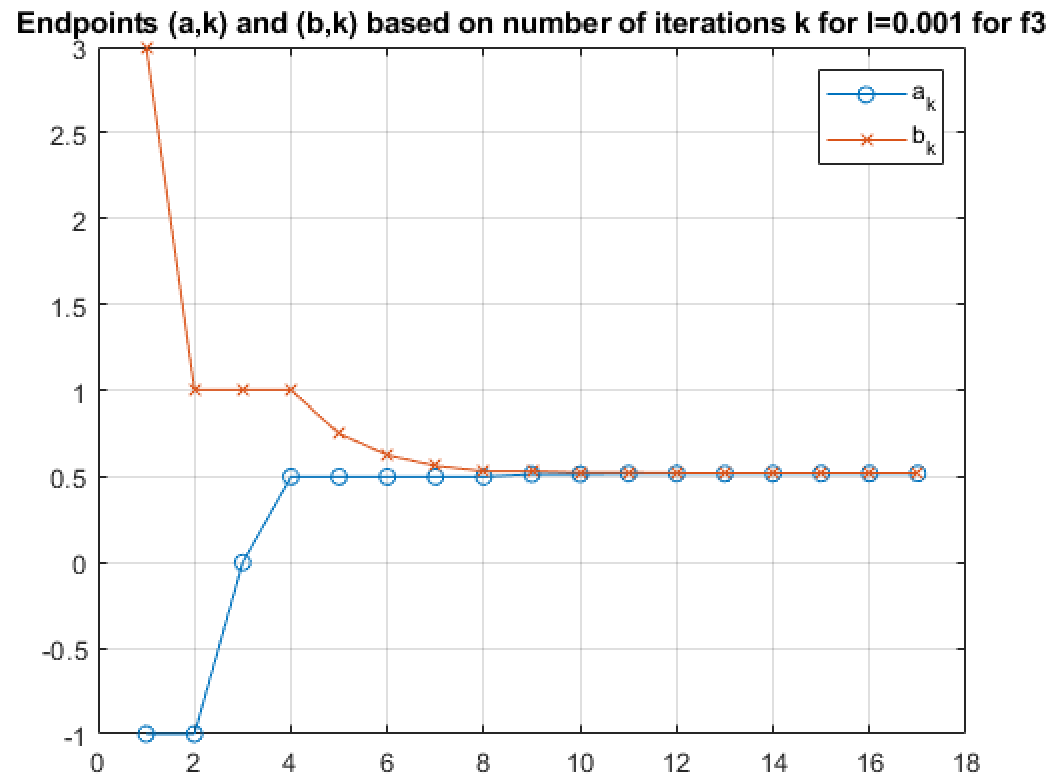
- Στο τρίτο υποερώτημα ζητούνται τα διαγράμματα των άκρων του διαστήματος$[\alpha_\kappa, \beta_\kappa]$ συναρτήσει του δείκτη επαναλήψεων k , δηλαδή (κ, α_κ) και (κ, β_κ), για διάφορες τιμές του τελικού εύρους αναζήτησης l . Τα διαγράμματα που παρουσιάζονται παρακάτω είναι για τις ακραίες τιμές του l ,$l = 0.001$ και $l = 0.1$. Παρατηρούμε πως για μικρό l , άρα για περισσότερες επαναλήψεις, η προσέγγιση του τελικού διαστήματος είναι πιο ακριβής. Στα πρώτα 2 διάγραμματα παρουσιάζεται η$f_1(x)$, στα επόμενα 2 η $f_2(x)$, και στα τελευταία 2 η $f_3(x)$.

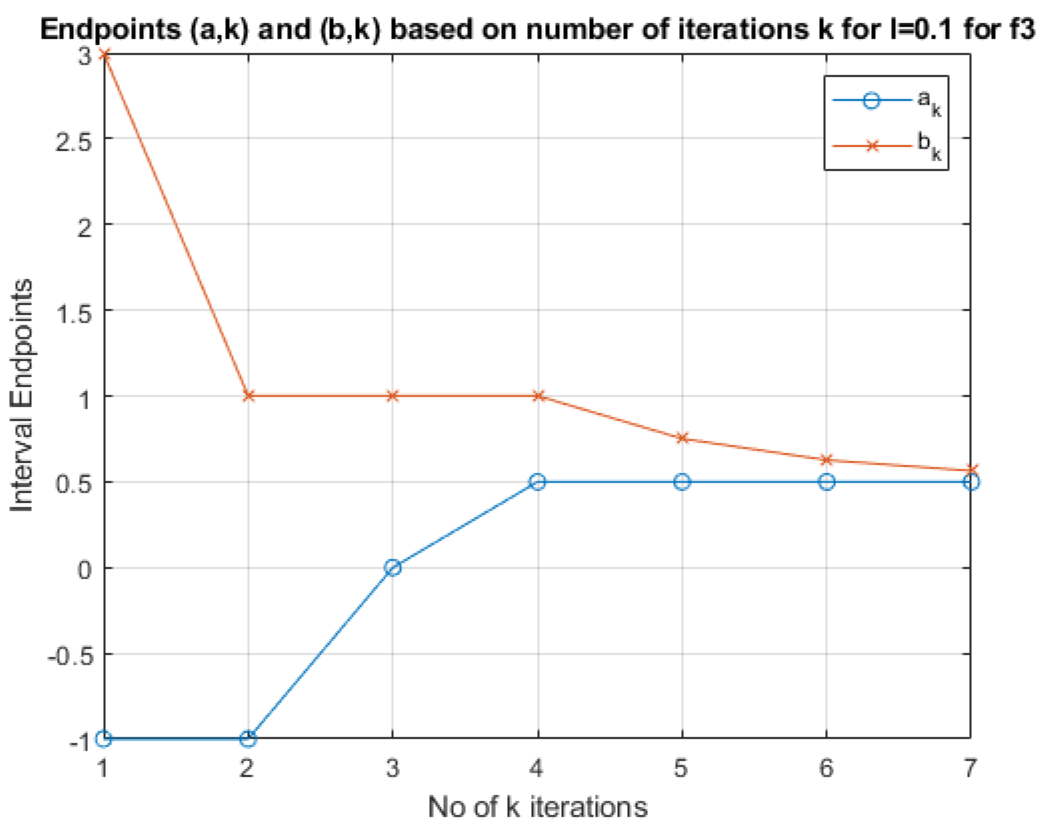

Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f1
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k and for l=0.1 for f1
a_k
b_k
Interval Endpoints
No of k iterations


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f2
a_k
b_k
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f2
a_k
b_k
Interval Endpoints
3
2.5
2
1.5
1
0.5
0
-0.5
-1
1
2
3
4
5
6
7
No of k iterations

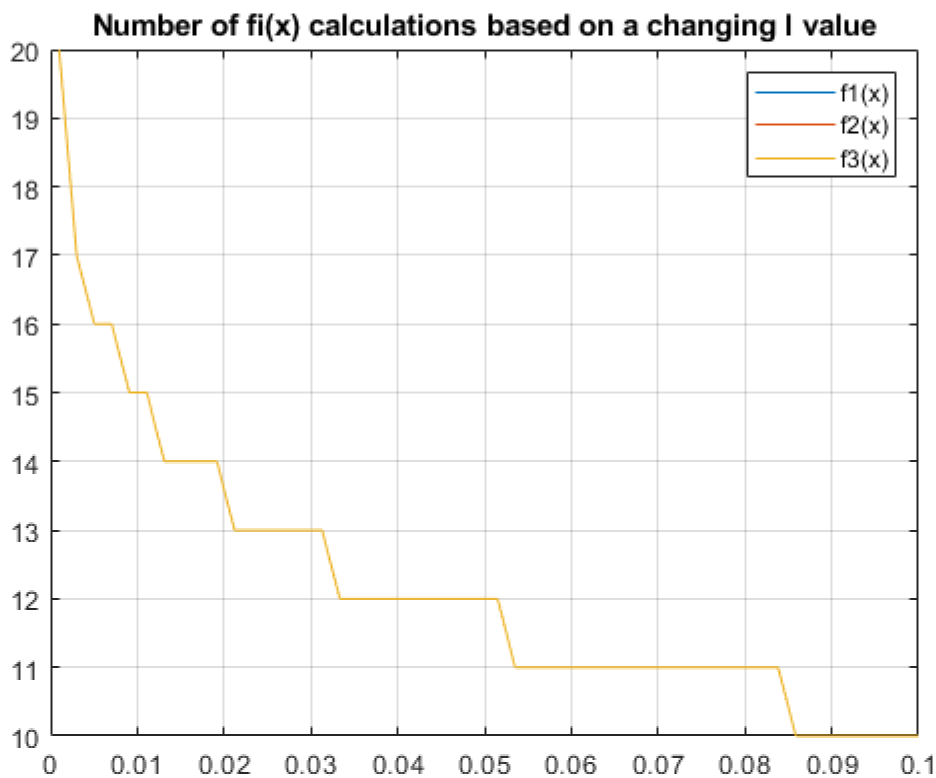

Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f3
a_k
b_k
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
14
16
18


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f3
a_k
b_k
Interval Endpoints
3
2.5
2
1.5
1
0.5
0
-0.5
-1
1
2
3
4
5
6
7
No of k iterations

Θέμα 2

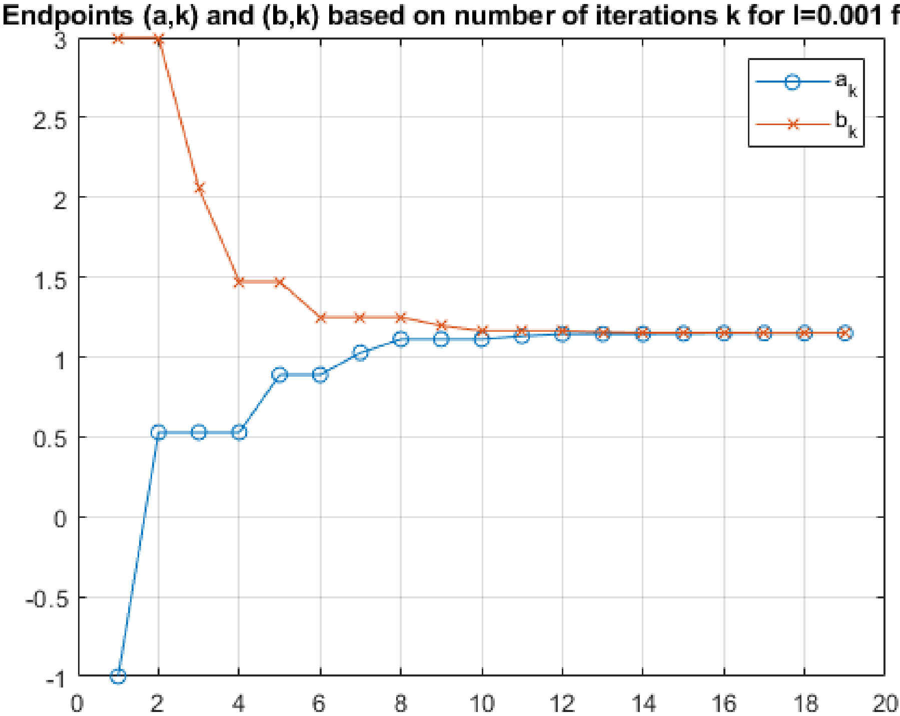
Στο θέμα 2 ζητείται η υλοποίηση της μεθόδου του χρυσού τομέα. Τα αρχικά σημεία αναζήτησης δίνονται από τον τύπο $x_{11} = \alpha + (1-\gamma)(\beta-\alpha)$ και $x_{21} = a + \gamma(\beta - \alpha)$, όπου $\gamma \sim 0.618$. Σε αντίθεση με την μέθοδο της διχοτόμου, τώρα τα x_{1k+1} και x_{2k+1} επιλέγονται ώστε $x_{1k+1} = x_{2k}$ ή $x_{2k+1} = x_{1k}$. Ο αλγόριθμος τερματίζει όταν $\beta - \alpha \leq l$. Σε κάθε επανάληψη του αλγορίθμου χρειάζεται ένας μόνο υπολογισμός της αντικειμενικής συνάρτησης.

- Το πρώτο υποερώτημα είναι το ίδιο με το 1β, απλά με την χρήση της μεθόδου του χρυσού τομέα. Η εξήγηση είναι η ίδια. Η αντικειμενική συνάρτηση υπολογίζεται λιγότερες φορές από ότι στην μέθοδο διχοτόμου.

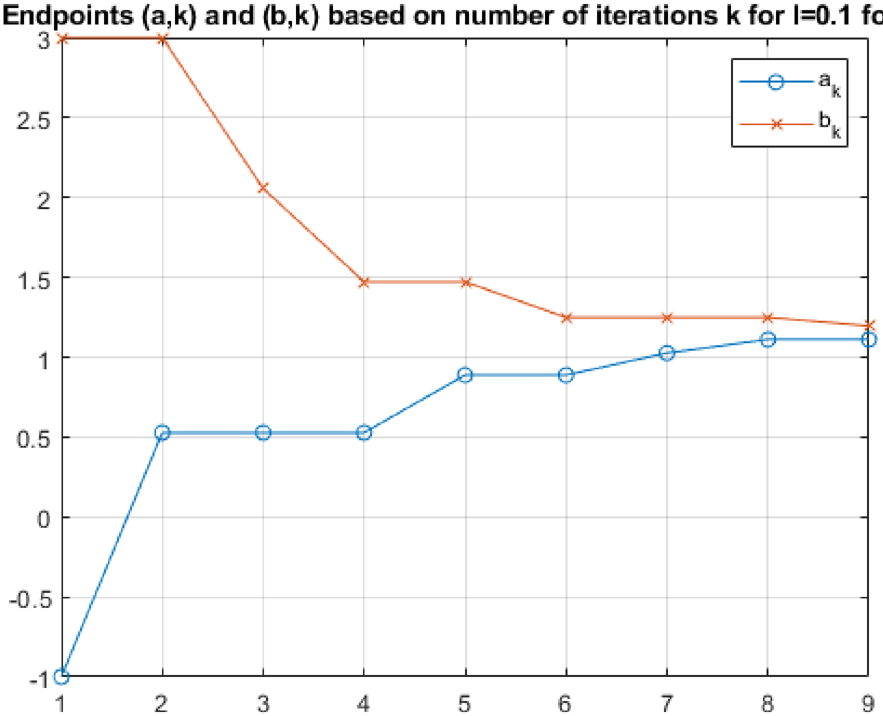


- Το δεύτερο υποερώτημα είναι το ίδιο με το 1γ απλά με την χρήση της μεθόδου του χρυσού τομέα. Τα διαγράμματα παρουσιάζονται με την ίδια σειρά όπως στο 1γ.

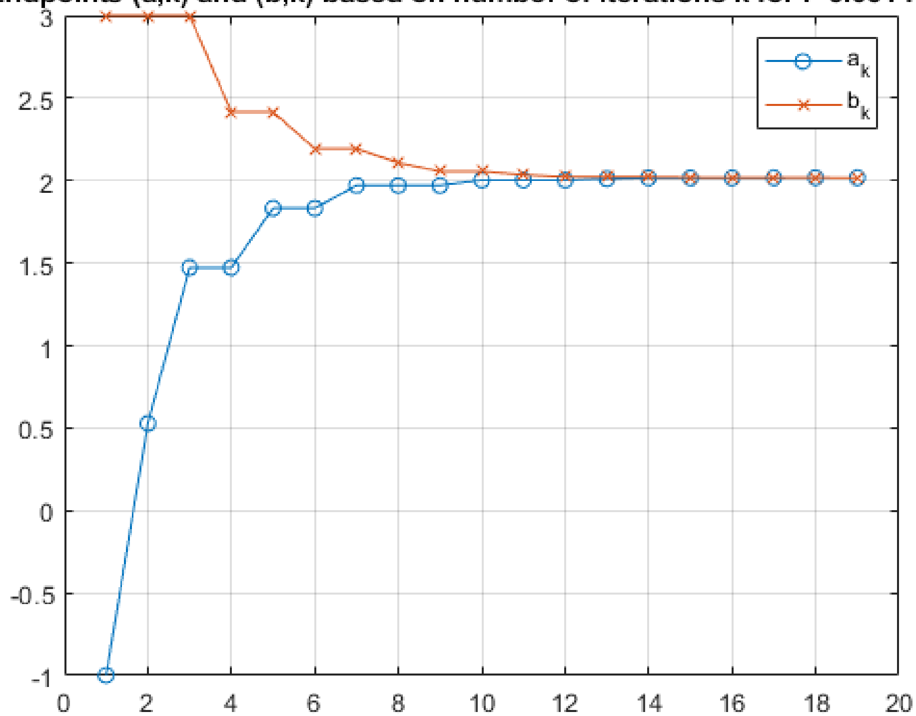
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f1



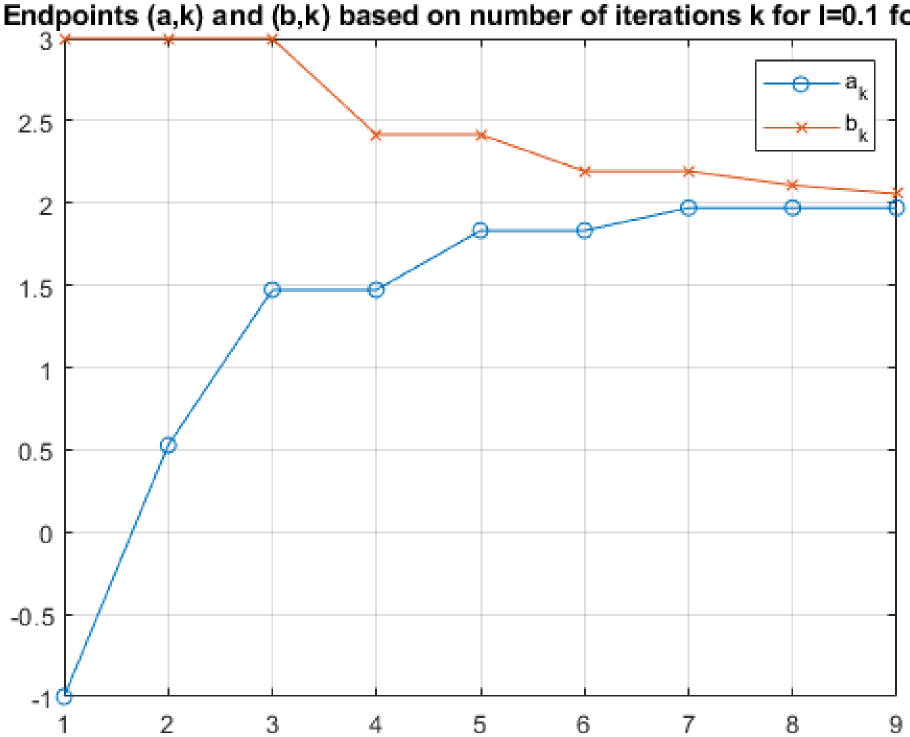
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f1



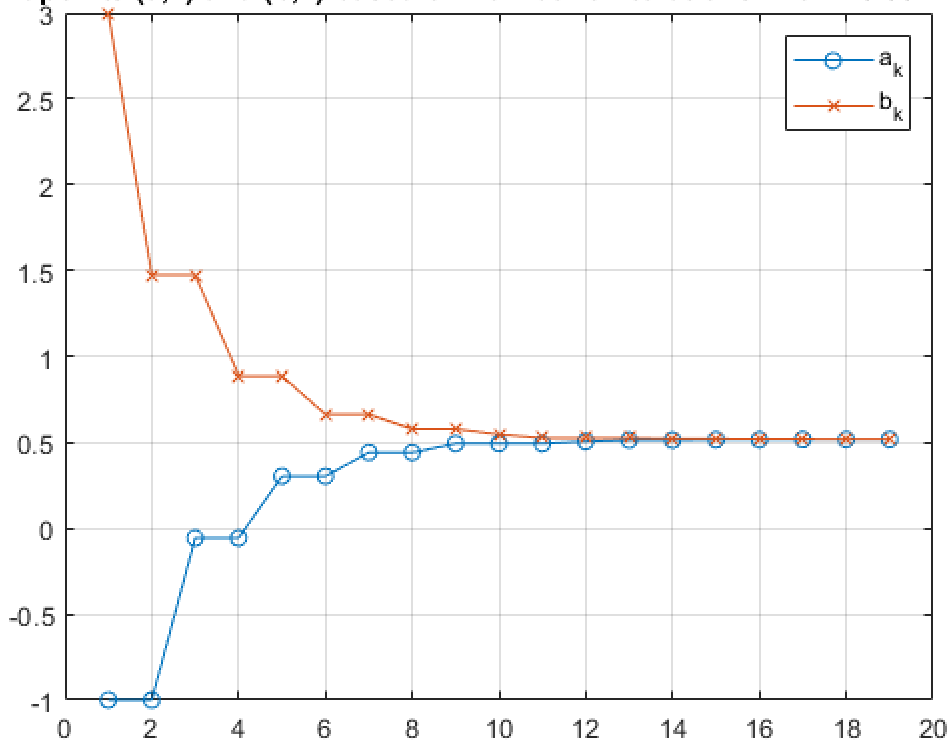
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f2



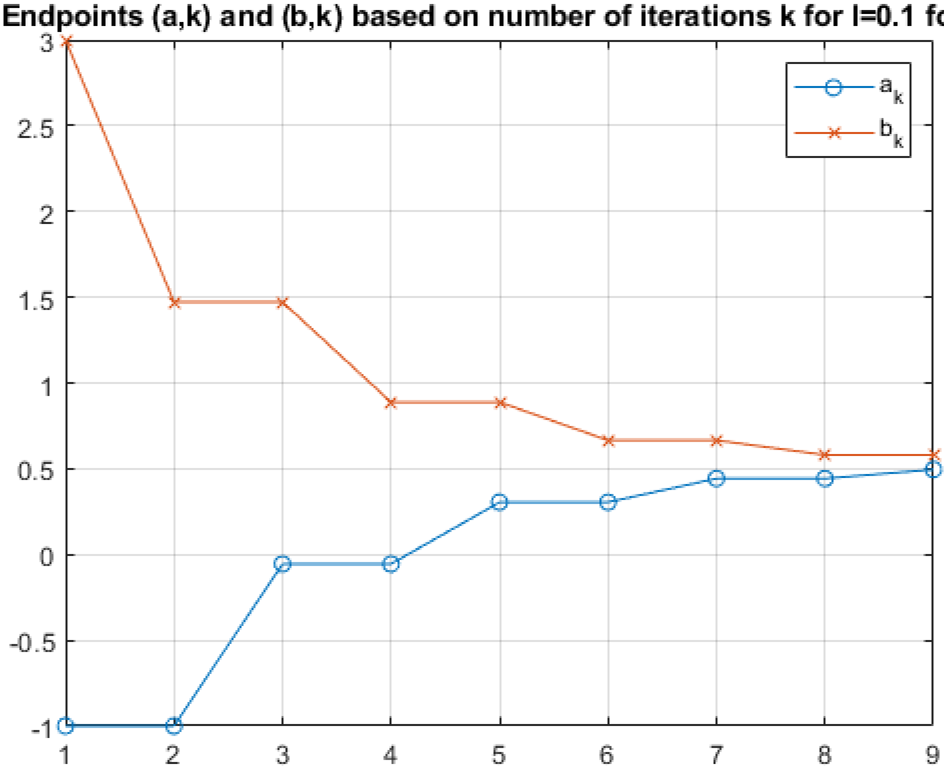
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f2



Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f3



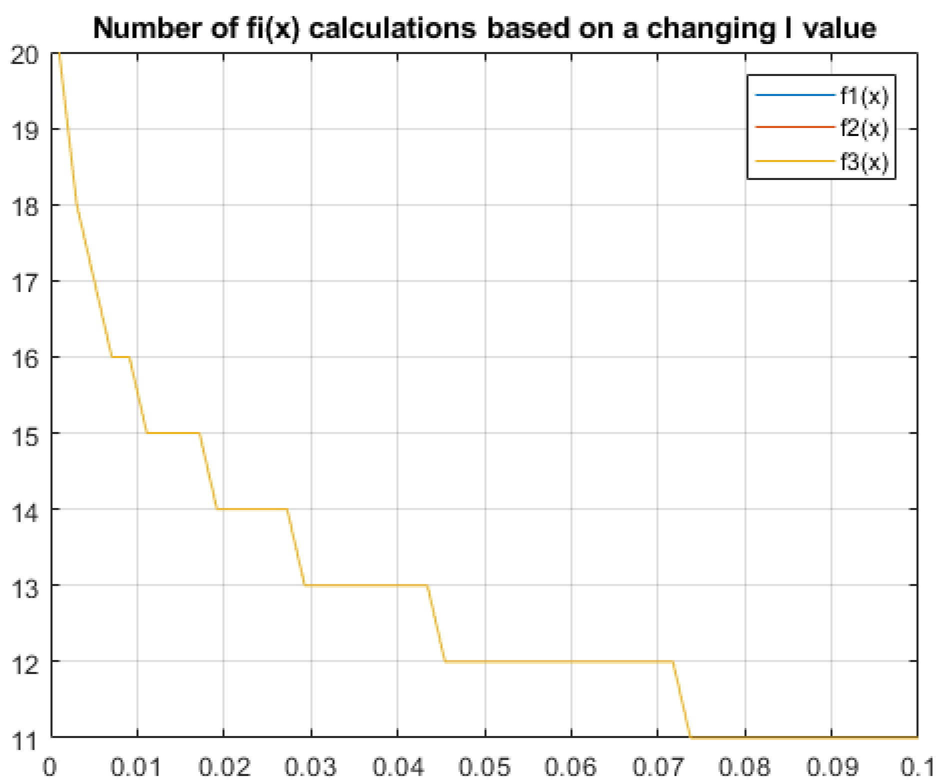
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f3



Θέμα 3

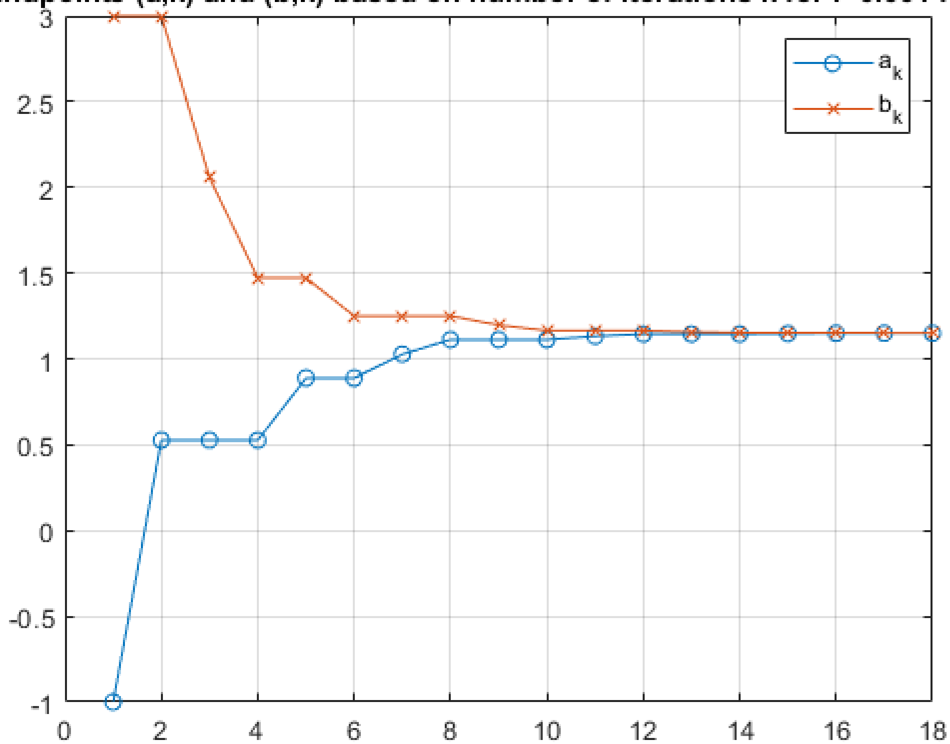
Στο θέμα 3 ζητείται η υλοποίηση της μεθόδου fibonacci. Ο αλγόριθμος υπολογίζει την αντικειμενική συνάρτηση μία φορά σε κάθε επανάληψη. Γενικά για την ελαχιστοποίηση της συνάρτησης, ο αλγόριθμος βασίζεται στην ακολουθία Fibonacci. Σε αντίθεση με τις μεθόδους που αναφέρθηκαν εώς τώρα, η μέθοδος Fibonacci απαιτεί τον εκ των προτέρων προσδιορισμό των συνολικών υπολογισμών n της αντικειμενικής συνάρτησης. Για αυτόν πρέπει να ικανοποιείται η $F_n > \frac{(\beta_1 - \alpha_1)}{l}$, όπου F_n το n στοιχείο της ακολουθίας Fibonacci.

- Το πρώτο υποερώτημα είναι το ίδιο με το 1β και 2α απλά με την μέθοδο fibonacci. Η εξήγηση είναι η ίδια. Βλέπουμε πως η fibonacci είναι σχεδόν ταυτόσημη με την μέθοδο του χρυσού τομέα.

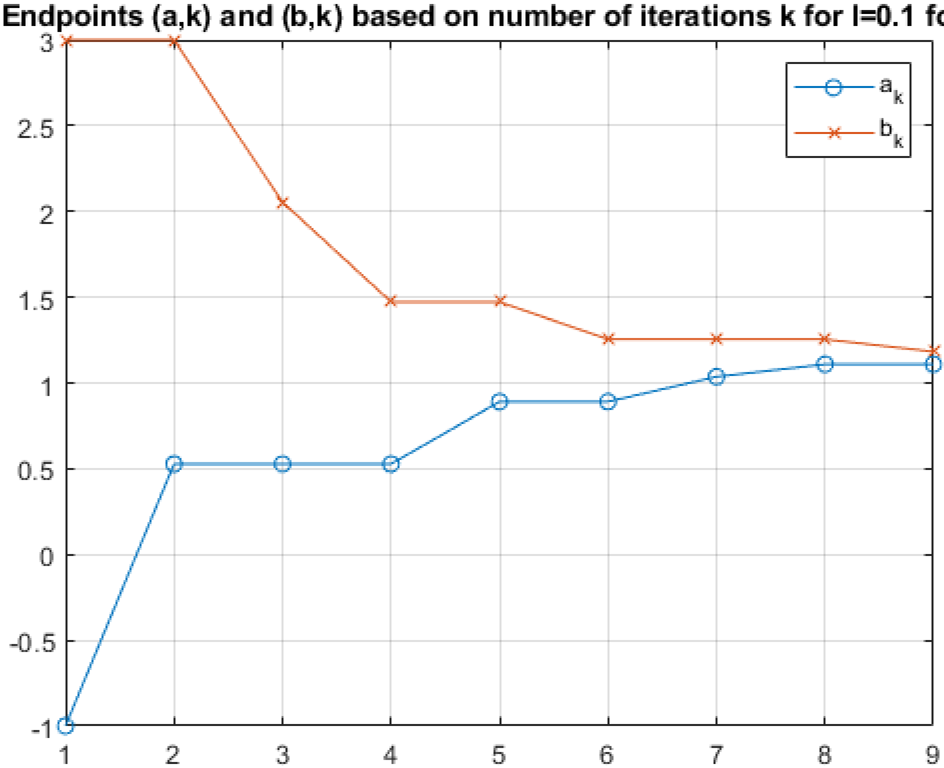


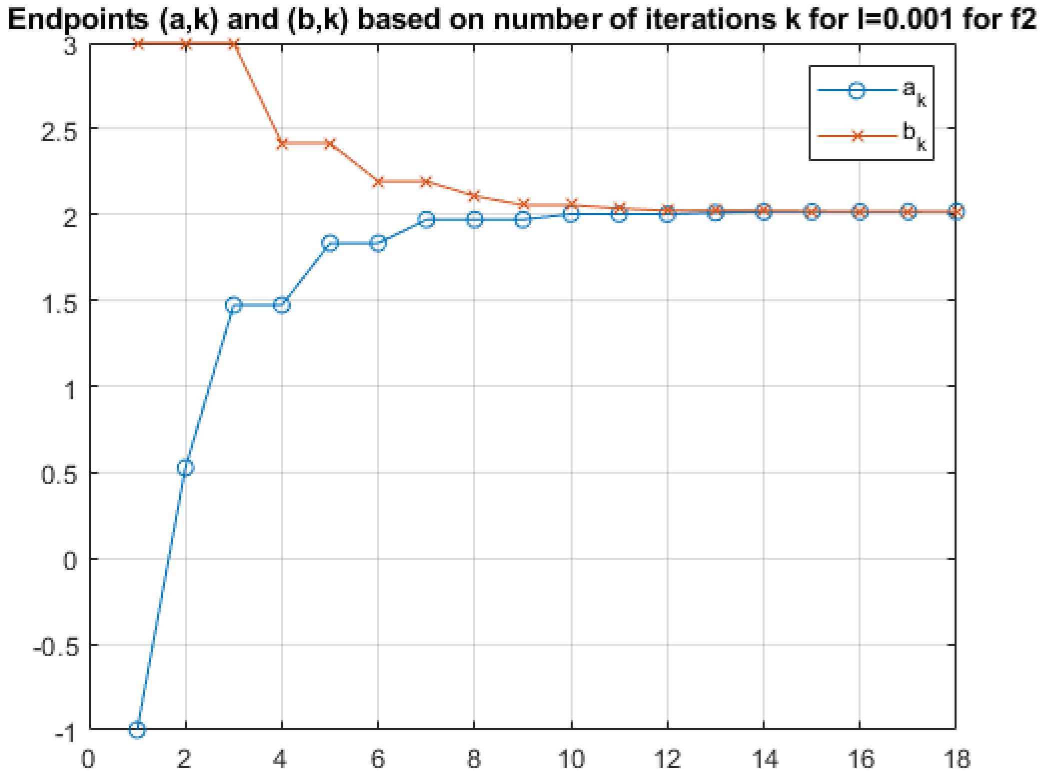
- Το δεύτερο υποερώτημα είναι το ίδιο με το 1γ, 2β απλά με την χρήση της μεθόδου fibonacci. Τα διαγράμματα παρουσιάζονται με την ίδια σειρά όπως στο 1γ, 2β.

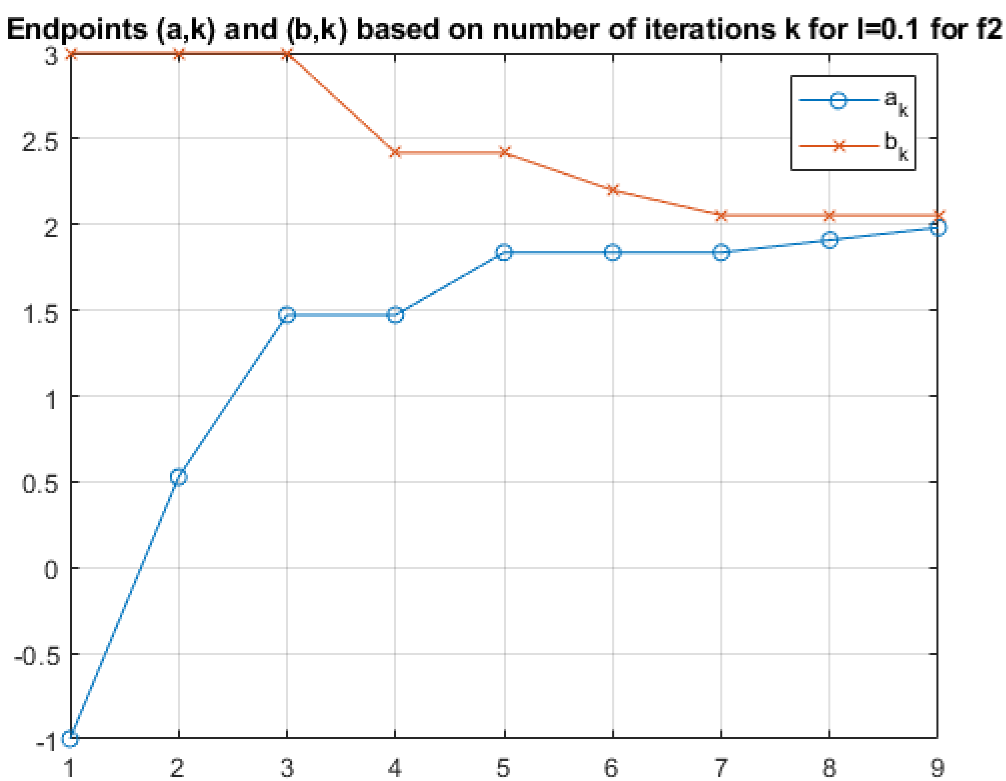
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f1

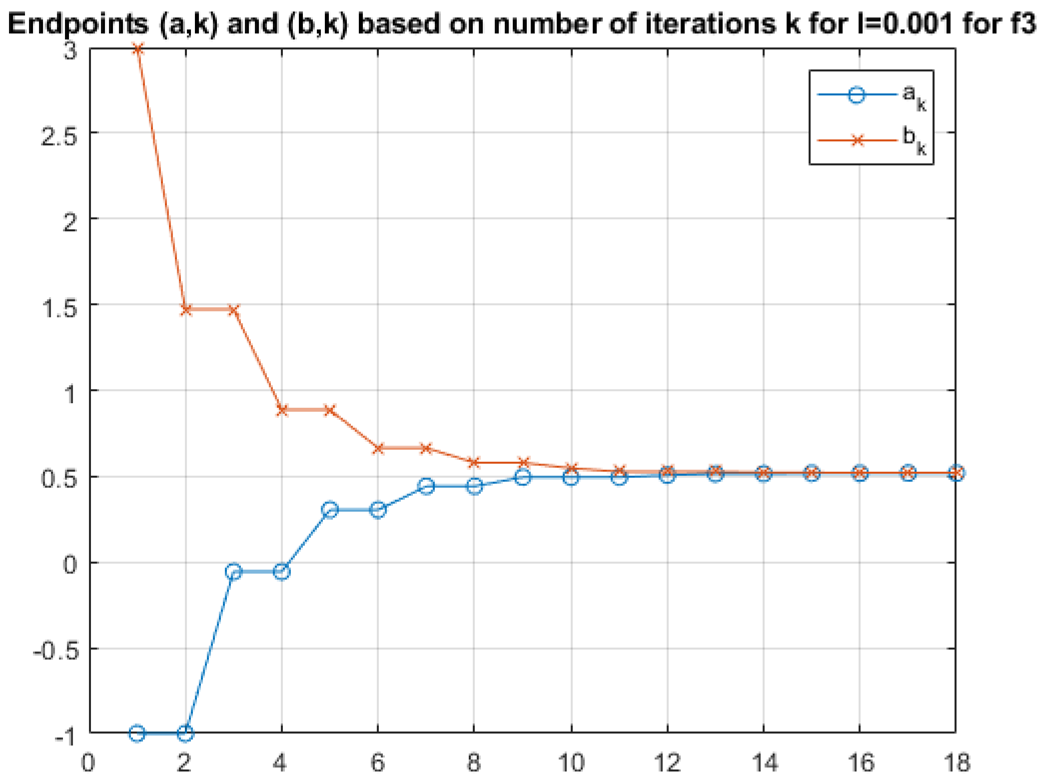


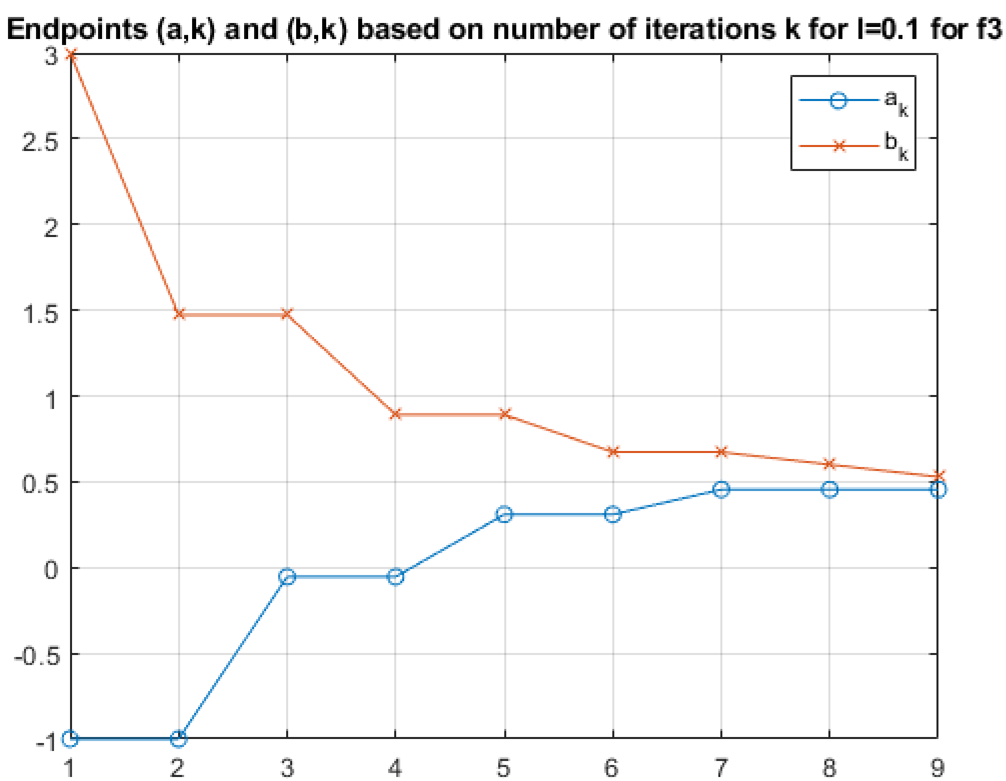
Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f1




Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f2
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f2
a_k
b_k

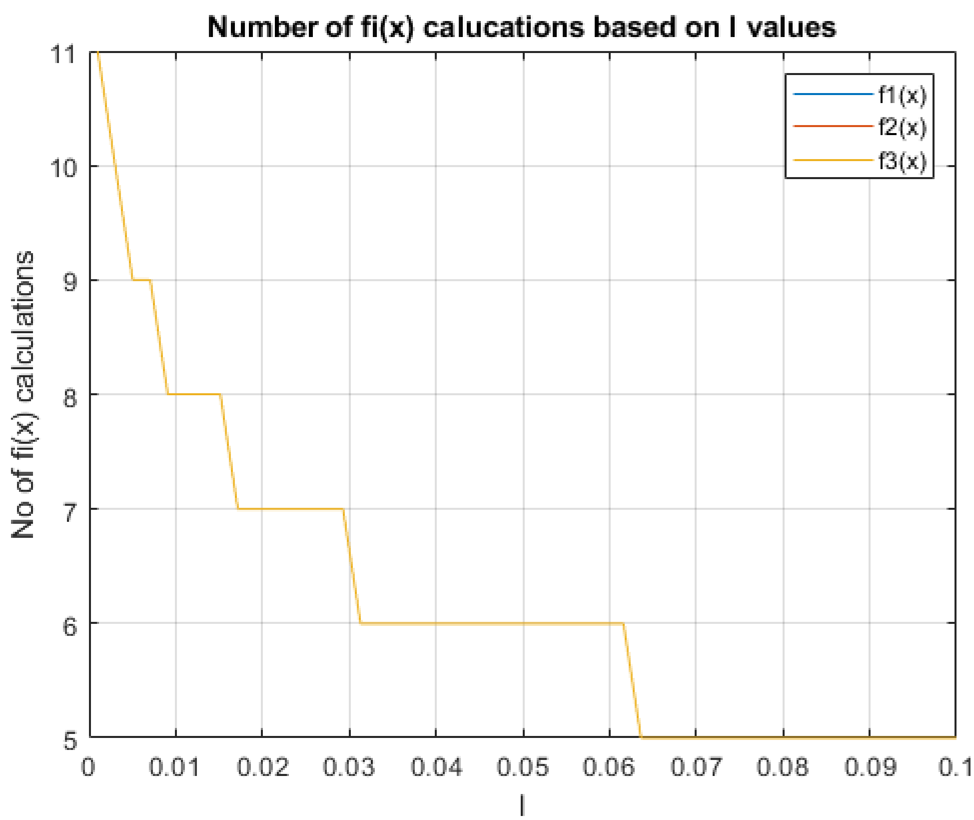

Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f3
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f3
a_k
b_k

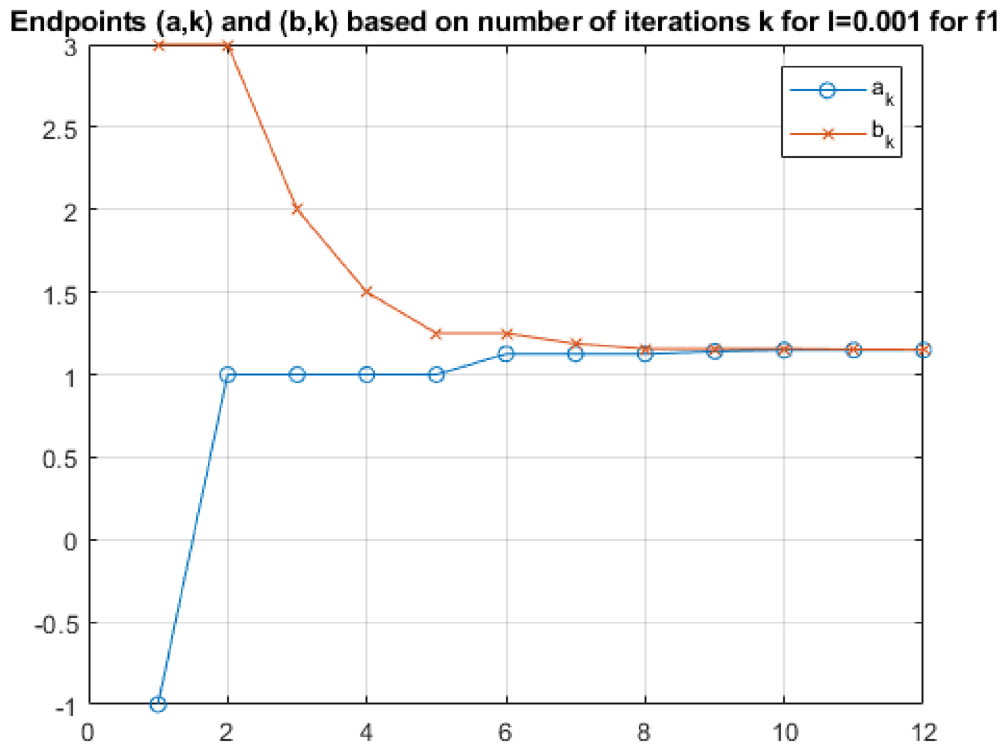
Θέμα 4

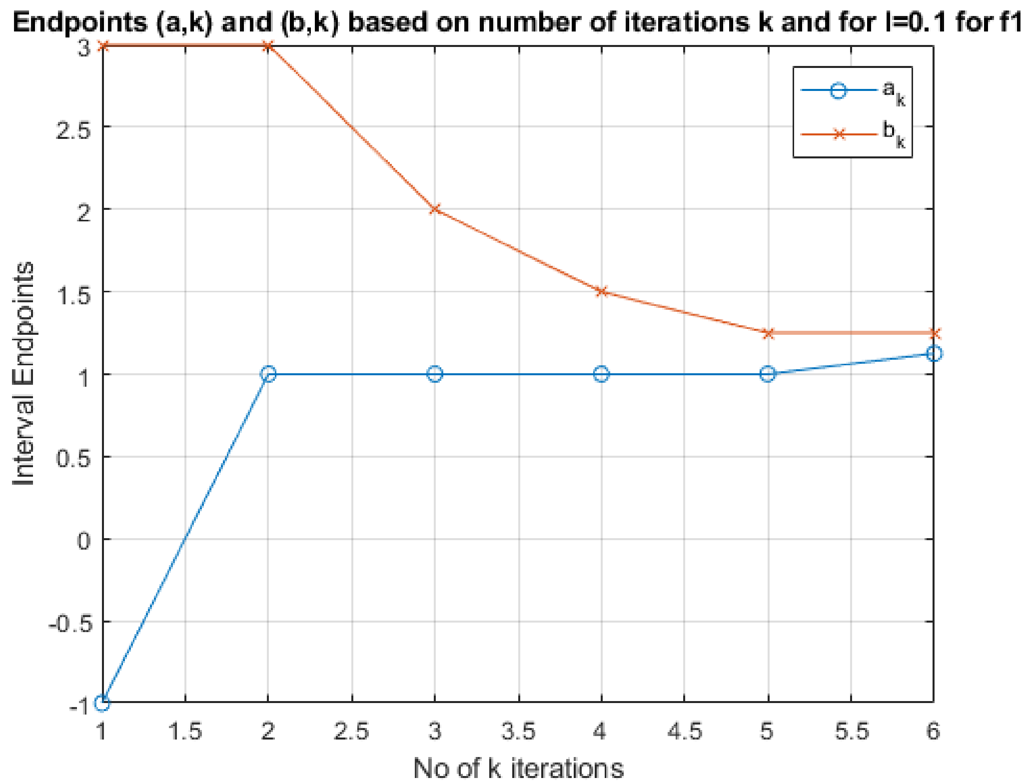
Στο θέμα 4 ζητείται η υλοποίηση της μεθόδου της διχοτόμου με την χρήση παραγώγων. Σε αυτήν επιλέγουμε τον συνολικό αριθμό υπολογισμών n της αντικειμενικής συνάρτησης $f(x)$ ώστε να ικανοποιεί τον περιορισμό $2^{-n} \leq \frac{l}{b_1 - a_1}$. Ο αλγόριθμος υπολογίζει την παράγωγο στο μέσο του εκάστοτε διάστηματος και ανάλογα με το πρόσημο της, συρρικνώνει το διάστημα αναζήτησης. Σε κάθε επανάληψη η αντικειμενική συνάρτηση υπολογίζεται μόνο μια φορά, όμως με την χρήση του Symbolic Math Toolbox για τον υπολογισμό της παραγώγου στο Matlab η μέθοδος είναι αρκετά πιο αργή από τις υπόλοιπες. Προφανώς, προϋπόθεση είναι πως η συνάρτηση είναι διαφορίσιμη στο δοσμένο διάστημα.

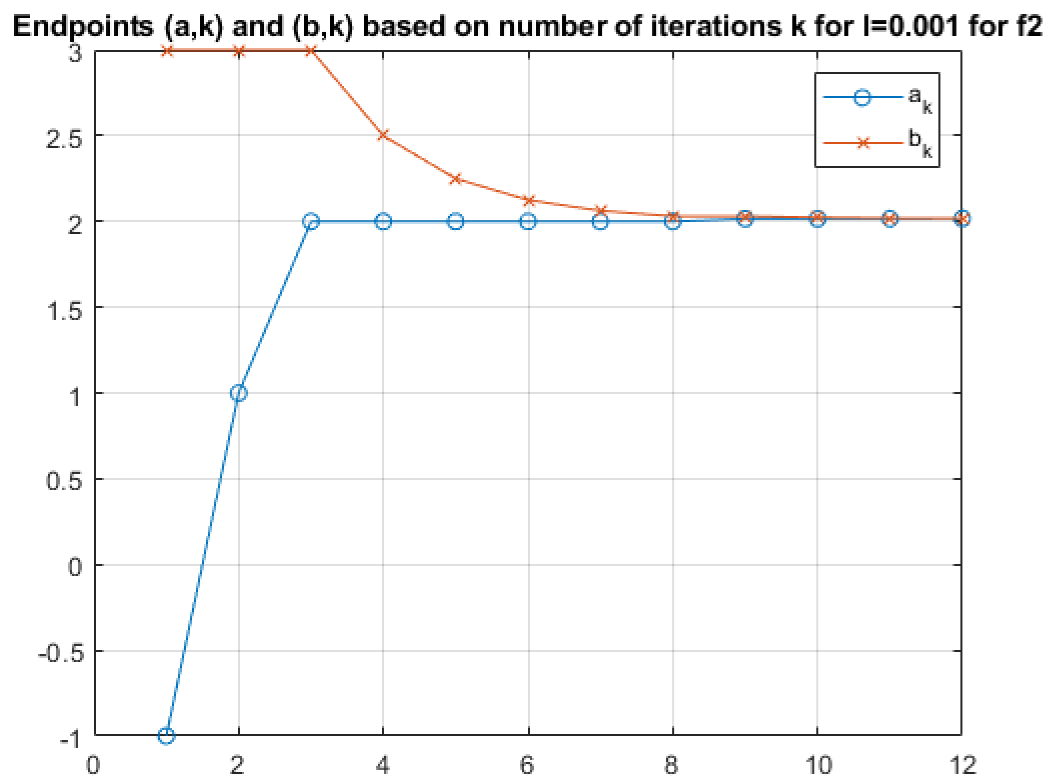
- Το πρώτο υποερώτημα είναι το ίδιο με το 1β, απλά με την χρήση της διχοτόμου με την χρήση παραγώγων. Η εξήγηση είναι η ίδια. Βλέπουμε όμως πως η συγκεκριμένη μέθοδος απαιτεί τον ελάχιστο αριθμό υπολογισμών της αντικειμενικής συνάρτησης.

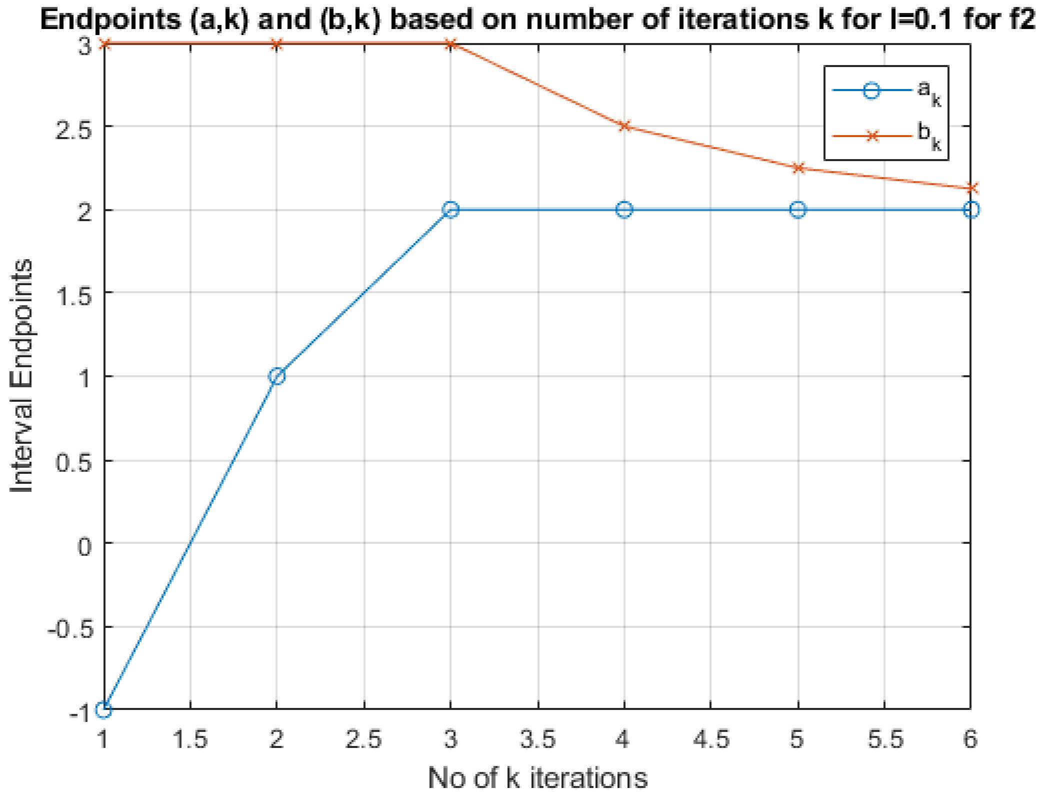


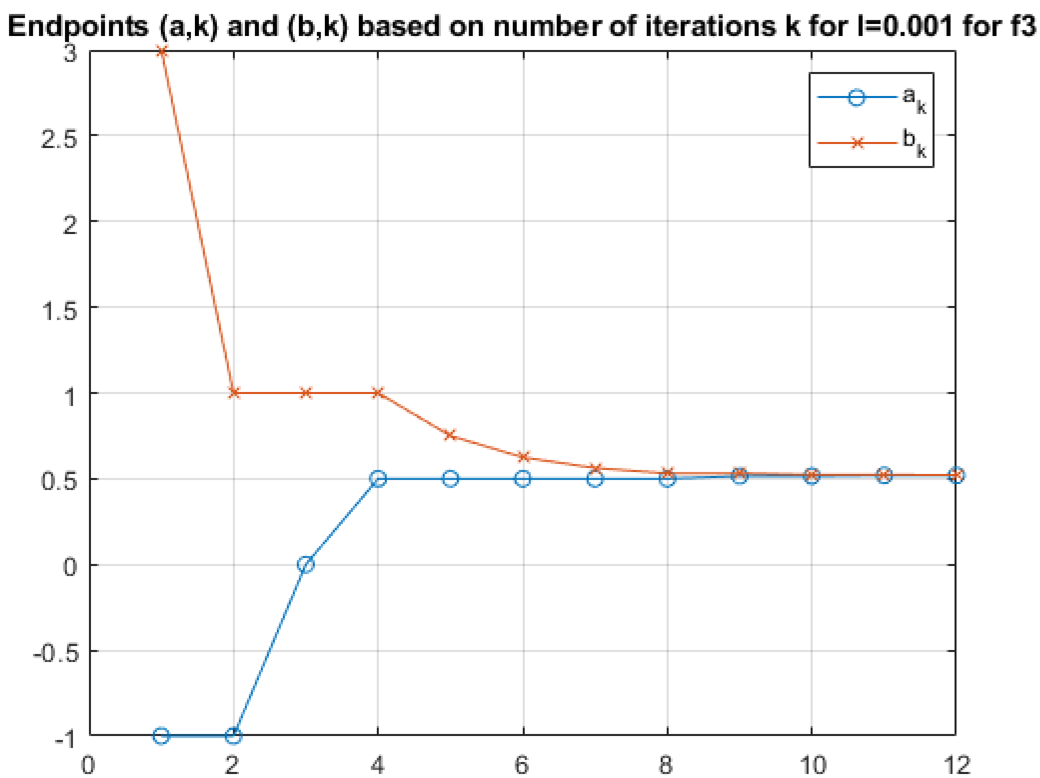
- Το δεύτερο υποερώτημα είναι το ίδιο με το 1γ, 2β, 3β απλά με την χρήση της μεθόδου της διχοτόμου με την χρήση παραγώγων. Τα διαγράμματα παρουσιάζονται με την ίδια σειρά όπως στο 1γ, 2β, 3β.

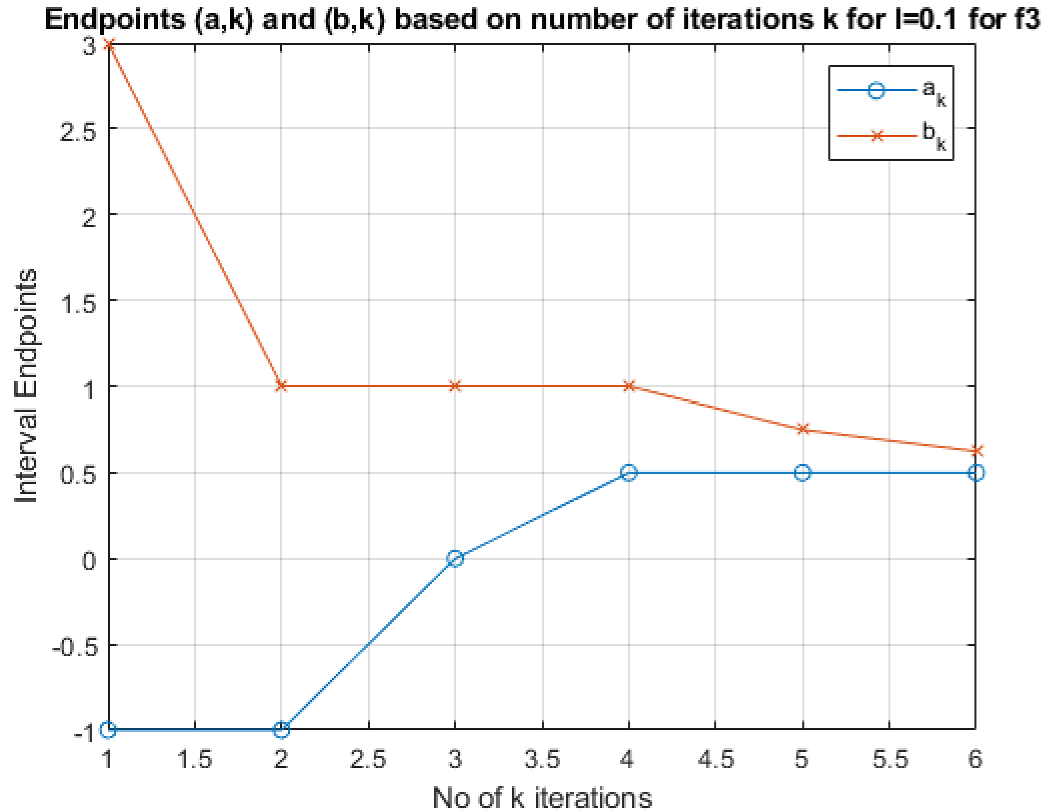



Endpoints (a,k) and (b,k) based on number of iterations k and for l=0.1 for f1
Interval Endpoints
No of k iterations
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f2
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.1 for f2
Interval Endpoints
No of k iterations
a_k
b_k


Endpoints (a,k) and (b,k) based on number of iterations k for l=0.001 for f3
a_k
b_k



Γενικά, συγκρίνοντας της μεθόδους χωρίς την χρήση παραγώγων μεταξύ τους και εκτιμώντας τον μικρότερο θετικό ακέραιο που ικανοποιεί τις παρακάτω σχέσεις, καθώς και συγκρίνοντας τα αποτελέσματα των διαγραμμάτων μεταξύ τους συμπεραίνουμε τα εξής:

- Μέθοδος της διχοτόμου: $2^{-\frac{n}{2}} \leq \frac{l}{\beta_1-\alpha_1}$
- Μέθοδος του χρυσού τομέα: $0.618^{n-1} \leq \frac{l}{\beta_1-\alpha_1}$
- Μέθοδος Fibonacci: $F_n \geq \frac{(\beta_1-\alpha_1)}{l}$

Για σταθερό πηλίκο, όσο μικρότερο είναι το n , τόσο πιο αποτελεσματικός είναι ο αλγόριθμος. Συνεπώς, πιο αποτελεσματική είναι η Fibonacci, μετά η μέθοδος του χρυσόυ τομέα και μετά η μέθοδος της διχοτόμου. Η μέθοδος της διχοτόμου με την χρήση παραγώγων, παρόλο που απαιτεί τον λιγότερο αριθμό επαναλήψεων, είναι πιο αργή(ίσως να φταίει και η υλοποίηση), καθώς απαιτεί τον υπολογισμό της παραγώγου της συνάρτησης.