$1^{η}$ Εργασία Τεχνικές Βελτιστοποίησης

Ηλιάνα Κόγια

ΑΕΜ: 10090

2022-2023

Ζητούμενο της εργασίας είναι η ελαχιστοποίηση μιας δοσμένης κυρτής συνάρτησης f(x) όταν $x \in [a, b]$

[a, b] = [-1, 3]

$f_1 = (x-2)^2 + x\ln(x+3)$

$f_2 = 5^x + (2-cosx)^2$

$f_3 = e^x\,(x^3-1) + (x-1)\sin x$

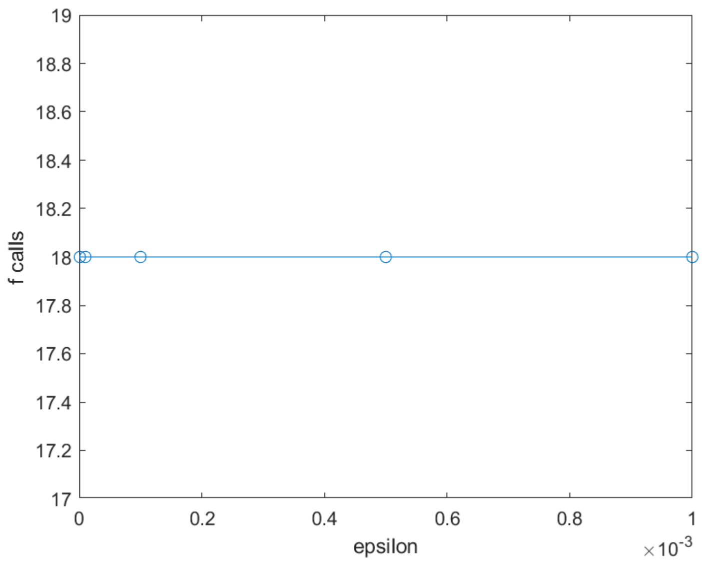
1. ***<u>Μέθοδος Διχοτόμου</u>***

<u>Σταθερό τελικό εύρος αναζήτησης l = 0.01 και μεταβαλλόμενο ε>0</u>.

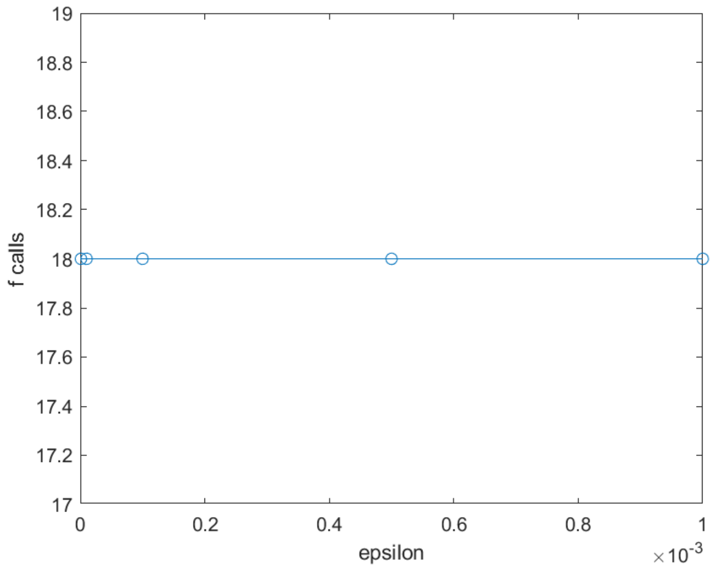
Επιλέγουμε: $\varepsilon = \{10^{-3}, 5*10^{-4}, 10^{-4}, 10^{-5}, 10^{-6}\}$

Μελετούμε τους υπολογισμούς της αντικειμενικής συνάρτησης f_i καθώς το ε μεταβάλλεται. Για κάθε ε βρίσκουμε πόσες φορές κλήθηκε η συνάρτηση στον αλγόριθμο. Το διάγραμμα που προκύπτει για κάθε συνάρτηση είναι:

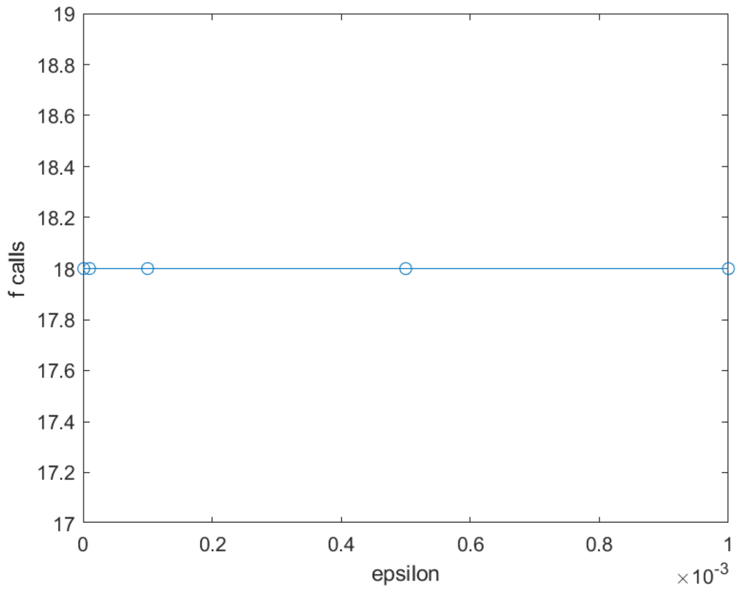
f_1



f_2



f_3



Παρατηρούμε πως το ε δεν επηρεάζει τους υπολογισμούς της αντικειμενικής συνάρτησης f. Επίσης, για μεταβαλλόμενο ε ο αριθμός κλήσεων είναι ίδιος για κάθε συνάρτηση.

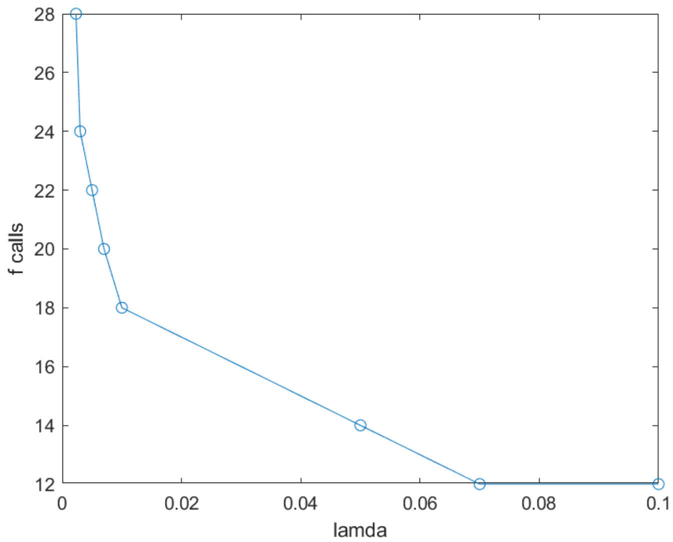
<u>*Σταθερή απόσταση από διχοτόμο ε = 0.001 και μεταβαλλόμενο l*</u>

Επιλέγουμε: l = {0.1, 0.07, 0.05, 0.01, 0.007, 0.005, 0.003, 0.0023}

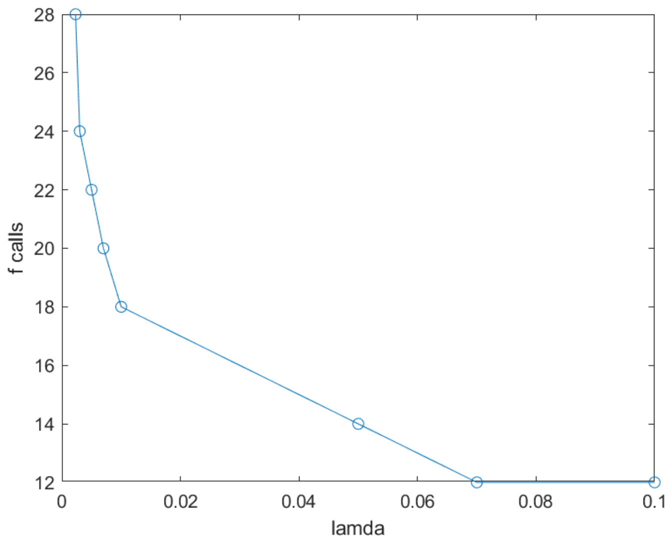
<u>*Να σημειωθεί ότι στη μέθοδο της Διχοτόμου δεν πρέπει η ακρίβεια l να είναι μικρότερη από 2ε = 0.002.*</u> *Γι' αυτό επιλέγουμε l≥0.0023>0.002*

Μελετούμε τους υπολογισμούς της αντικειμενικής συνάρτησης f_i καθώς το l μεταβάλλεται. Για κάθε l βρίσκουμε πόσες φορές κλήθηκε η συνάρτηση στον αλγόριθμο. Το διάγραμμα που προκύπτει για κάθε συνάρτηση είναι:

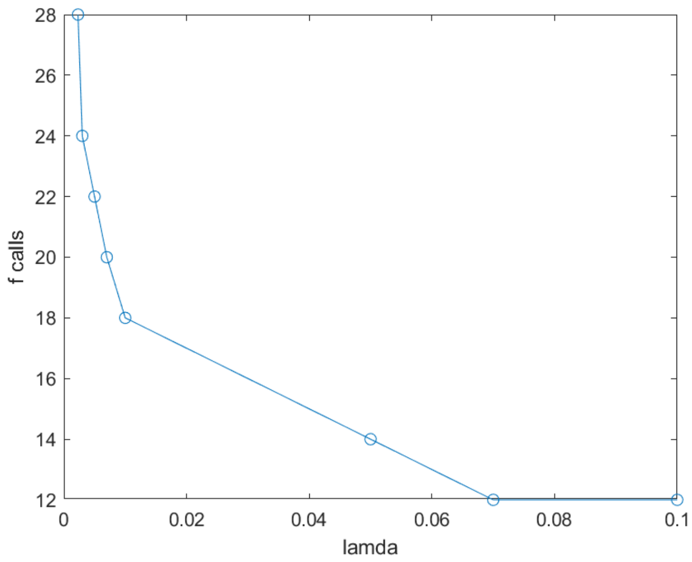
f_1



f_2



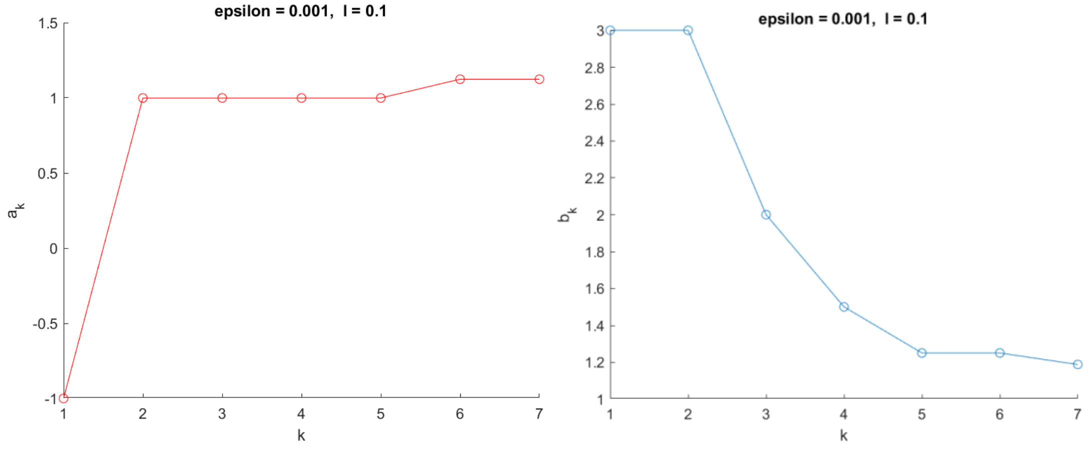
f_3

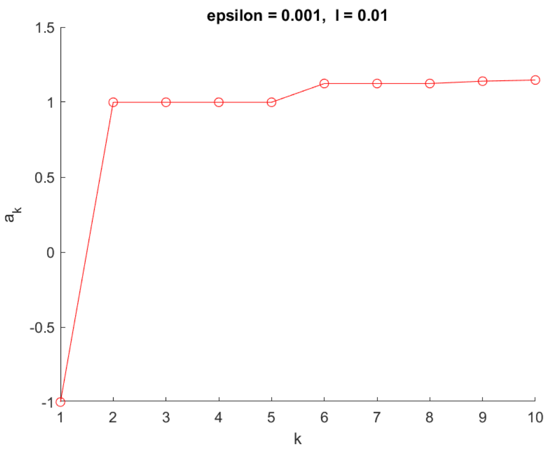


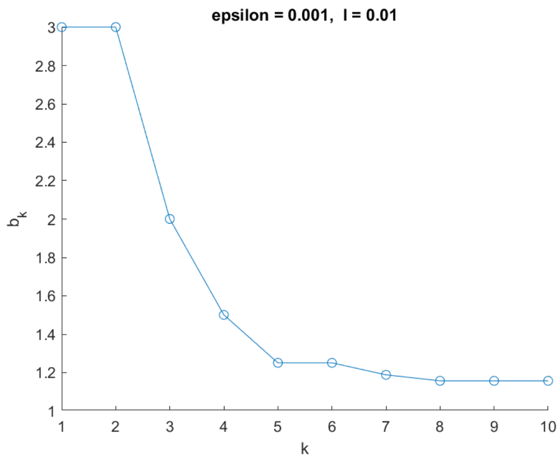
Παρατηρούμε ότι ο αριθμός κλήσεων είναι ίδιος σε όλες τις συναρτήσεις για δεδομένη ακρίβεια l. (Στα επόμενα διαγράμματα παρατηρούμε όντως ότι η τιμή του k προέκυψε ίδια σε κάθε συνάρτηση f_i για δεδομένα l και ε. Άρα, και ο αριθμός των κλήσεων της συνάρτησης θα είναι ίδιος σε κάθε περίπτωση.)

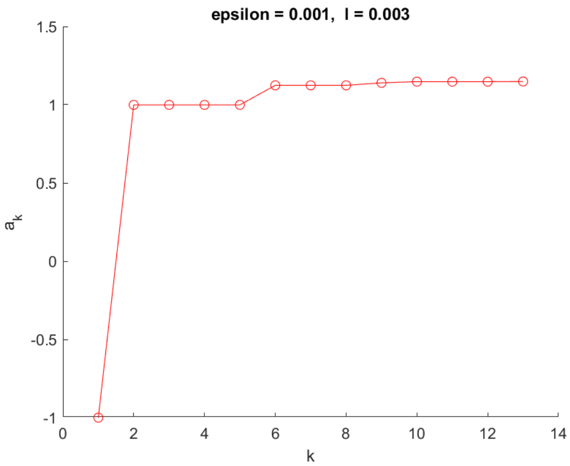
Σχεδιάζουμε, επιπλέον, για l={0.1, 0.01, 0.003, 0.0023} και ε = 0.001 τις γραφικές παραστάσεις των άκρων [a_k b_k] συναρτήσει του βήματος k για καθεμία αντικειμενική συνάρτηση:

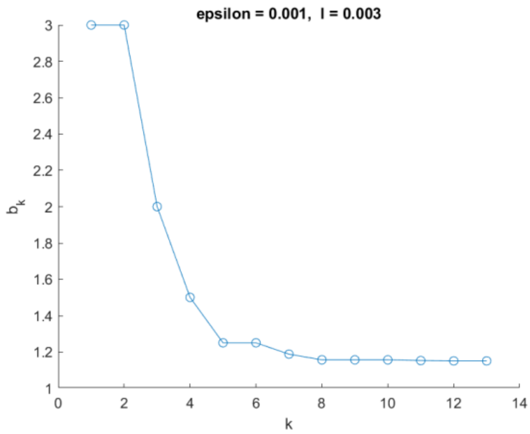
f_1

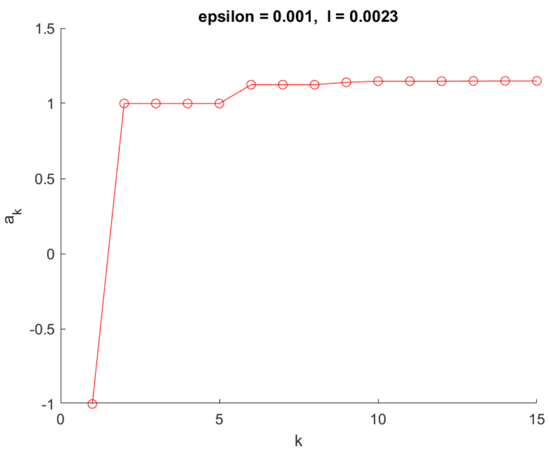


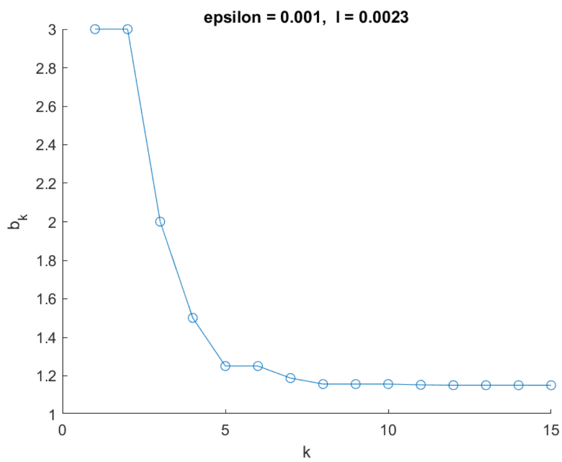

epsilon = 0.001, l = 0.01
a_k
k


epsilon = 0.001, l = 0.01
b_k
k

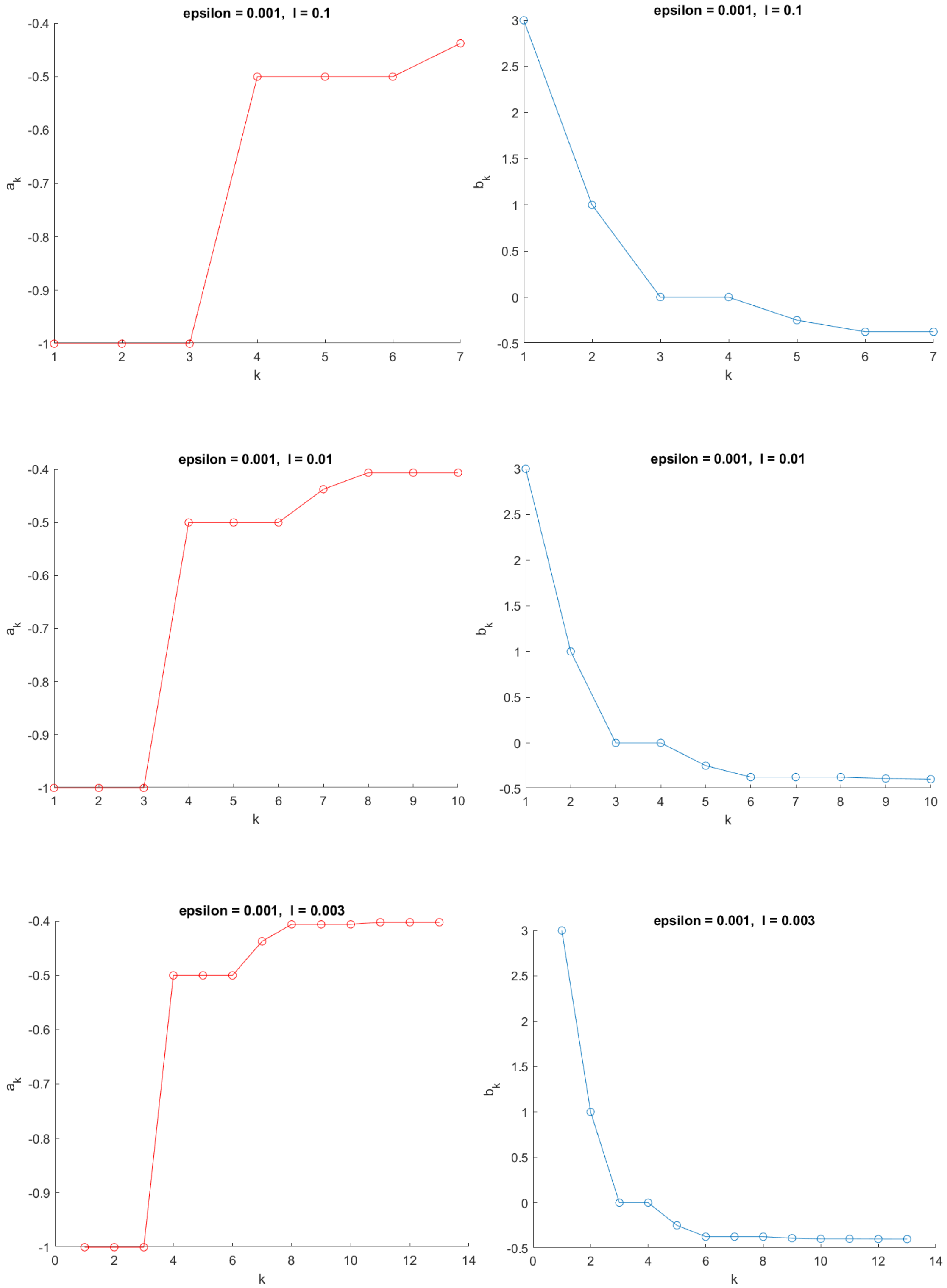

epsilon = 0.001, l = 0.003
a_k
k


epsilon = 0.001, l = 0.003
b_k
k

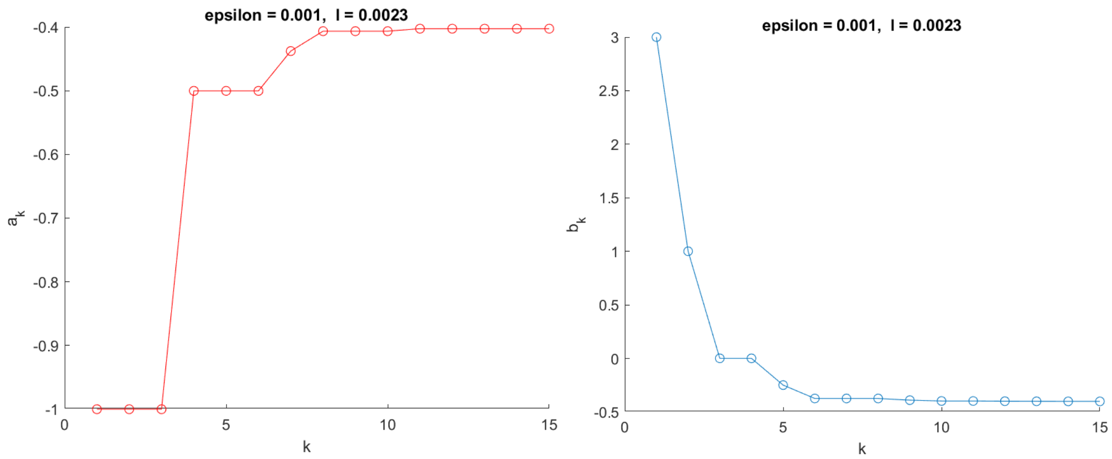

epsilon = 0.001, l = 0.0023
a_k
k


epsilon = 0.001, l = 0.0023
b_k
k

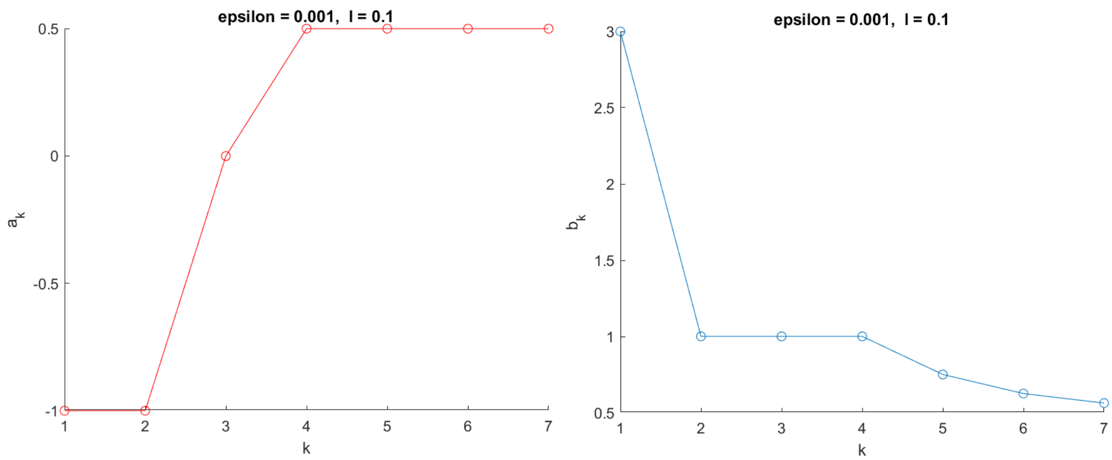
f_2

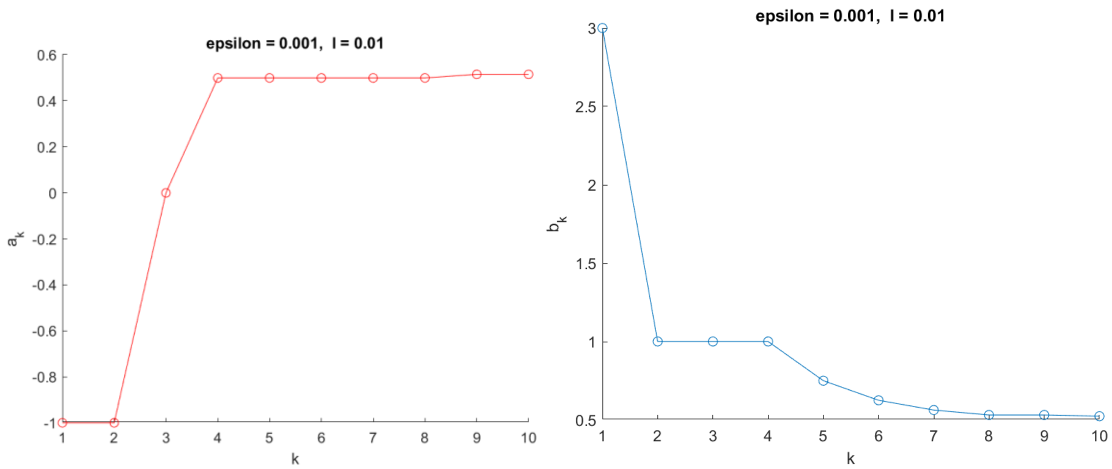


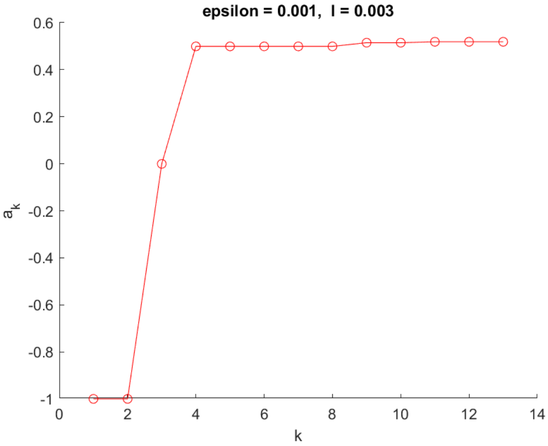
epsilon = 0.001, l = 0.1
epsilon = 0.001, l = 0.1
epsilon = 0.001, l = 0.01
epsilon = 0.001, l = 0.01
epsilon = 0.001, l = 0.003
epsilon = 0.001, l = 0.003
a_k
b_k
k

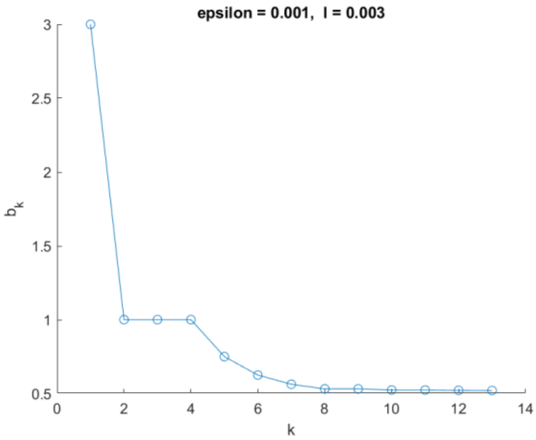

epsilon = 0.001, l = 0.0023
a_k
k
epsilon = 0.001, l = 0.0023
b_k
k

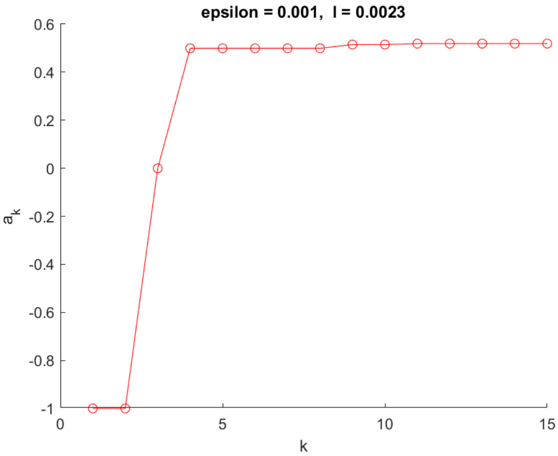
f_3

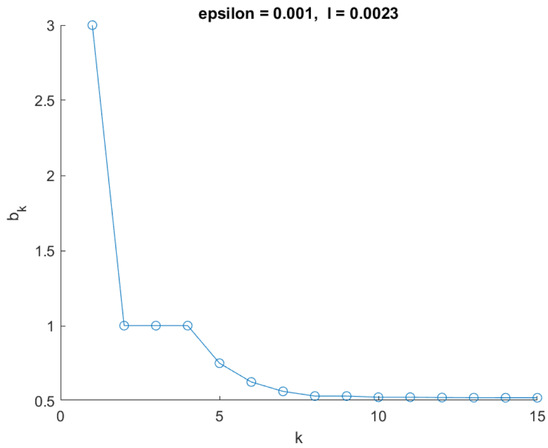

epsilon = 0.001, l = 0.1
a_k
k
epsilon = 0.001, l = 0.1
b_k
k


epsilon = 0.001, l = 0.01
a_k
k
epsilon = 0.001, l = 0.01
b_k
k









Από τα παραπάνω διαγράμματα προκύπτει το συμπέρασματα ότι όσο η τιμή του l μικραίνει, γίνεται μικρότερο και το τελικό εύρος του διαστήματος αναζήτησης x, όπου βρίσκεται το ελάχιστο της συνάρτησης.

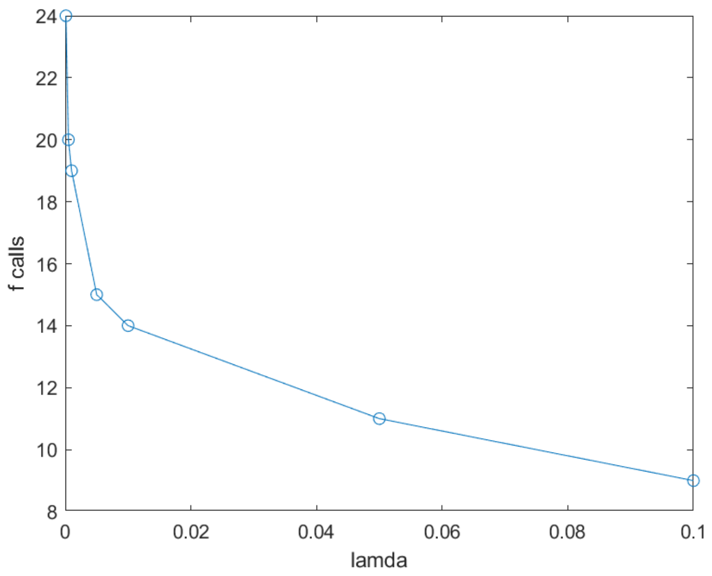
Επίσης, όσο μειώνεται το l (όμως πάντα l>2ε), τόσο ο αριθμός των επαναλήψεων k αυξάνεται, όπως ήταν αναμενόμενο.

Το k, επιπλέον, είναι ίδιο για δεδομένα l και ε για τις 3 συναρτήσεις

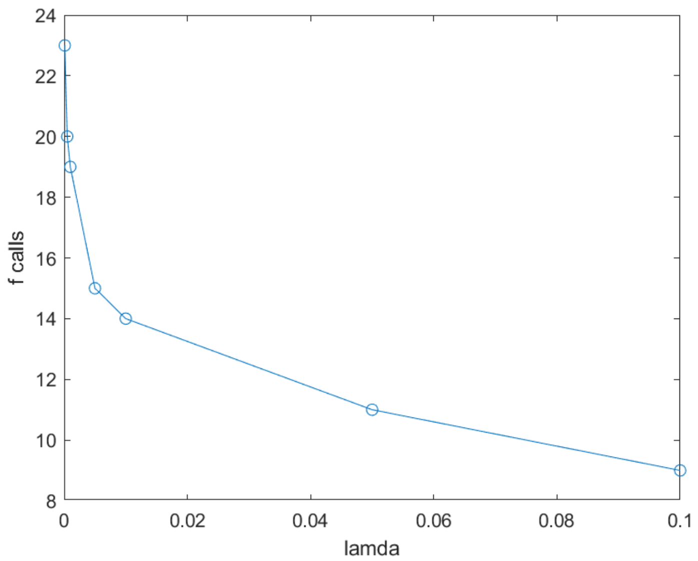
2. ***Μέθοδος Χρυσού Τομέα***

Επιλέγουμε l = {0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
Μελετούμε τους υπολογισμούς της αντικειμενικής συνάρτησης f_i καθώς το l μεταβάλλεται. Για κάθε l βρίσκουμε πόσες φορές κλήθηκε η συνάρτηση στον αλγόριθμο. Το διάγραμμα που προκύπτει για κάθε συνάρτηση είναι:

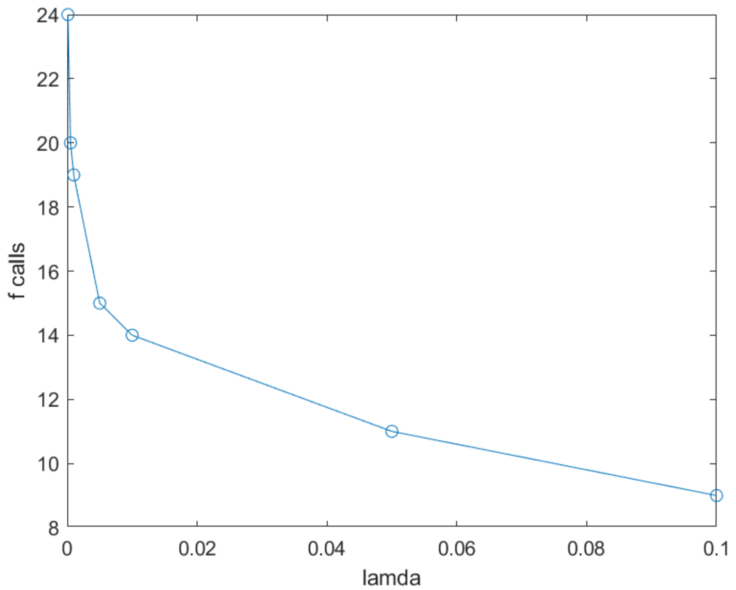
f_1



f_2

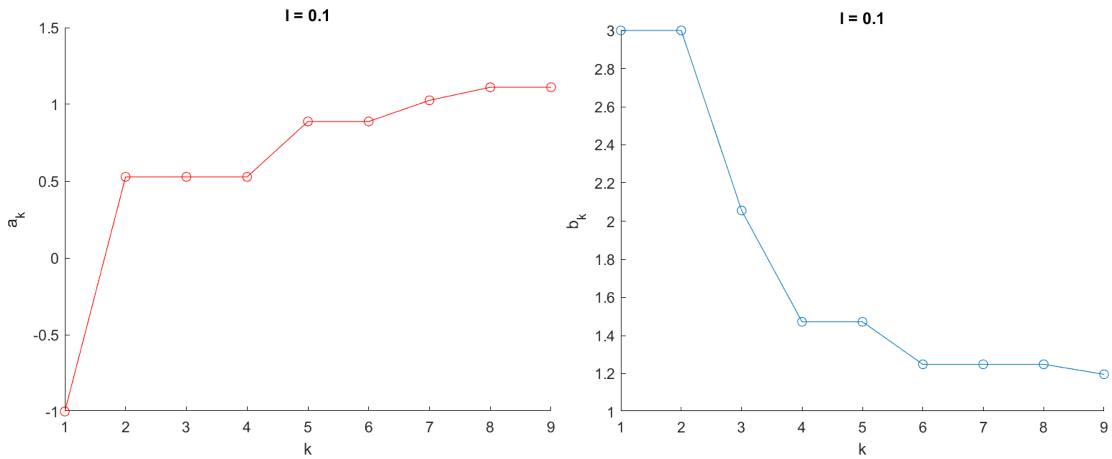


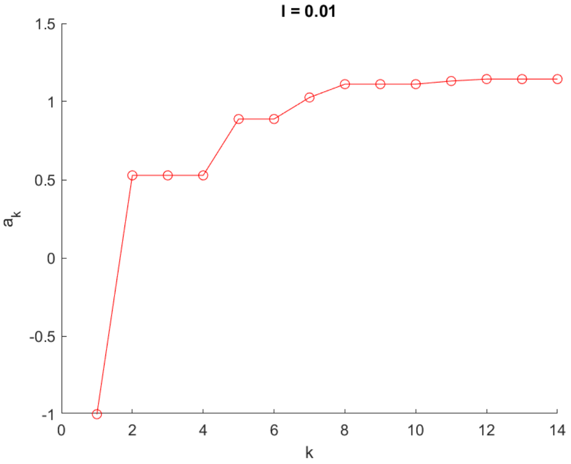
f_3

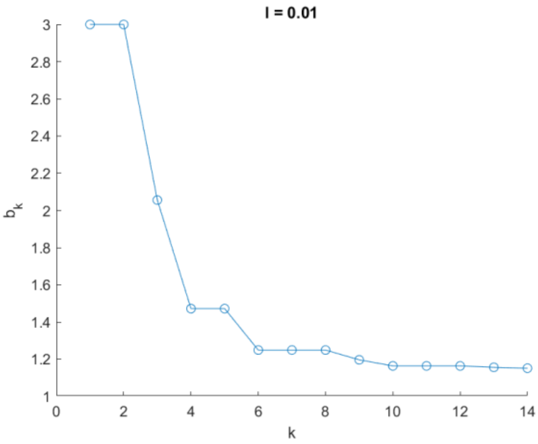


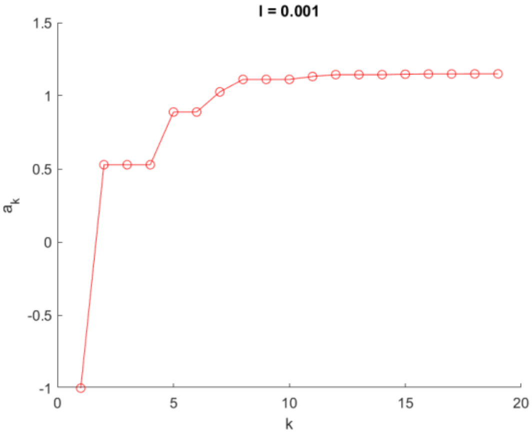
Σχεδιάζουμε, επιπλέον, για l = {0.1, 0.01, 0.001, 0.0001} τις γραφικές παραστάσεις των άκρων [a_k b_k] συναρτήσει του βήματος k για καθεμία αντικειμενική συνάρτηση:

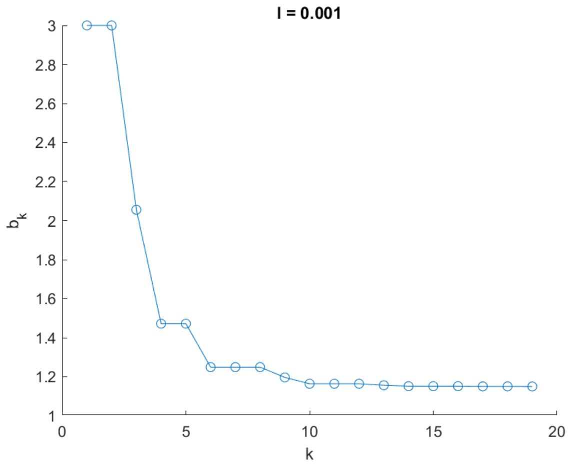
f_1

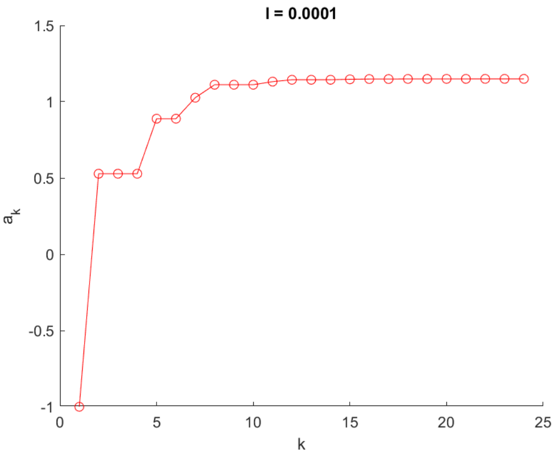


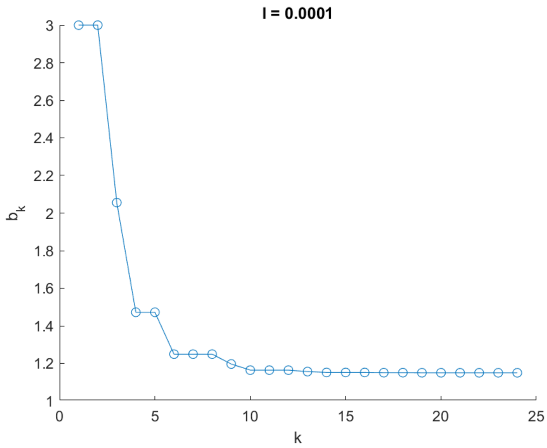

I = 0.01
1.5
1
0.5
0
-0.5
-1
a_k
0
2
4
6
8
10
12
14
k


I = 0.01
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
b_k
0
2
4
6
8
10
12
14
k

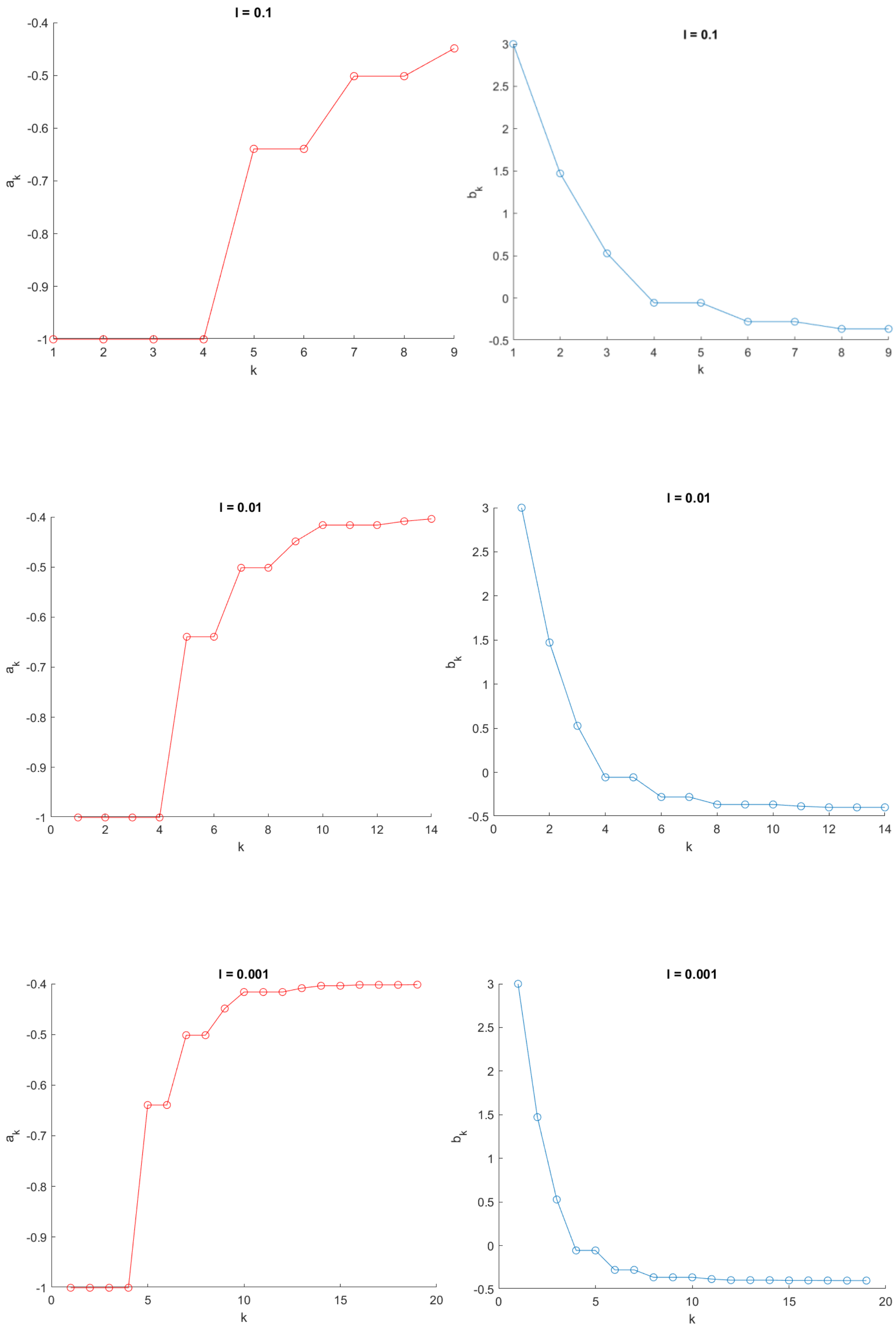

I = 0.001
1.5
1
0.5
0
-0.5
-1
a_k
0
5
10
15
20
k

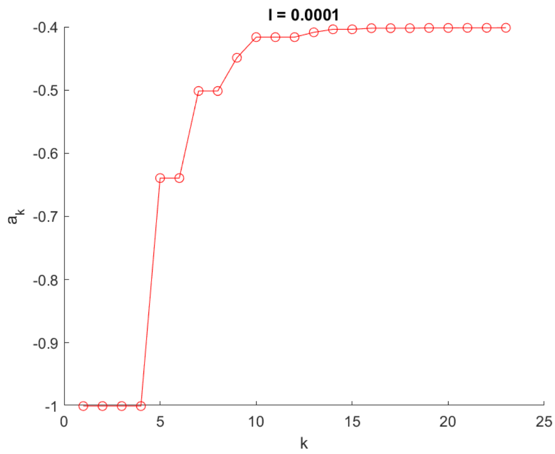

I = 0.001
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
b_k
0
5
10
15
20
k

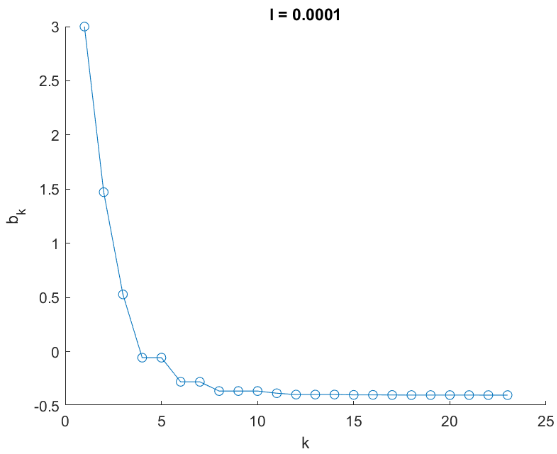

I = 0.0001
1.5
1
0.5
0
-0.5
-1
a_k
0
5
10
15
20
25
k


I = 0.0001
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
b_k
0
5
10
15
20
25
k

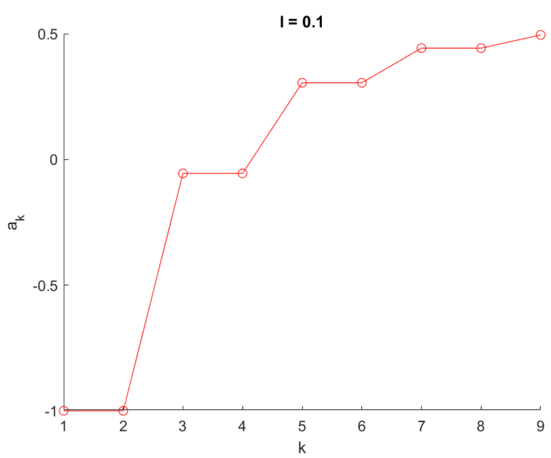
f_2

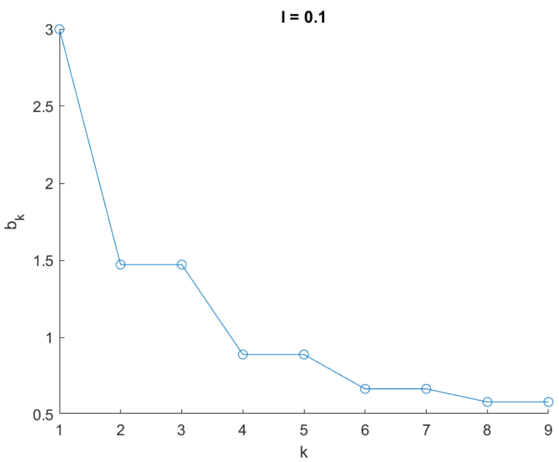


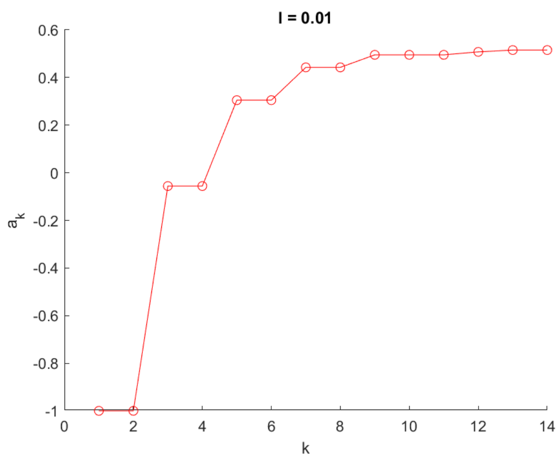


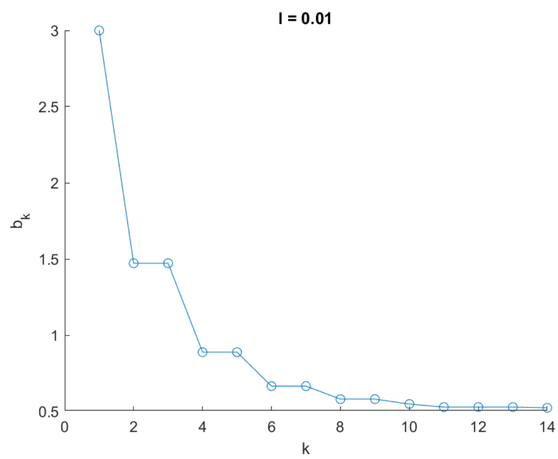


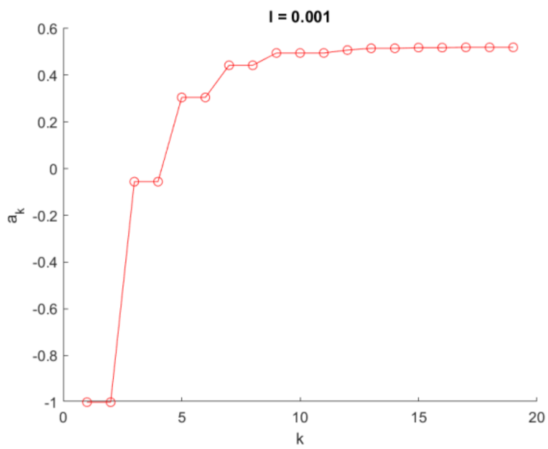
f₃

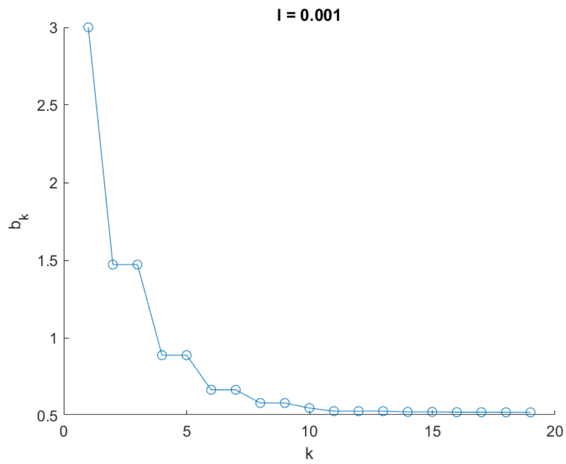


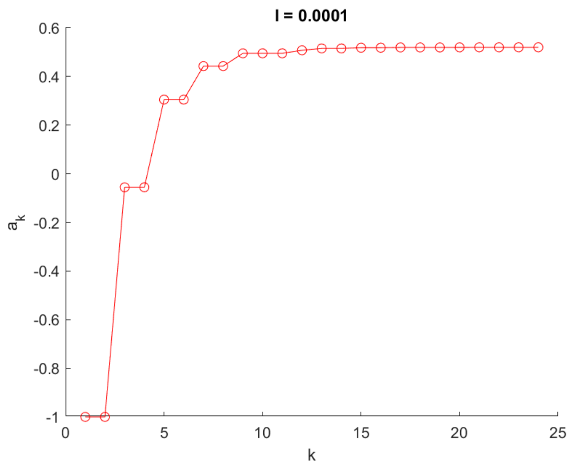


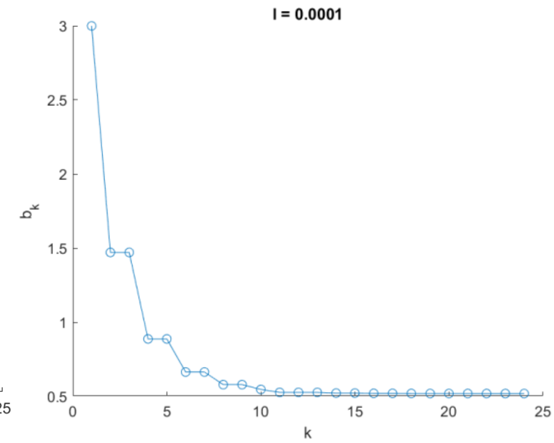





I = 0.001
a_k
k


I = 0.001
b_k
k


I = 0.0001
a_k
k


I = 0.0001
b_k
k

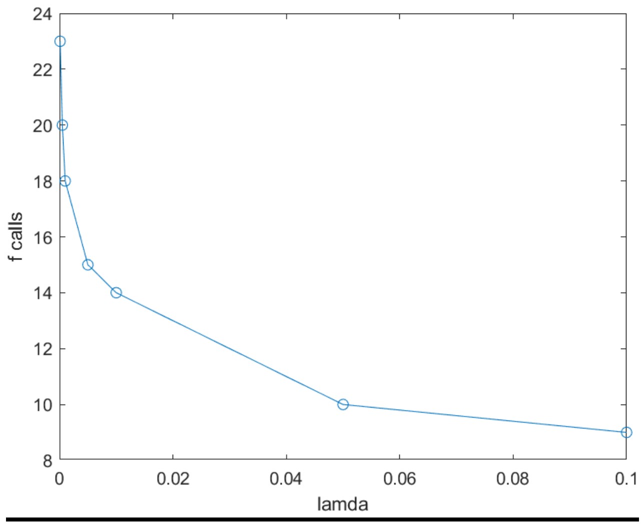
3. <u>Μέθοδος Fibonacci</u>

Επιλέγουμε εξ αρχής το συνολικό αριθμό n των υπολογισμών της συνάρτησης ώστε $F_n > (b\text{-}a)/l$

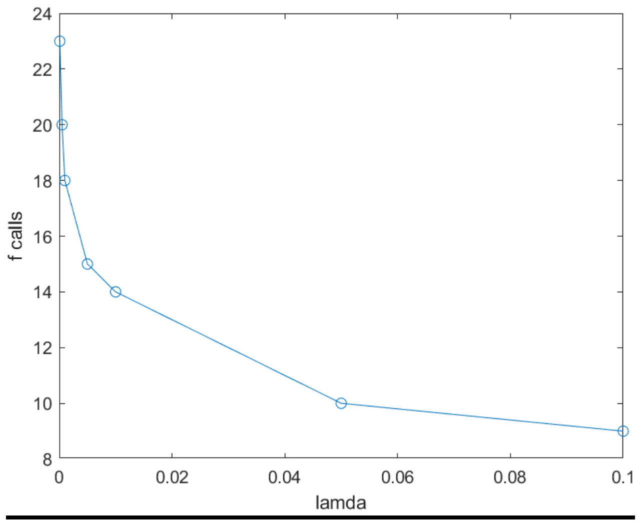
Μεταβάλλουμε το τελικό εύρος l.

Επιλέγουμε l = {0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
Μελετούμε τους υπολογισμούς της αντικειμενικής συνάρτησης f_i καθώς το l μεταβάλλεται και ε=0,001. Για κάθε l βρίσκουμε πόσες φορές κλήθηκε η συνάρτηση στον αλγόριθμο. Το διάγραμμα που προκύπτει για κάθε συνάρτηση είναι:

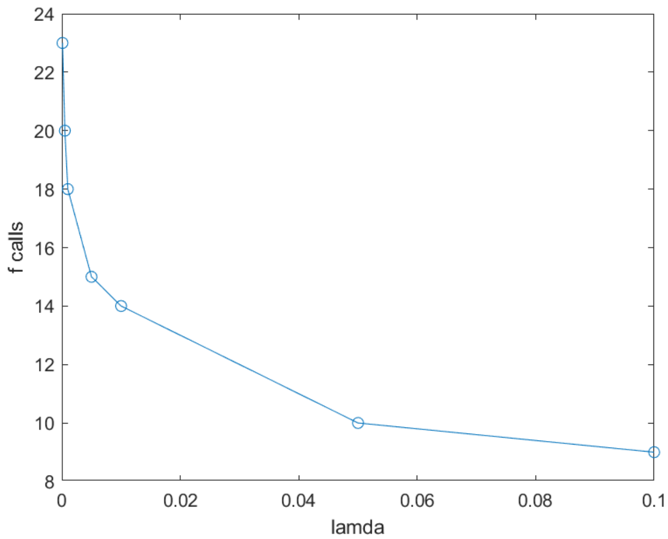
f_1



f_2

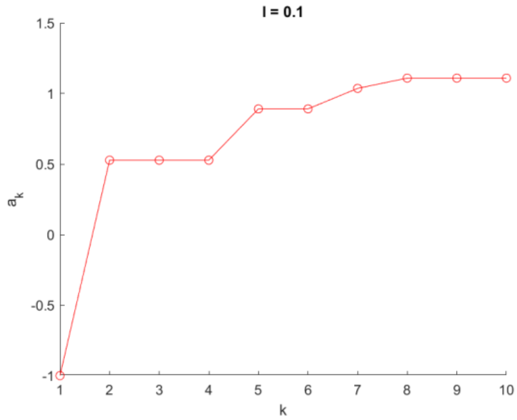


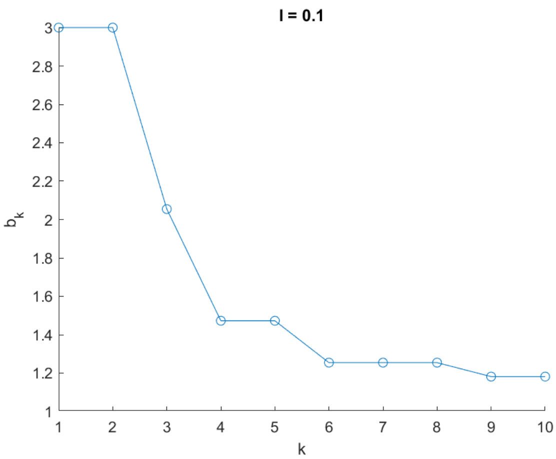
f₃

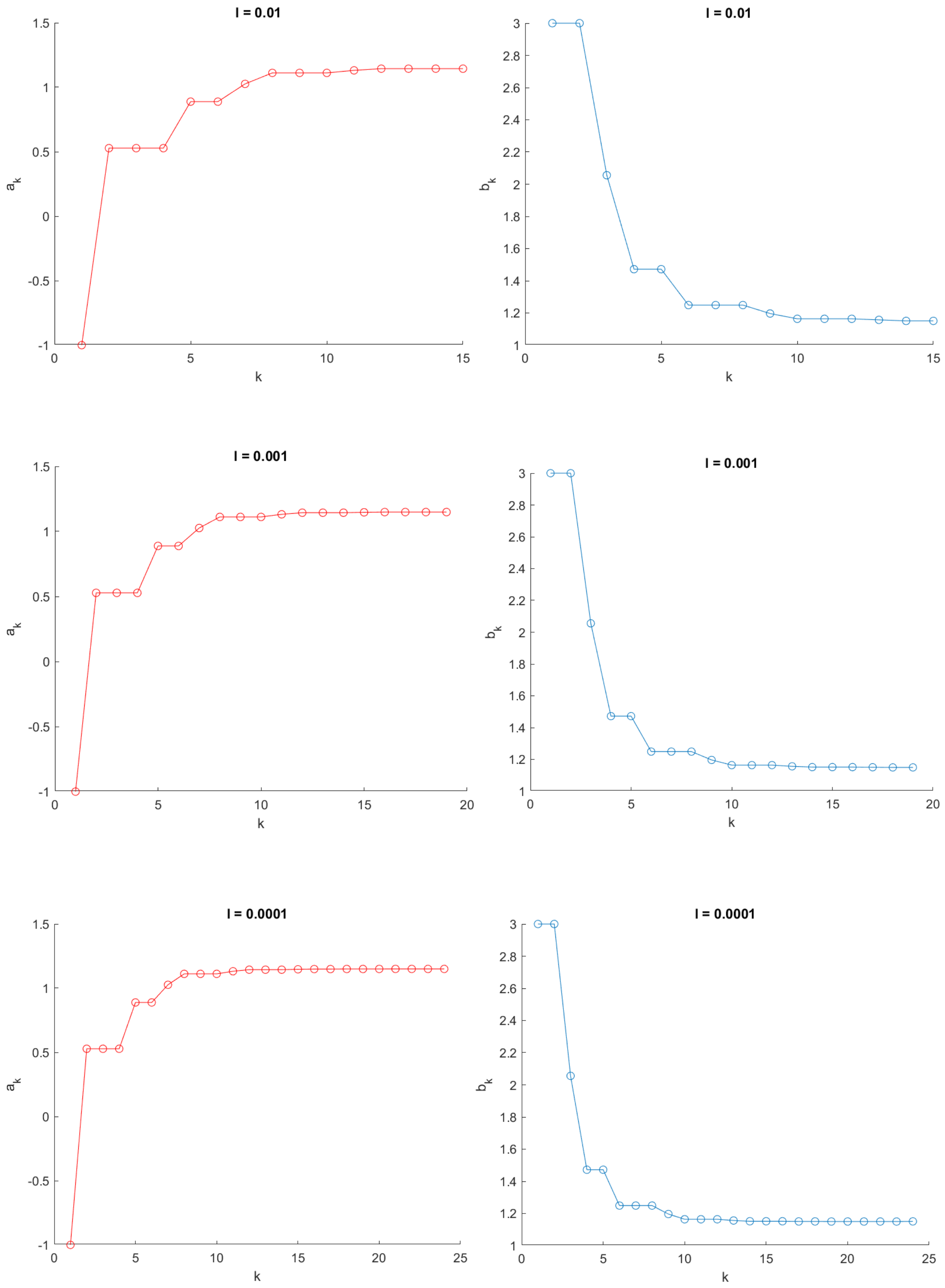


Σχεδιάζουμε, επιπλέον, για l = {0.1, 0.01, 0.001, 0.0001}
και ε = 0.001 τις γραφικές παραστάσεις των άκρων [a_k b_k] συναρτήσει του βήματος k για καθεμία αντικειμενική συνάρτηση:

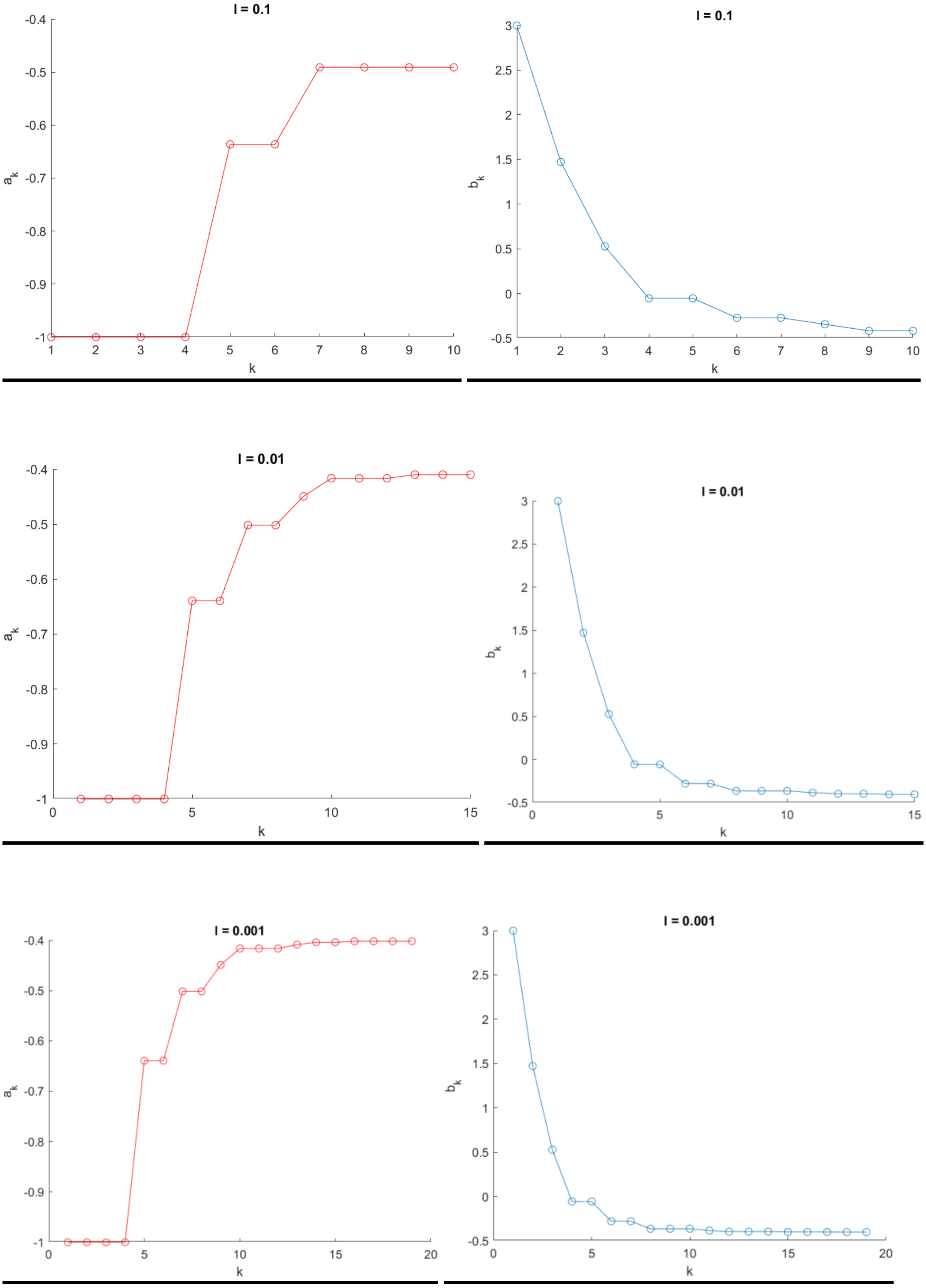
f₁

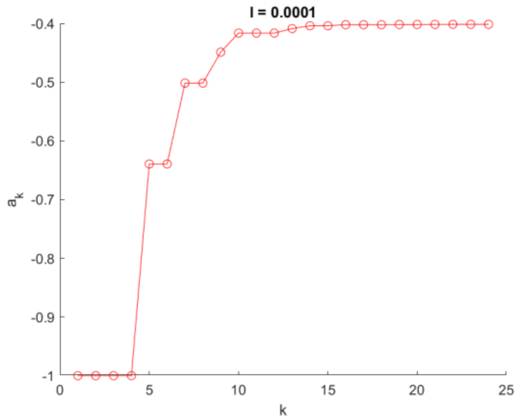


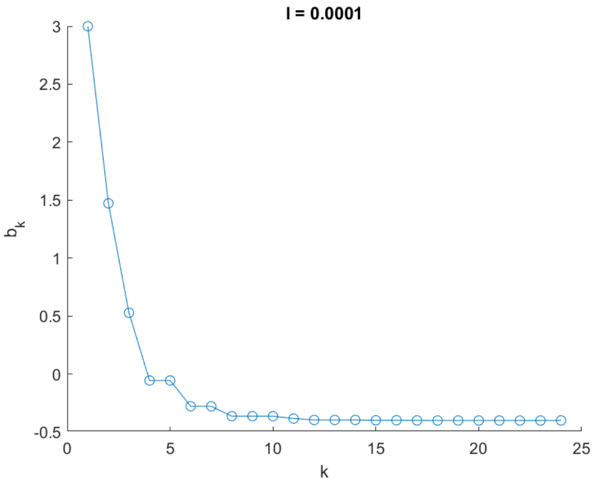



l = 0.01
a_k
k
l = 0.01
b_k
k
l = 0.001
a_k
k
l = 0.001
b_k
k
l = 0.0001
a_k
k
l = 0.0001
b_k
k

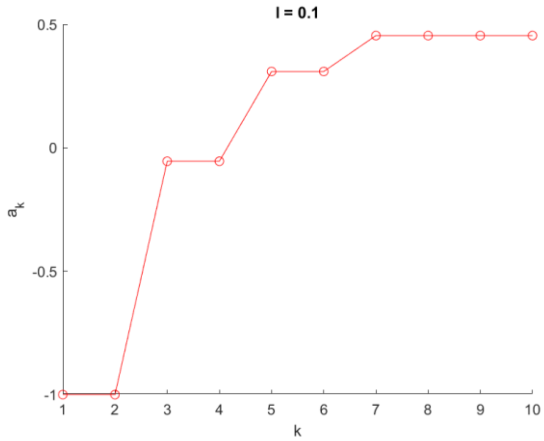
f_2

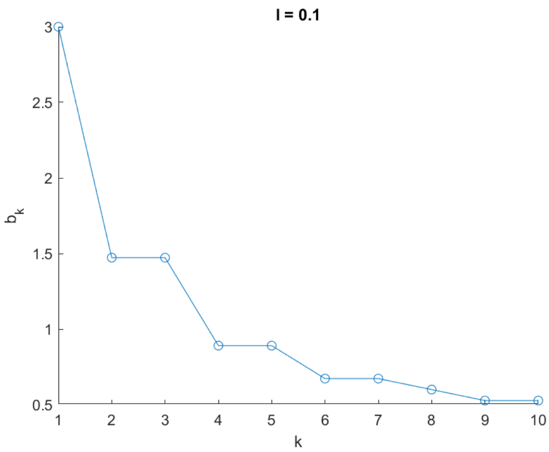


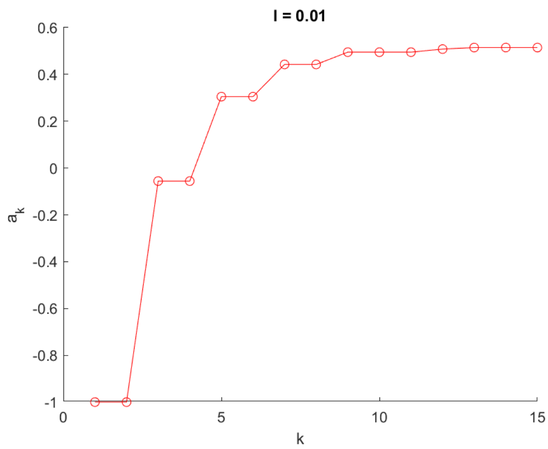


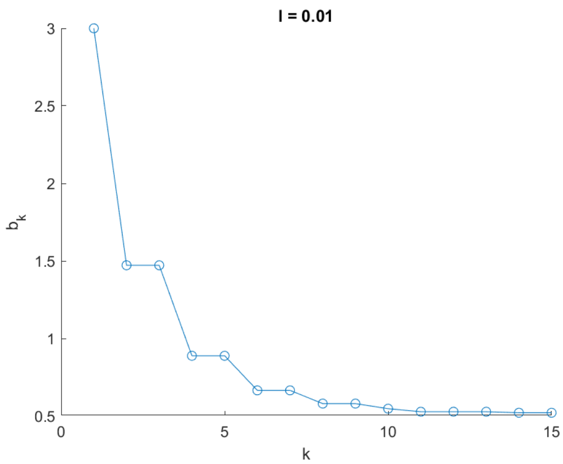


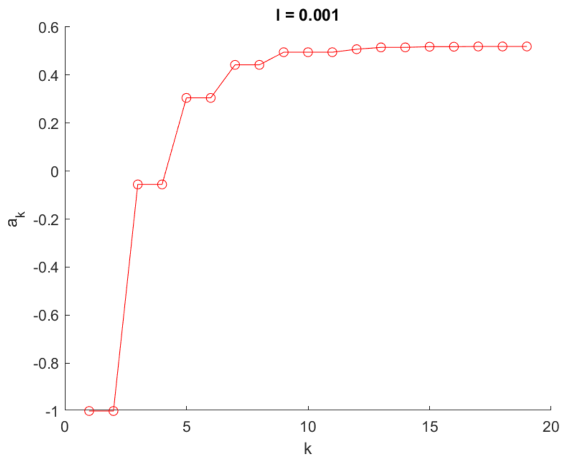
f3

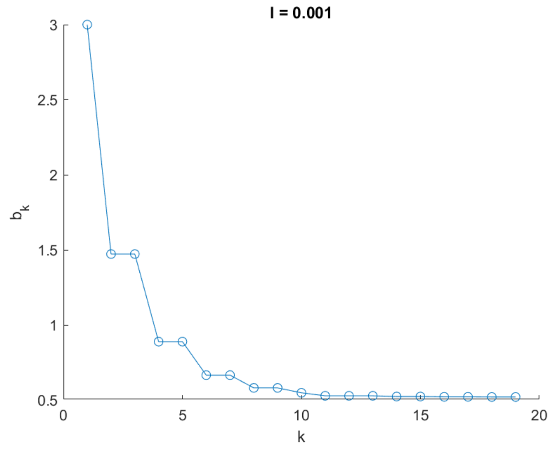


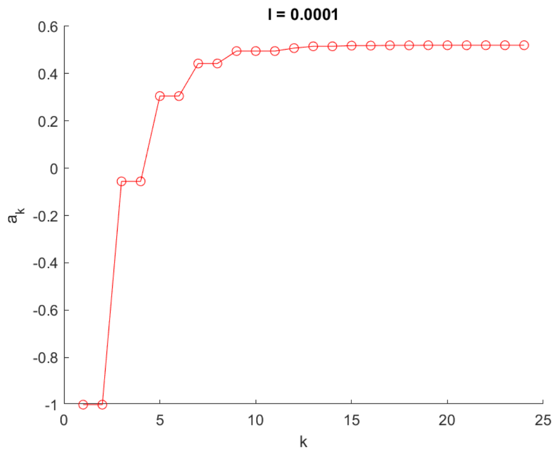


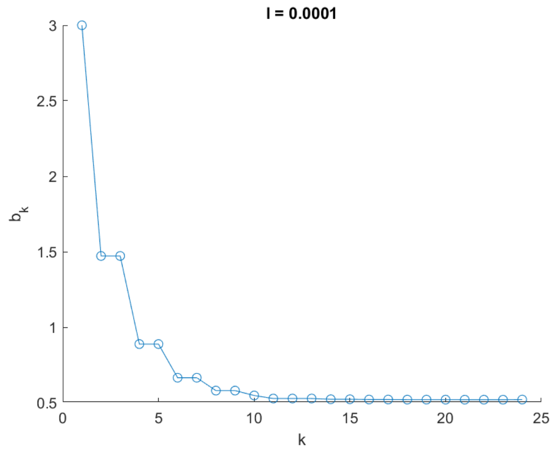





I = 0.001
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
a_k
0
5
10
15
20
k


I = 0.001
3
2.5
2
1.5
1
0.5
b_k
0
5
10
15
20
k


I = 0.0001
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
a_k
0
5
10
15
20
25
k


I = 0.0001
3
2.5
2
1.5
1
0.5
b_k
0
5
10
15
20
25
k

4. ***Μέθοδος Διχοτόμου με χρήση Παραγώγων***

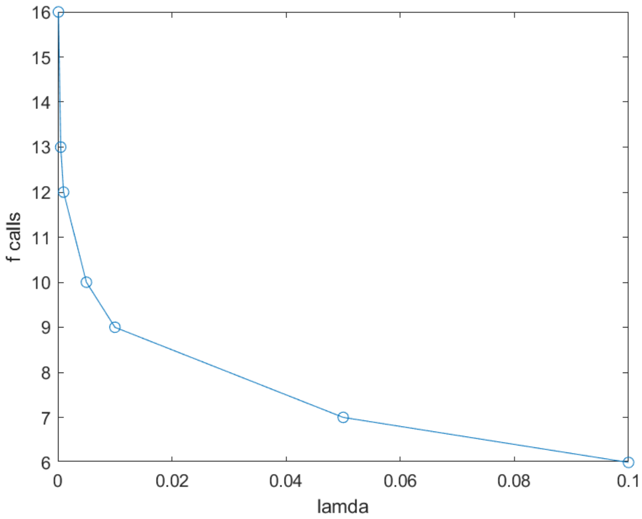
Στη μέθοδο αυτή επιλέγουμε εξ αρχής το συνολικό αριθμό n των υπολογισμών της συνάρτησης ώστε να πληρείται η συνθήκη:

$(\frac{1}{2})^n \leq l/(b\text{-}a)$

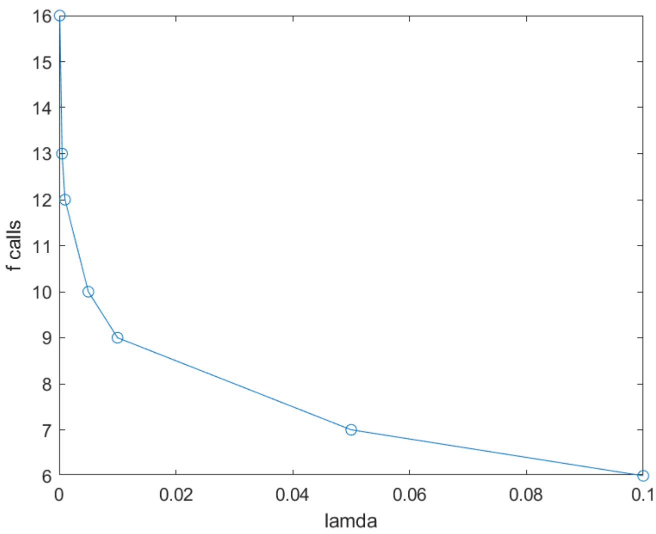
Μεταβάλλουμε το τελικό εύρος l.

Επιλέγουμε l = {0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001}
Μελετούμε τους υπολογισμούς της αντικειμενικής συνάρτησης f_i καθώς το l μεταβάλλεται. Για κάθε l βρίσκουμε πόσες φορές κλήθηκε η συνάρτηση στον αλγόριθμο. Το διάγραμμα που προκύπτει για κάθε συνάρτηση είναι:

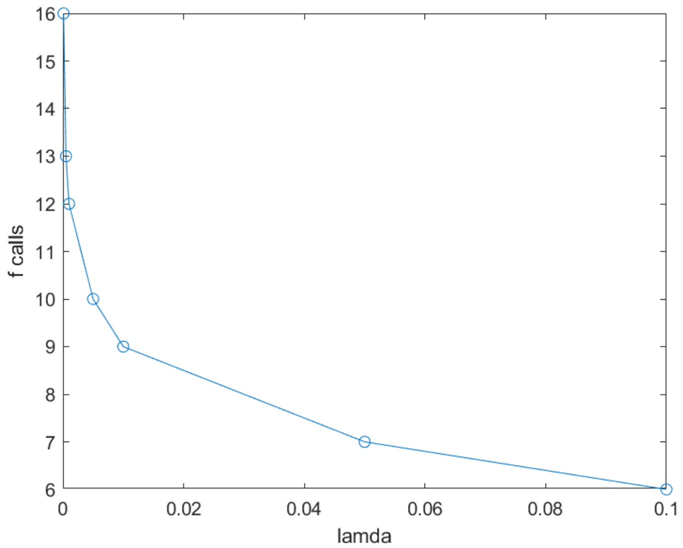
f_1



f_2

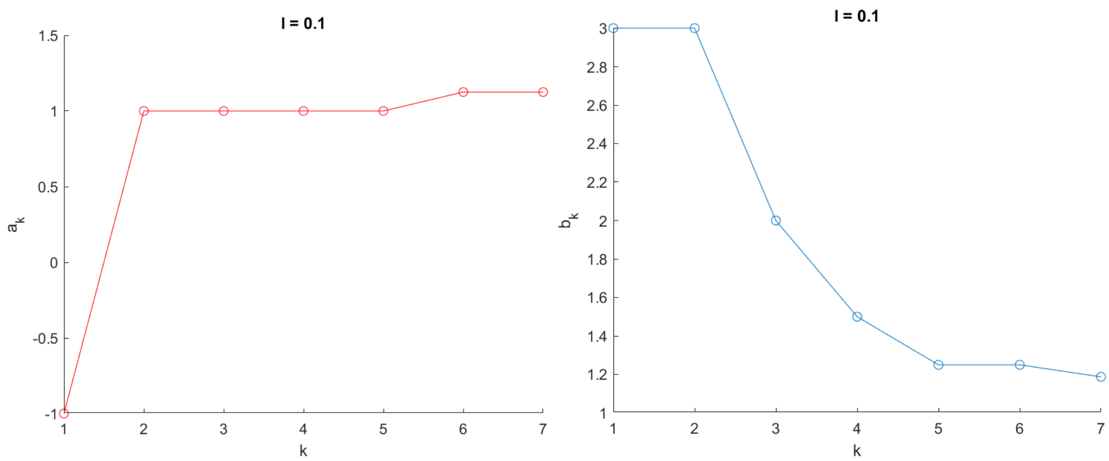


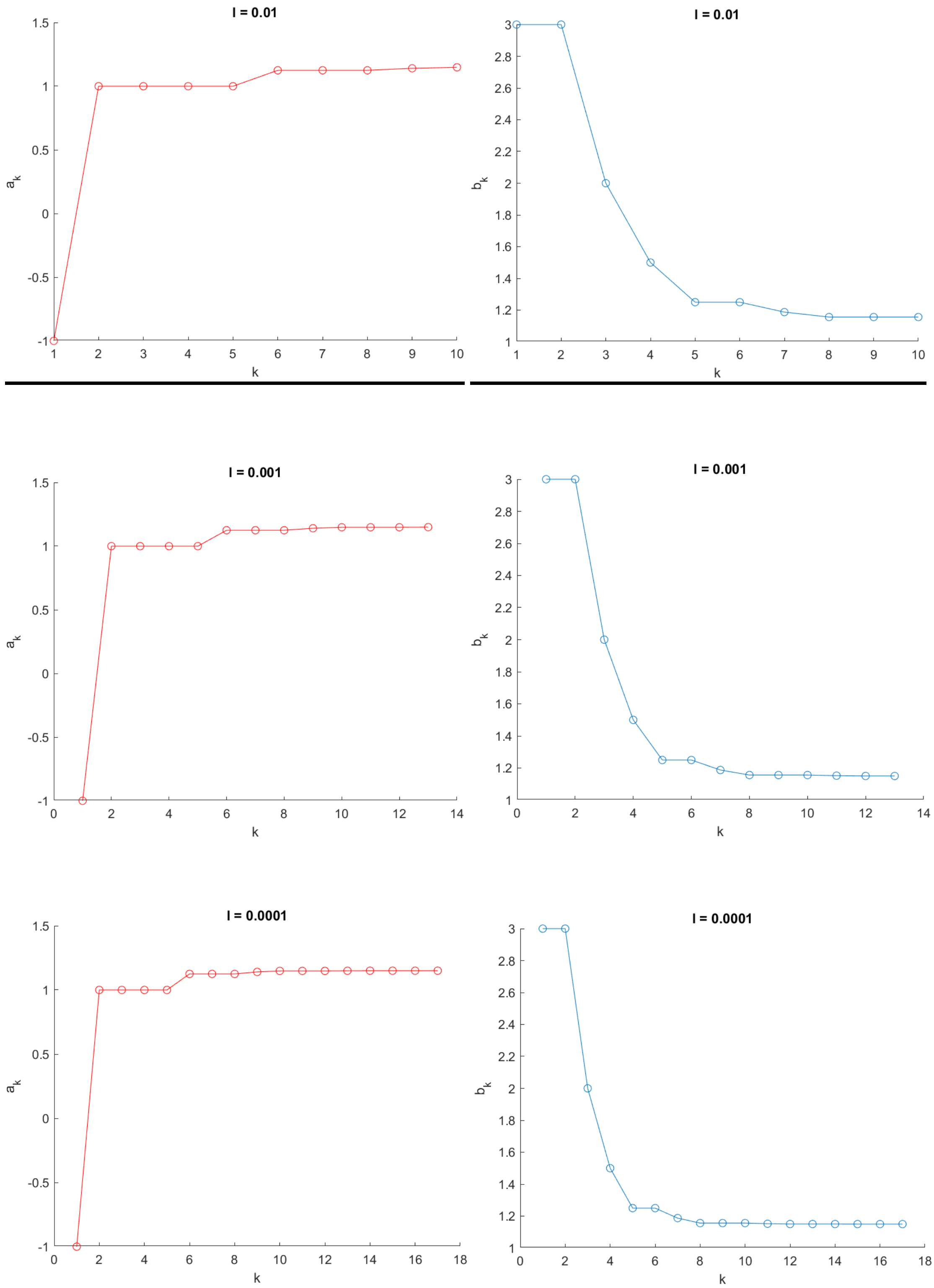
f_3



Σχεδιάζουμε, επιπλέον, για $l = \{0.1, 0.01, 0.001, 0.0001\}$ τις γραφικές παραστάσεις των άκρων $[a_k\ \ b_k]$ συναρτήσει του βήματος k για καθεμία αντικειμενική συνάρτηση:

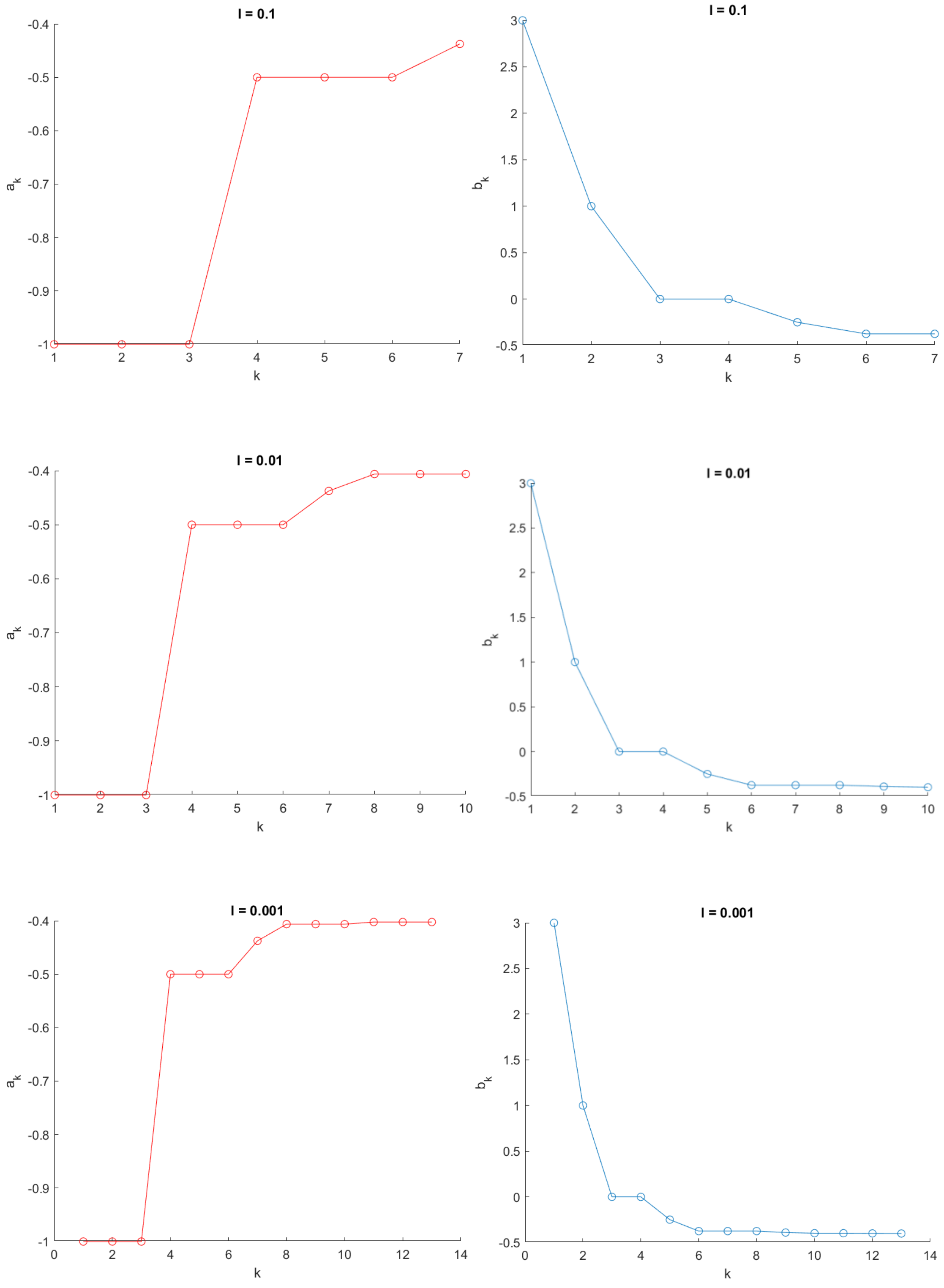
f_1

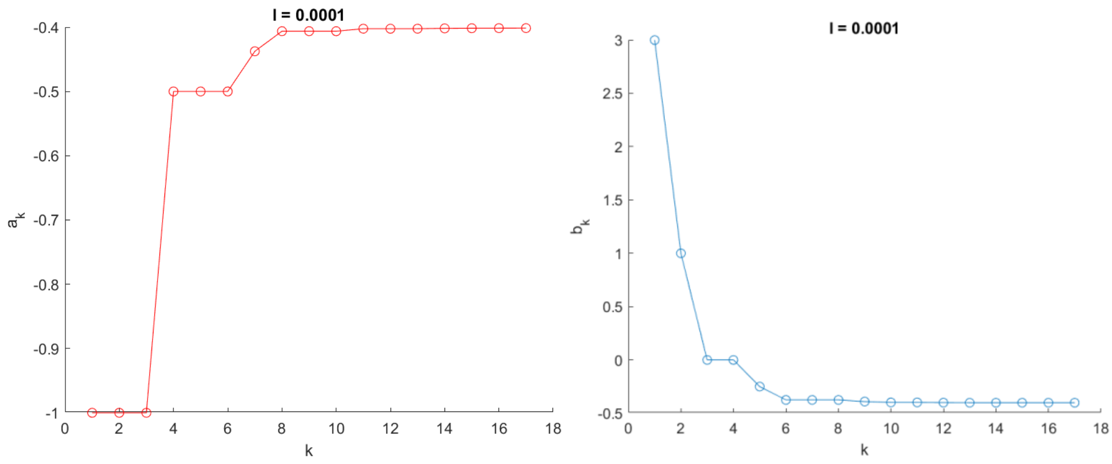




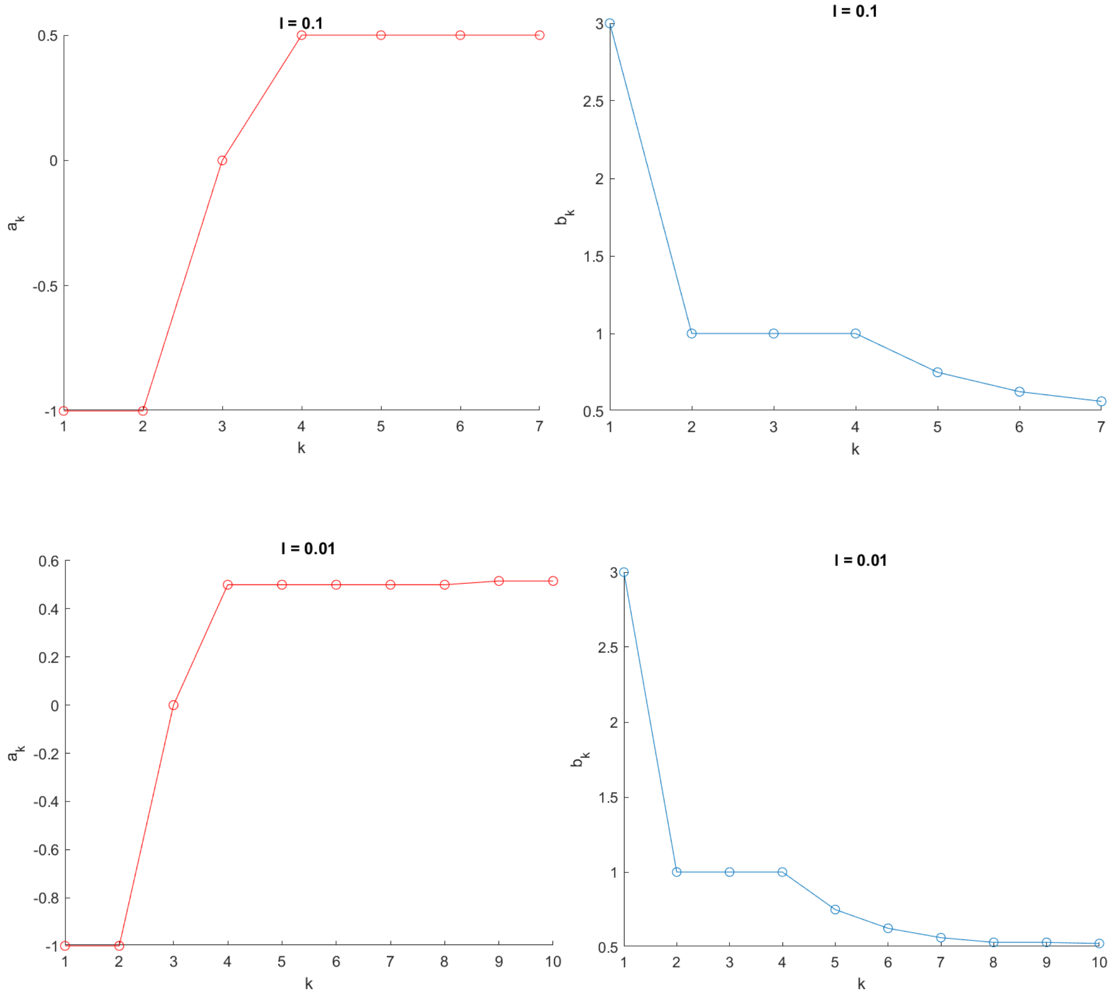
l = 0.01
a_k
k
l = 0.01
b_k
k
l = 0.001
a_k
k
l = 0.001
b_k
k
l = 0.0001
a_k
k
l = 0.0001
b_k
k

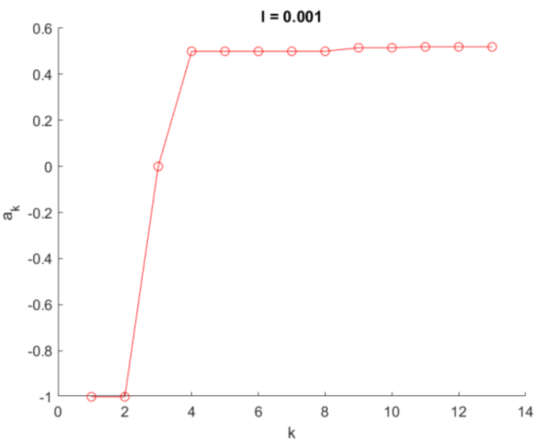
f_2

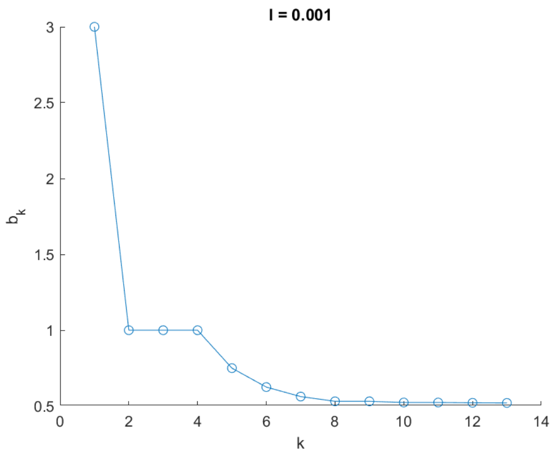


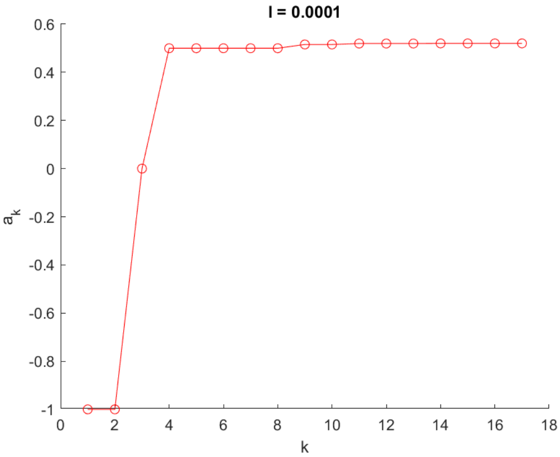


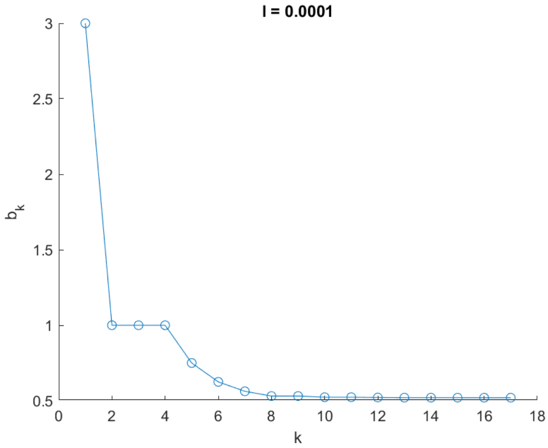
f3











<u>Σύγκριση Αποδοτικότητας των μεθόδων</u>

Από τα παραπάνω διαγράμματα διαπιστώνουμε πως αποδοτικότερη μέθοδος είναι η Fibonacci, ακολουθεί η μέθοδος του Χρυσού Τομέα και έπειτα η μέθοδος της διχοτόμου χωρίς παραγώγους, λαμβάνοντας υπόψην το μέγεθος του τελικού διαστήματος ανάζητησης $[a_k\, b_k]$ (για k = τελικό) και τον αριθμό των υπολογισμών της αντικειμενικής συνάρτησης. Συγκεκριμένα, όσο μικρότερο είναι το n τόσο αποτελεσματικότερος είναι ο αλγόριθμος για δεδομένο πηλίκο.

Επίσης, η μέθοδος της διχοτόμου με χρήση παραγώγων απαιτεί τις λιγότερες κλήσεις σχετικά με τις υπόλοιπες μεθόδους και άρα είναι αποτελεσματικότερη.