<u>ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ</u>

<u>2η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ</u>

ΟΝ/ΜΟ: Νήρας Δημήτρης

ΑΕΜ: 8057

Θέμα 1.

Με τη βοήθεια του αλγορίθμου 5.1.1 του βιβλίου, υλοποιήσαμε τον αλγόριθμο της μεθόδου της διχοτόμου στο Matlab. Με τη χρήση της εντολής syms x δηλώσαμε ως μεταβλητή των συναρτήσεων το x και με τις εντολές f1=(x-2)^3+(x-3)^2 , f2=(x-2)^2+sin(x) και f3=sqrt(x+1)+(x^2-2)*log(x+1) δηλώσαμε τις συναρτήσεις f1,f2,f3 αντίστοιχα.

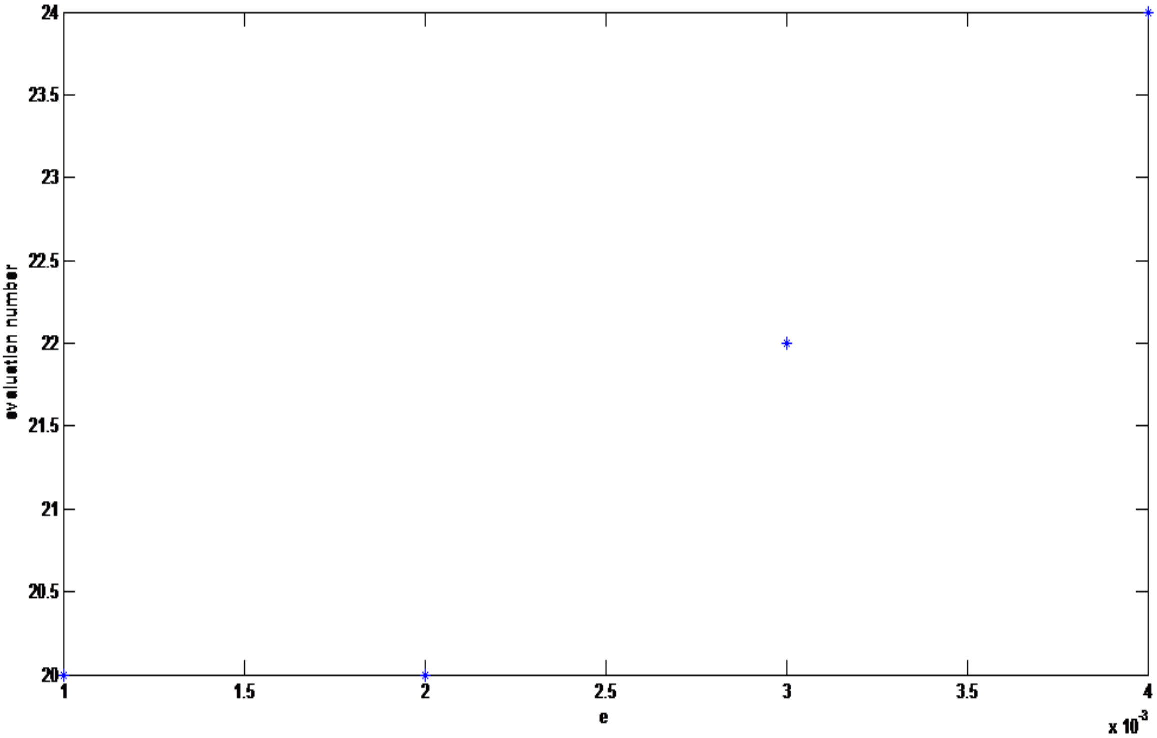
Ο αλγόριθμος ονομάζεται alg_dixotomou(f,e,l,a,b) και έχει ως ορίσματα τη συνάρτηση f για την οποία ψάχνουμε το ελάχιστο, την απόσταση e από τη διχοτόμο, το τελικό εύρος του διαστήματος αναζήτησης l και τα αρχικά όρια του διαστήματος αναζήτησης a,b.

Εφαρμόζοντας τον αλγόριθμο για τη συνάρτηση f1, με e=0.001 και l=0.01, καταλήγουμε σε ένα τελικό διάστημα [2.5472 2.5541], το οποίο βλέπουμε πως είναι αποδεκτό, αφού το τελικό εύρος του είναι μικρότερο από l, όπως προβλέπει και ο αλγόριθμος. Εφαρμόζοντάς τον και για τις f2, f3 καταλήγουμε στα διαστήματα [2.3488 2.3567] και [0.6209 0.6287] αντίστοιχα, τα οποία βλέπουμε πως είναι αποδεκτά και τα δύο, αφού το τελικό τους εύρος είναι μικρότερο από το l.

Ο αλγόριθμος εκτός από το τελικό διάστημα επιστρέφει και τον αριθμό c των υπολογισμών της αντικειμενικής συνάρτησης, καθώς και τον αριθμό k των επαναλήψεων. Ως αριθμό υπολογισμών της αντικειμενικής συνάρτησης θεωρούμε το πόσες φορές κλήθηκε η συνάρτηση f(x).

Κρατώντας σταθερό το τελικό εύρος l, μεταβάλλουμε την απόσταση e από την διχοτόμο από 0.001 μέχρι 0.004 με βήμα 0.001 και παρατηρούμε τις μεταβολές των υπολογισμών c των αντικειμενικών συναρτήσεων και για τις 3 συναρτήσεις, δημιουργώντας και τις αντίστοιχες γραφικές παραστάσεις.

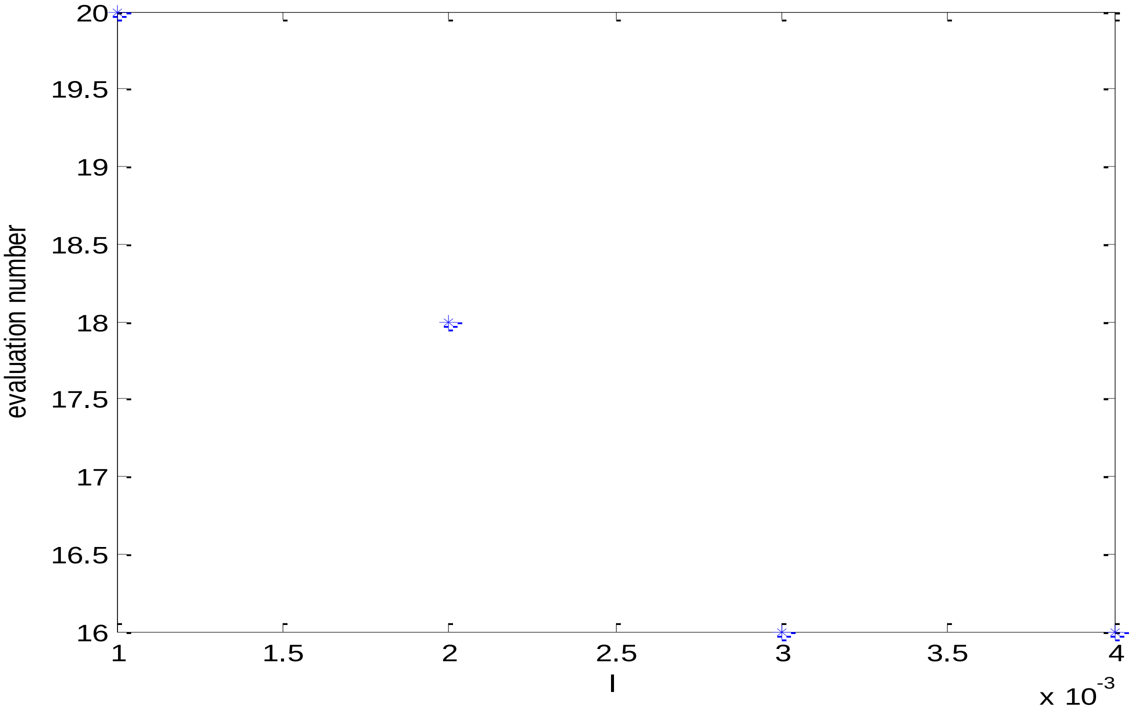
Επομένως για την f1 έχουμε:



Οι ίδιες γραφικές παραστάσεις προκύπτουν και για τις f2,f3. Παρατηρώντας τα αποτελέσματα, βλέπουμε πως καθώς αυξάνεται η απόσταση e από τη διχοτόμο, αυξάνεται και ο αριθμός c των υπολογισμών της αντικειμενικής συνάρτησης. Αυτό συμβαίνει διότι καθώς μεγαλώνει η τιμή e, μεγαλώνει αντίστοιχα και η απόσταση μεταξύ των εσωτερικών σημείων x1,x2 και επομένως απαιτείται μεγαλύτερος αριθμός βημάτων για την επίτευξη ενός τελικού εύρους μικρότερου του 0.01.

Κρατώντας σταθερή τώρα την απόσταση e από την διχοτόμο, μεταβάλλουμε το τελικό εύρος του διαστήματος l από 0.01 μέχρι 0.04 με βήμα 0.01 και παρατηρούμε τις μεταβολές των υπολογισμών c των αντικειμενικών συναρτήσεων και για τις 3 συναρτήσεις, δημιουργώντας και τις αντίστοιχες γραφικές παραστάσεις.

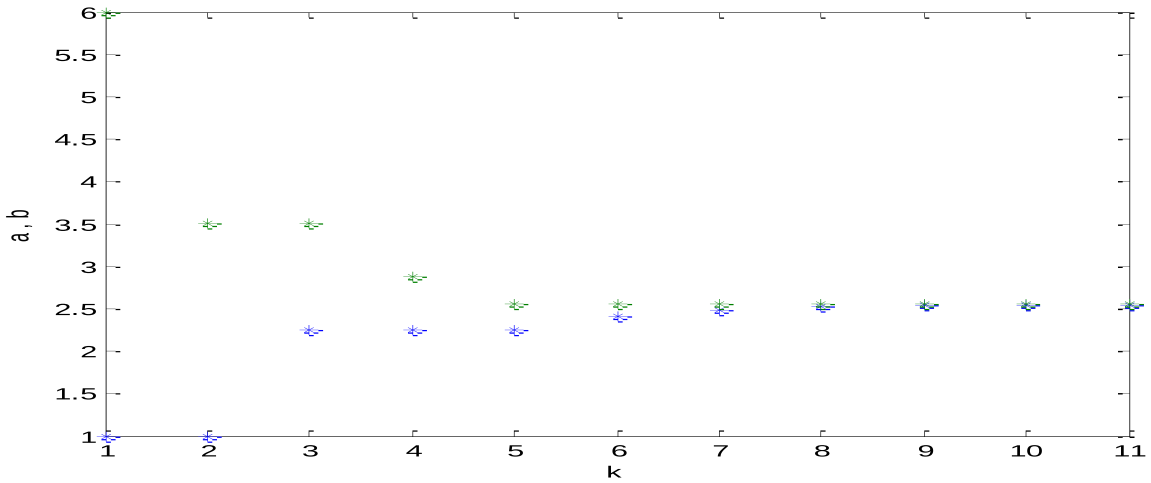
Επομένως για την f1 έχουμε:

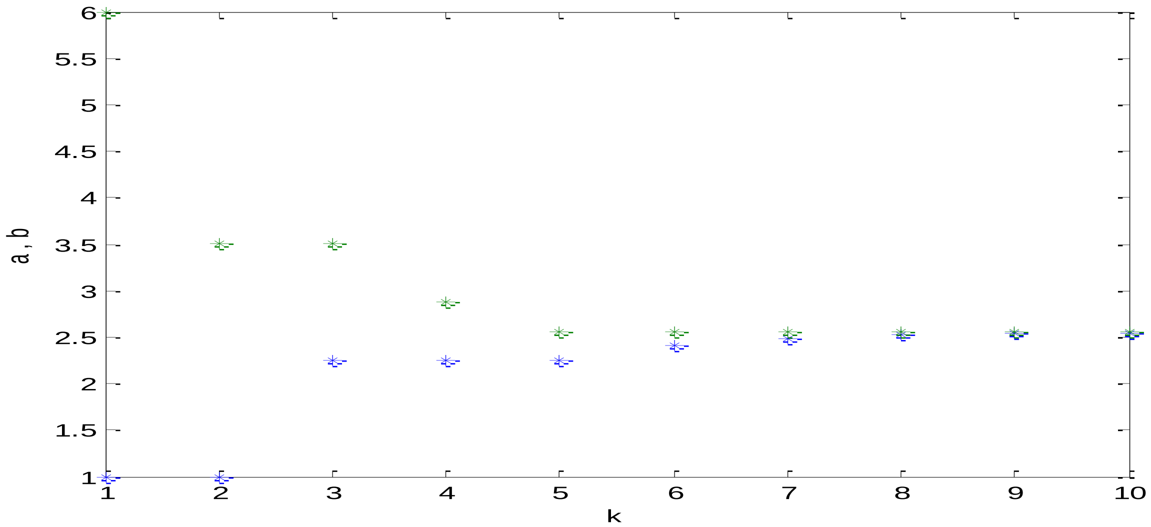


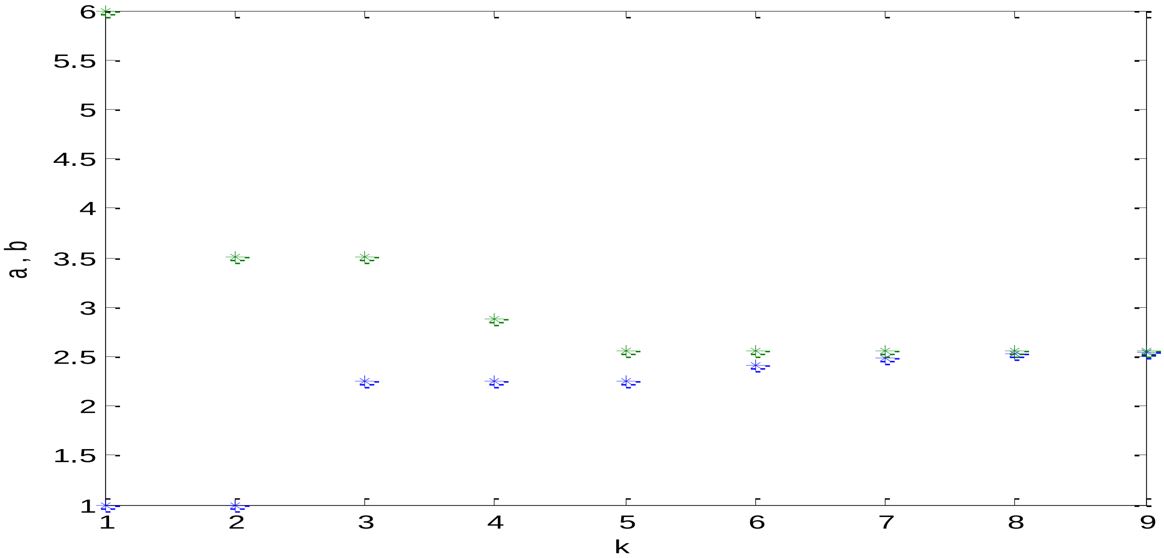
Οι ίδιες γραφικές παραστάσεις προκύπτουν και για τις f2,f3. Παρατηρώντας τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, ο αριθμός c των υπολογισμών της αντικειμενικής συνάρτησης μειώνεται. Αυτό συμβαίνει διότι μειώνοντας το τελικό εύρος αναζήτησης, μειώνουμε ουσιαστικά την τελική ακρίβεια που θέλουμε να έχουμε στον υπολογισμό του ελάχιστου της συνάρτησης και έτσι μειώνεται και ο αριθμός επαναλήψεων της μεθόδου, με συνέπεια να μειωθούν και οι υπολογισμοί της αντικειμενικής συνάρτησης.

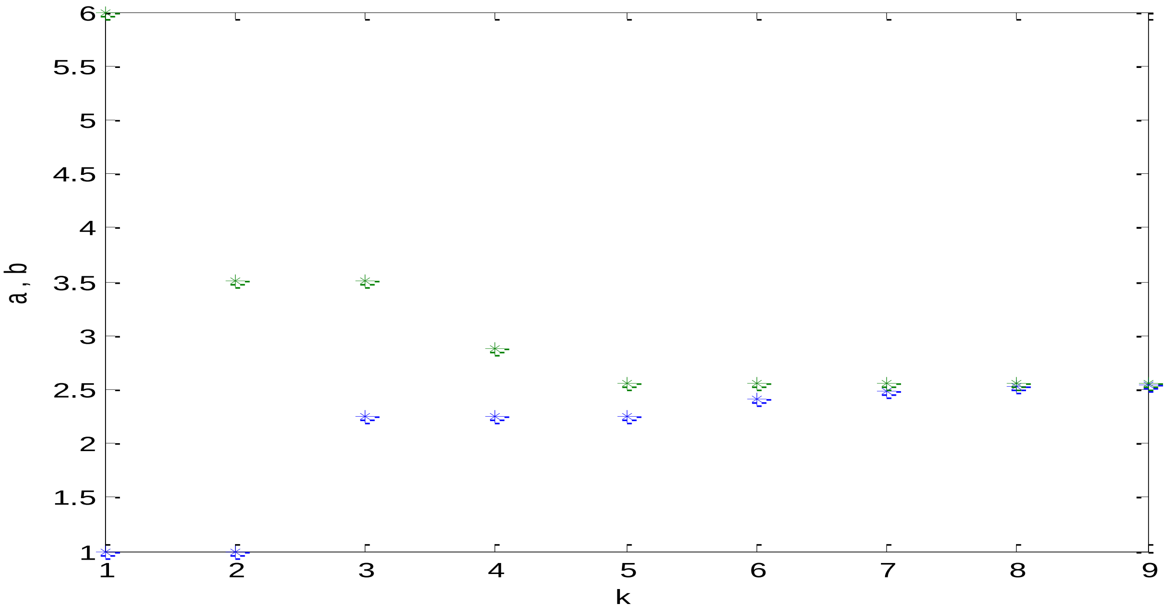
Μεταβάλλοντας τώρα το τελικό εύρος αναζήτησης l, υλοποιούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k,b_k]$ συναρτήσει του δείκτη βήματος k.

Επομένως για την f1 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:



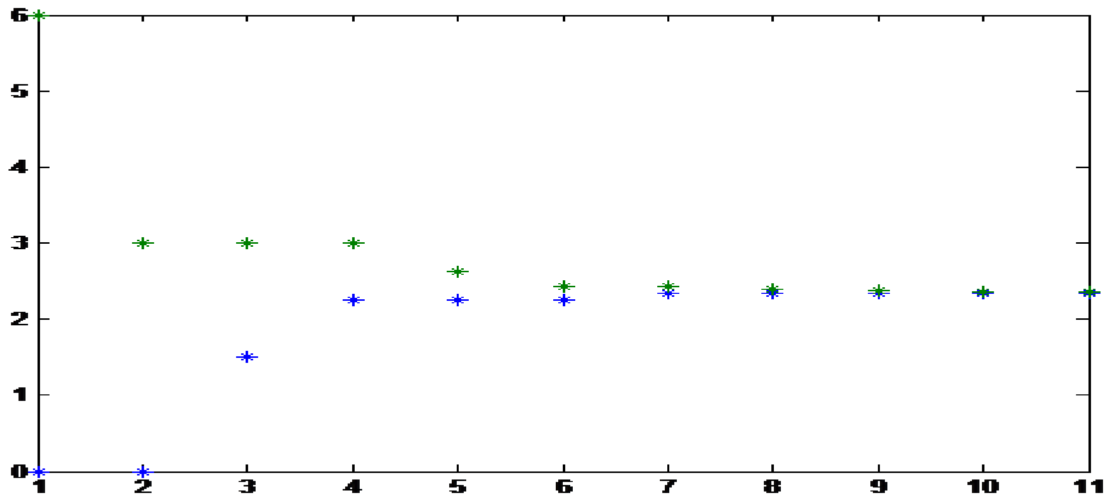


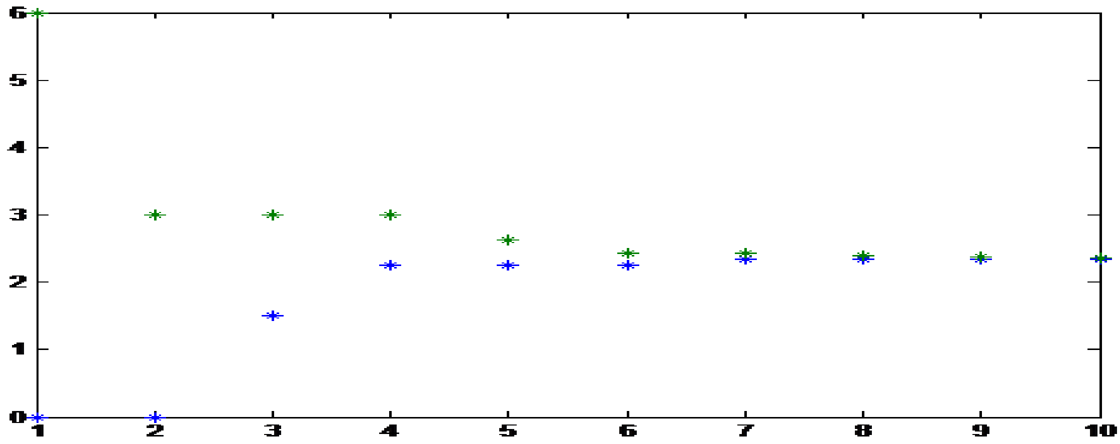


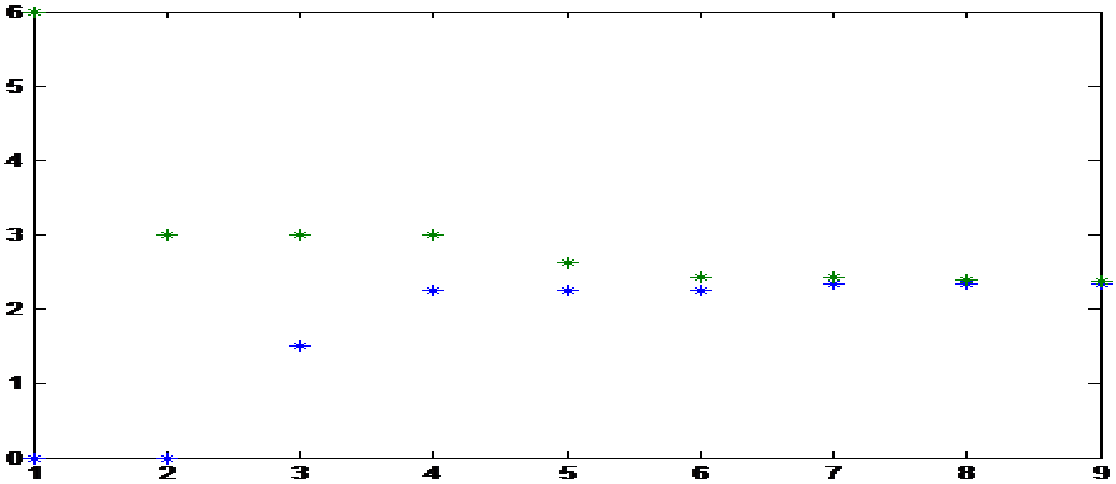


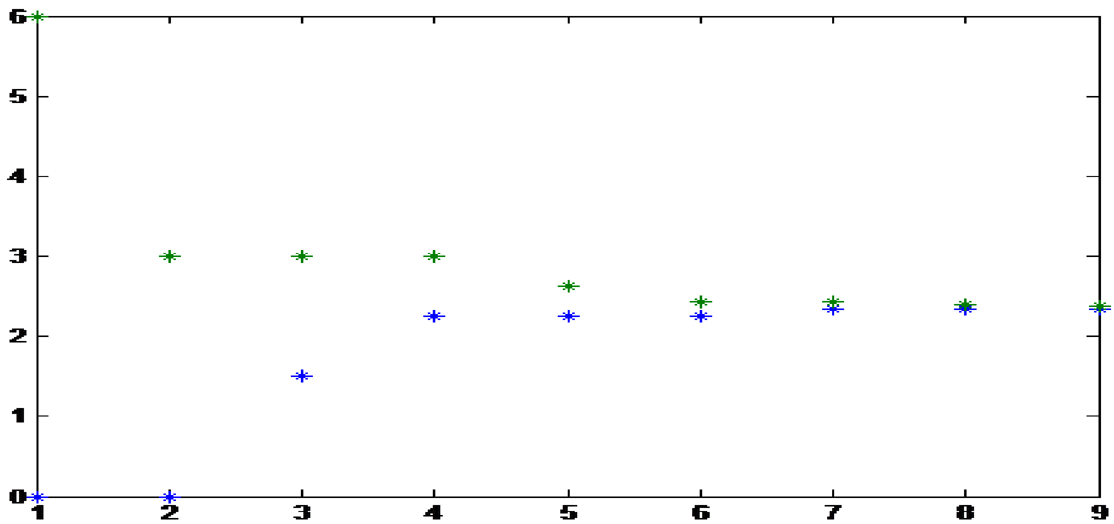
Παρατηρώντας τα διαγράμματα βλέπουμε πως και στα τέσσερα ο αλγόριθμος συγκλίνει στο τελικό διάστημα μετά από πεπερασμένο αριθμό βημάτων. Επίσης παρατηρούμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, μειώνεται ο αριθμός k των επαναλήψεων, διότι μειώνοντας την ακρίβεια ο αλγόριθμος υπολογίζει σε μικρότερο αριθμό βημάτων το τελικό διάστημα.

Για την f2 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:



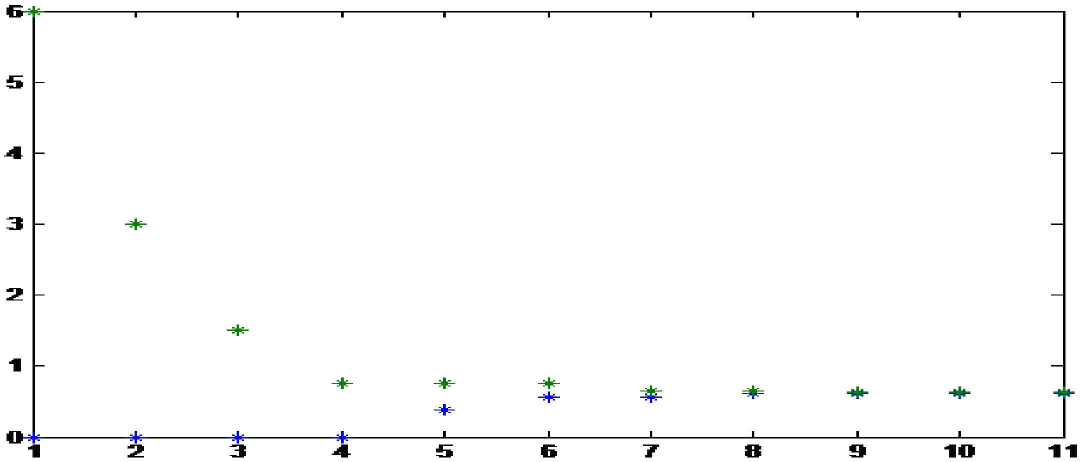

0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9
10

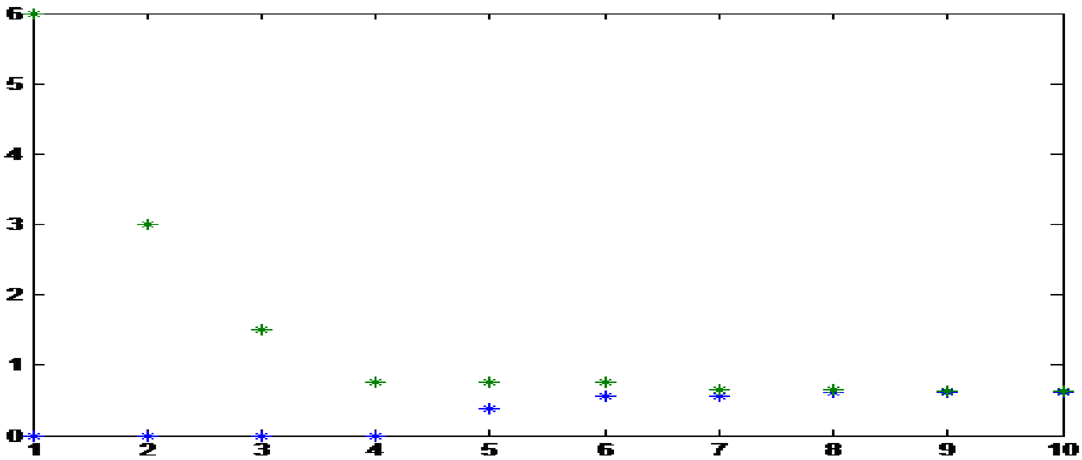

0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9

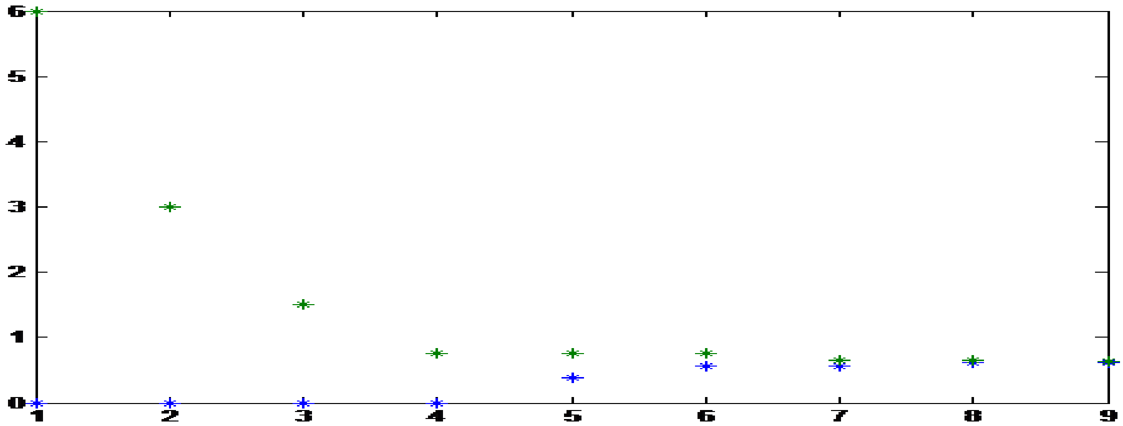

0
1
2
3
4
5
6
1
2
3
4
5
6
7
8
9

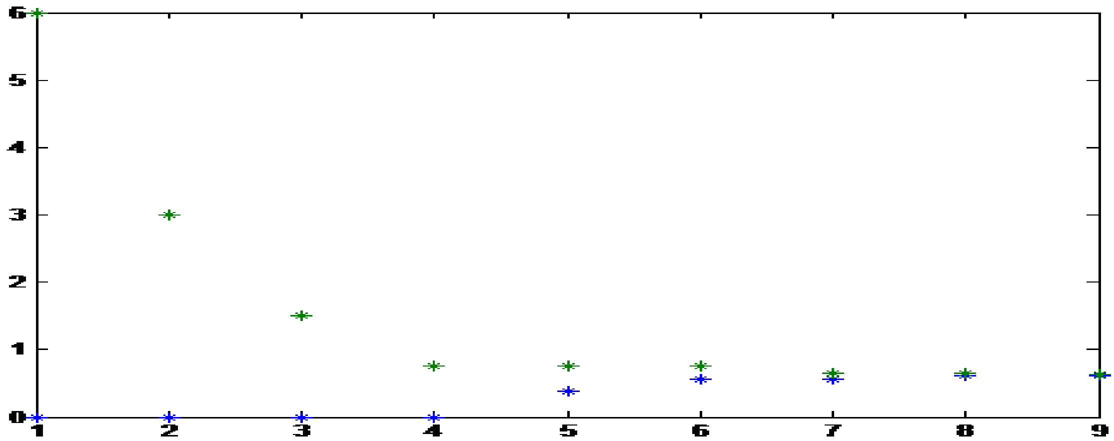
Παρατηρώντας και τα τέσσερα διαγράμματα, καταλήγουμε στα ίδια συμπεράσματα με την f1, δηλαδή ότι ο αλγόριθμος συγκλίνει έπειτα από πεπερασμένο αριθμό βημάτων και ότι αυξάνοντας το l μειώνεται ο αριθμός βημάτων k.

Για την f3 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:









Για την f3 προκύπτουν τα ίδια συμπεράσματα όπως και για τις f1,f2.

Παρατηρώντας στην πράξη τη μέθοδο της διχοτόμου, βλέπουμε πως έπειτα από ένα σχετικά μικρό αριθμό βημάτων συγκλίνει στο τελικό διάστημα, με τη δεδομένη ακρίβεια που έχουμε επιλέξει. Ο αλγόριθμος όμως απαιτεί έναν σχετικά μεγάλο αριθμό υπολογισμών της αντικειμενικής συνάρτησης, με συνέπεια να αυξάνεται αρκετά ο χρόνος υπολογισμού του τελικού διαστήματος.

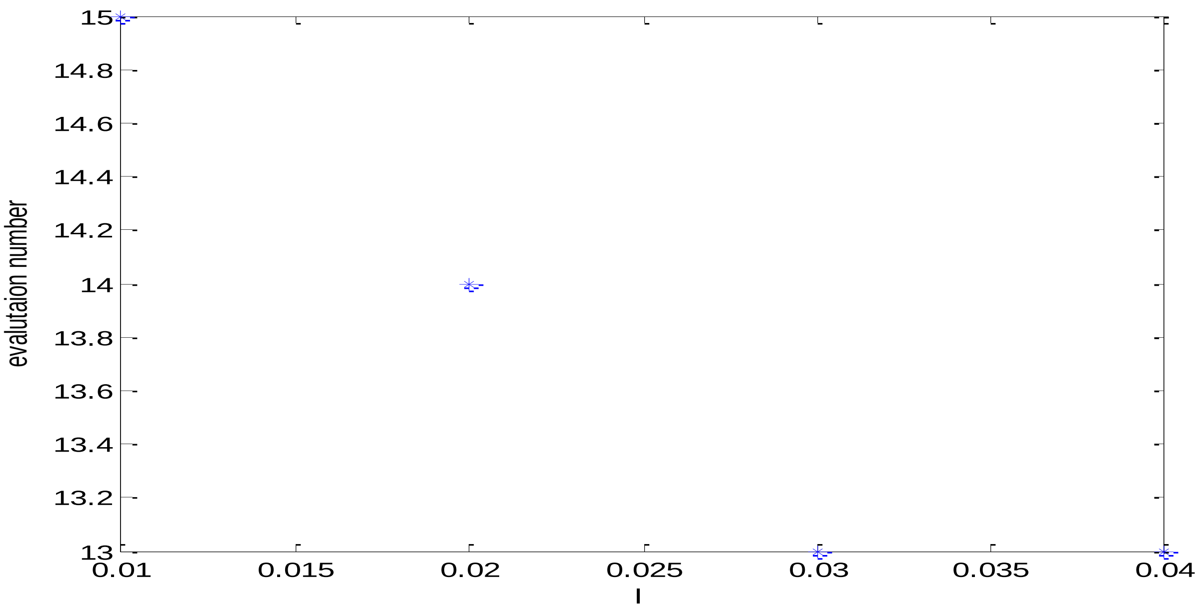
Θέμα 2.

Με τη βοήθεια του αλγορίθμου 5.1.2 του βιβλίου, υλοποιήσαμε τον αλγόριθμο του χρυσού τομέα στο Matlab. Ο αλγόριθμος ονομάζεται alg_xrusou_tomea(f,l,a,b) και έχει ως ορίσματα τη συνάρτηση f για την οποία ψάχνουμε το ελάχιστο, το τελικό εύρος του διαστήματος αναζήτησης l και τα αρχικά όρια του διαστήματος αναζήτησης a,b.

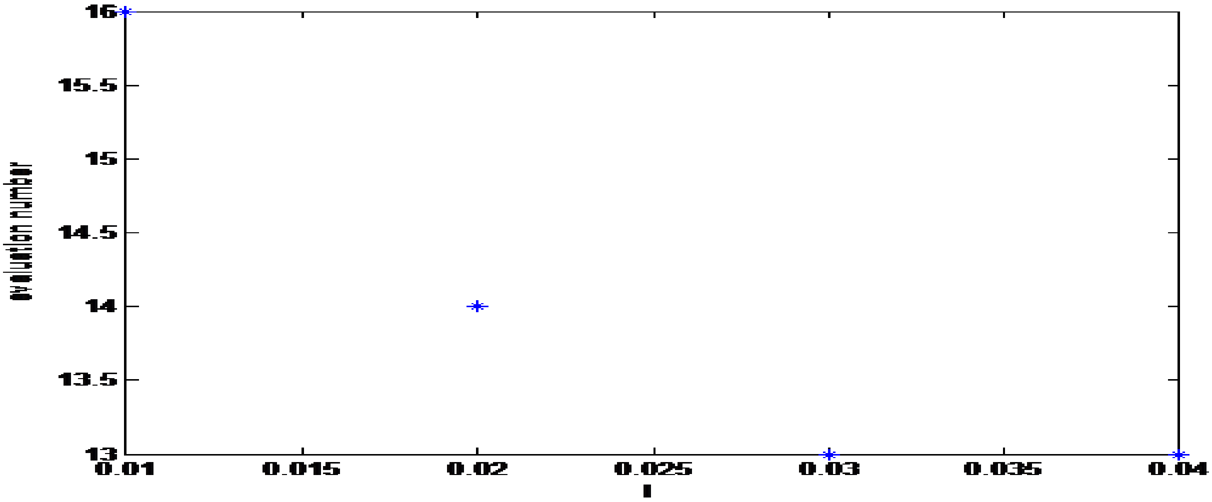
Εφαρμόζοντας τον αλγόριθμο για τη συνάρτηση f1, με l=0.01, καταλήγουμε σε ένα τελικό διάστημα [2.5463 2.5559], το οποίο βλέπουμε πως είναι αποδεκτό, αφού το τελικό εύρος του είναι μικρότερο από l, όπως προβλέπει και ο αλγόριθμος. Εφαρμόζοντάς τον και για τις f2 και f3 καταλήγουμε στα διαστήματα [2.3478 2.3548] και [0.6200 0.6272] αντίστοιχα, τα οποία βλέπουμε πως είναι αποδεκτά και τα δύο, αφού το τελικό τους εύρος είναι μικρότερο από το l.

Μεταβάλλοντας το τελικό εύρος αναζήτησης l από 0.01 μέχρι 0.04 με βήμα 0.01, παρατηρούμε τις μεταβολές των υπολογισμών c των αντικειμενικών συναρτήσεων και για τις 3 συναρτήσεις, δημιουργώντας και τις αντίστοιχες γραφικές παραστάσεις.

Επομένως για την f1 έχουμε:



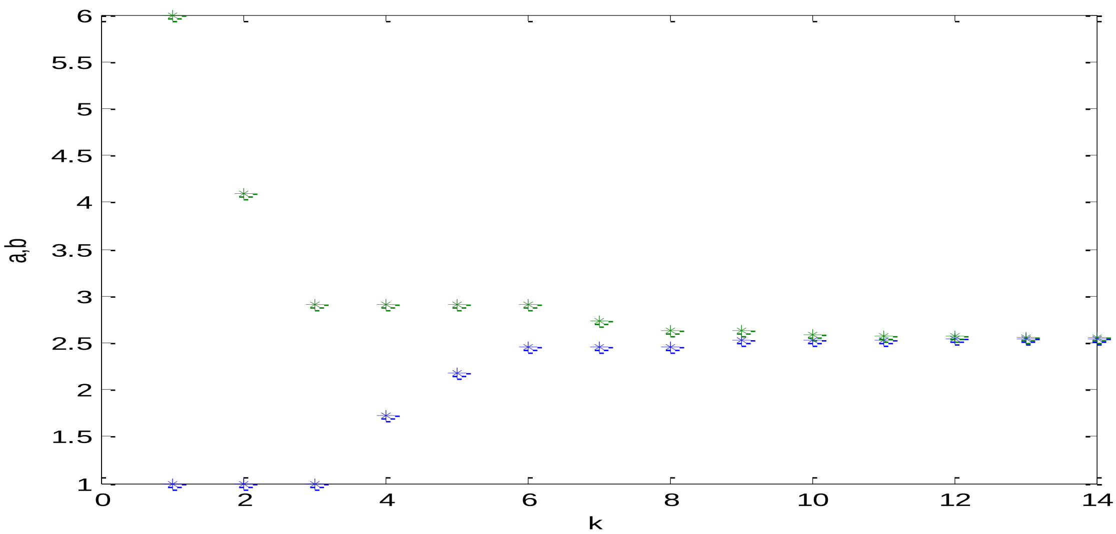
Και για τις f2,f3:

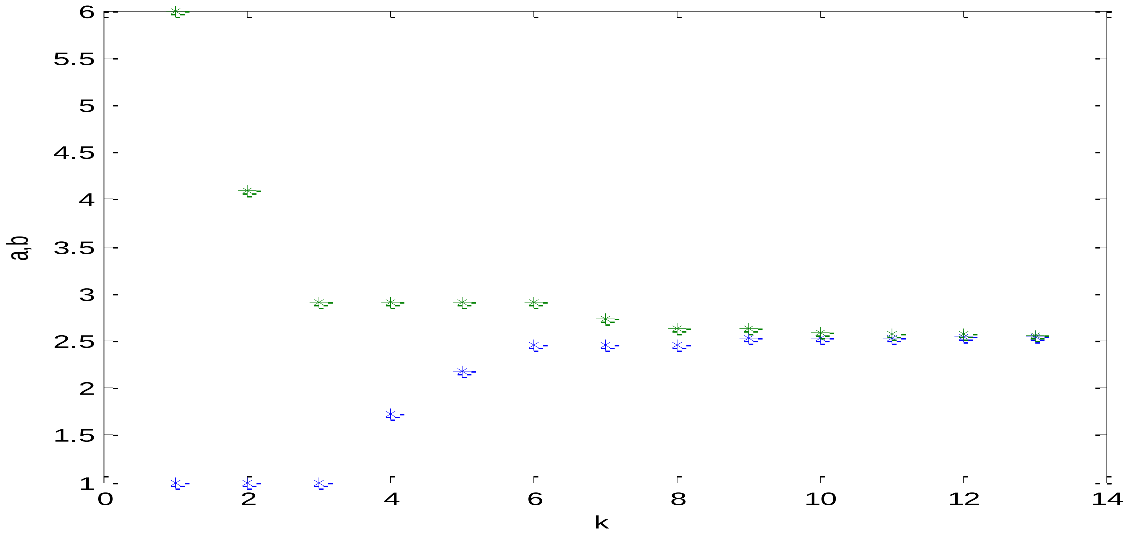


Παρατηρώντας τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, ο αριθμός c των υπολογισμών της αντικειμενικής συνάρτησης μειώνεται. Αυτό συμβαίνει διότι μειώνοντας το τελικό εύρος αναζήτησης, μειώνουμε ουσιαστικά την τελική ακρίβεια που θέλουμε να έχουμε στον υπολογισμό του ελάχιστου της συνάρτησης και έτσι μειώνεται και ο αριθμός επαναλήψεων της μεθόδου, με συνέπεια να μειωθούν και οι υπολογισμοί της αντικειμενικής συνάρτησης.

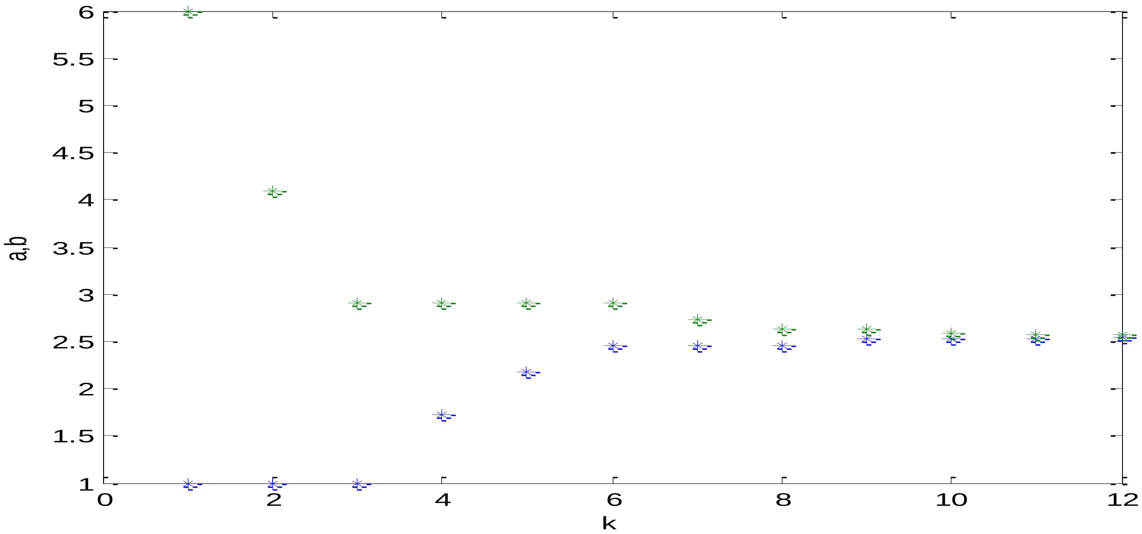
Μεταβάλλοντας τώρα το τελικό εύρος αναζήτησης l, υλοποιούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k,b_k]$ συναρτήσει του δείκτη βήματος k.

Επομένως για την f1 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:

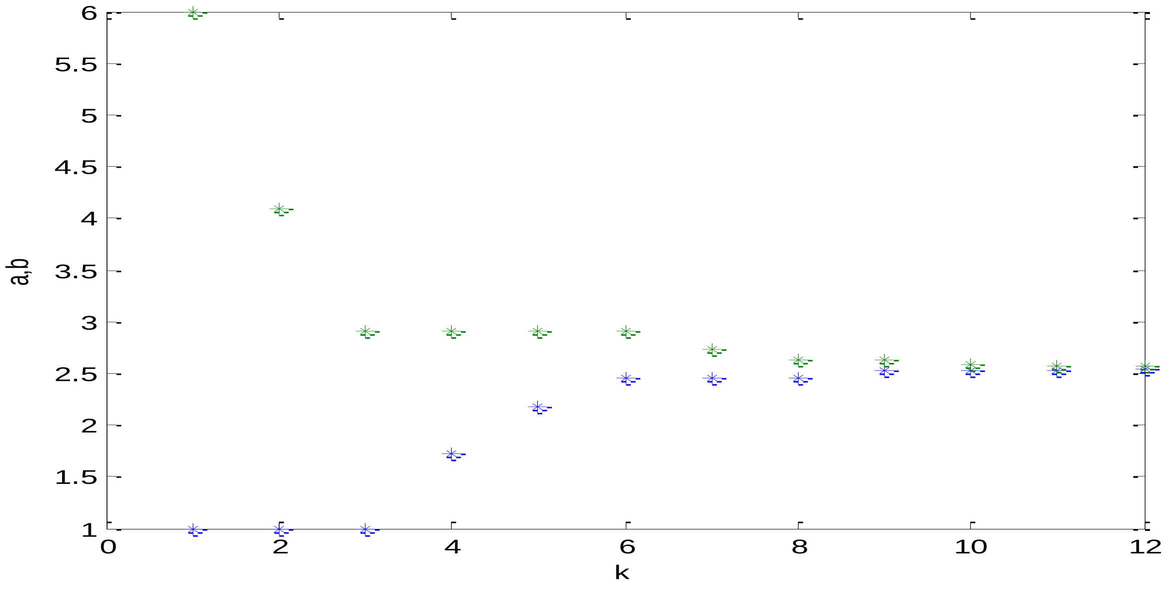




a,b
k



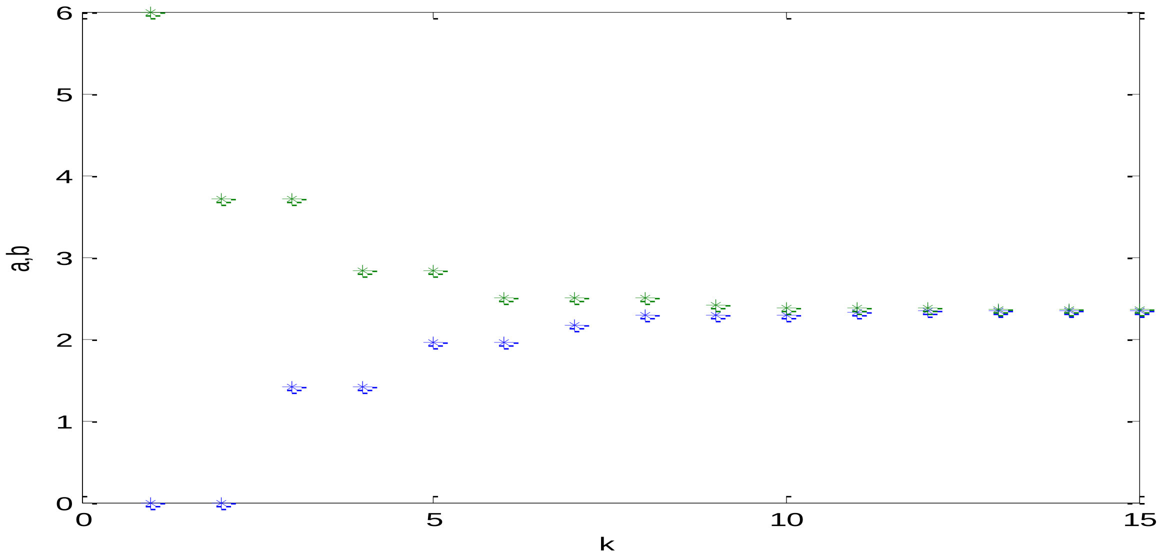
a,b
k

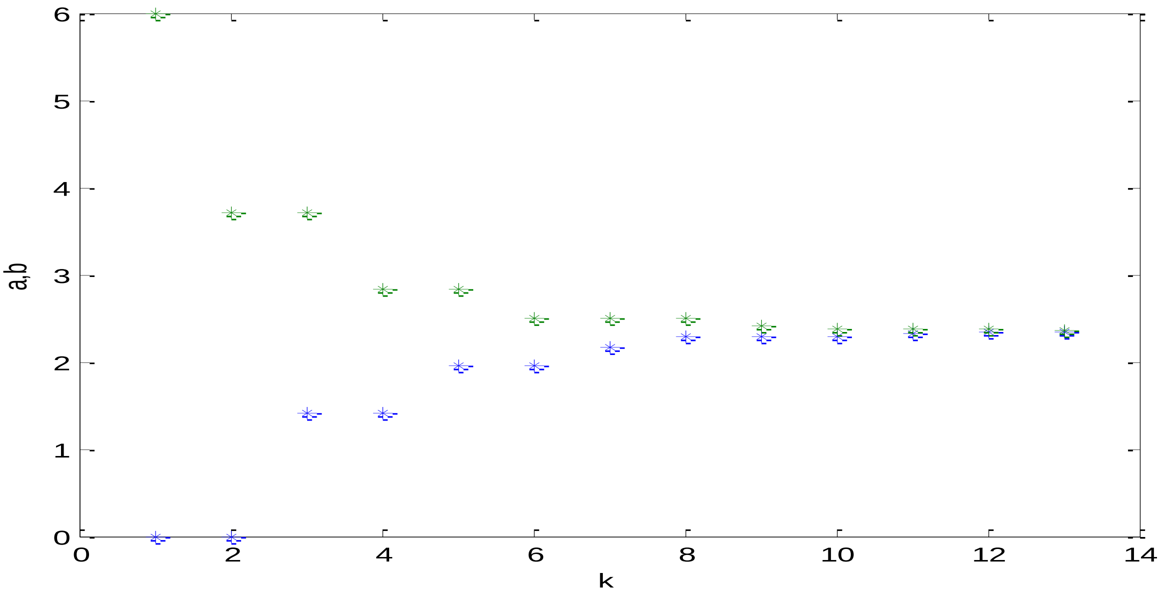


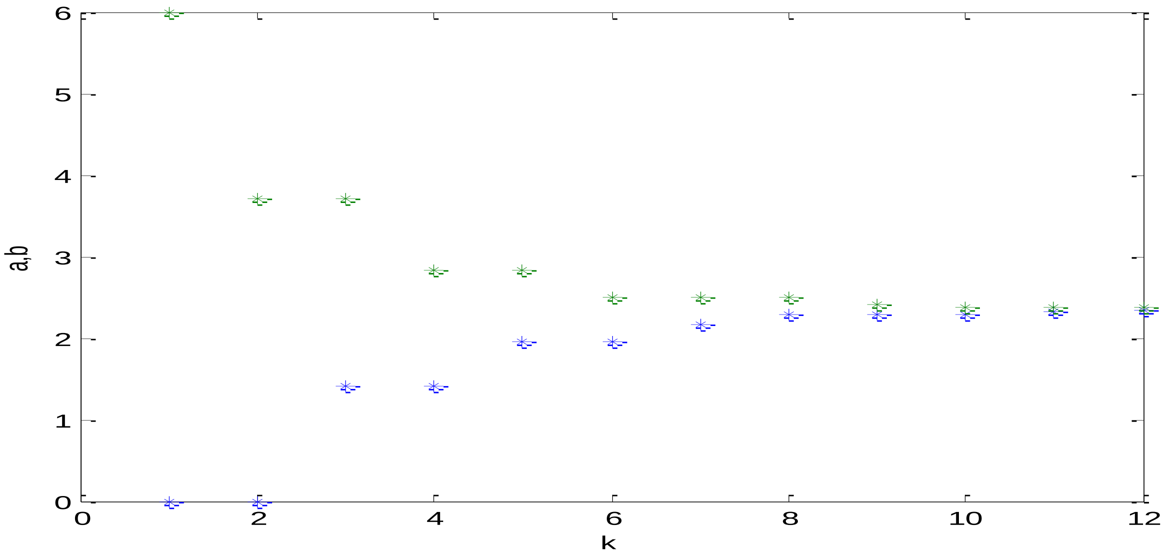
a,b
k

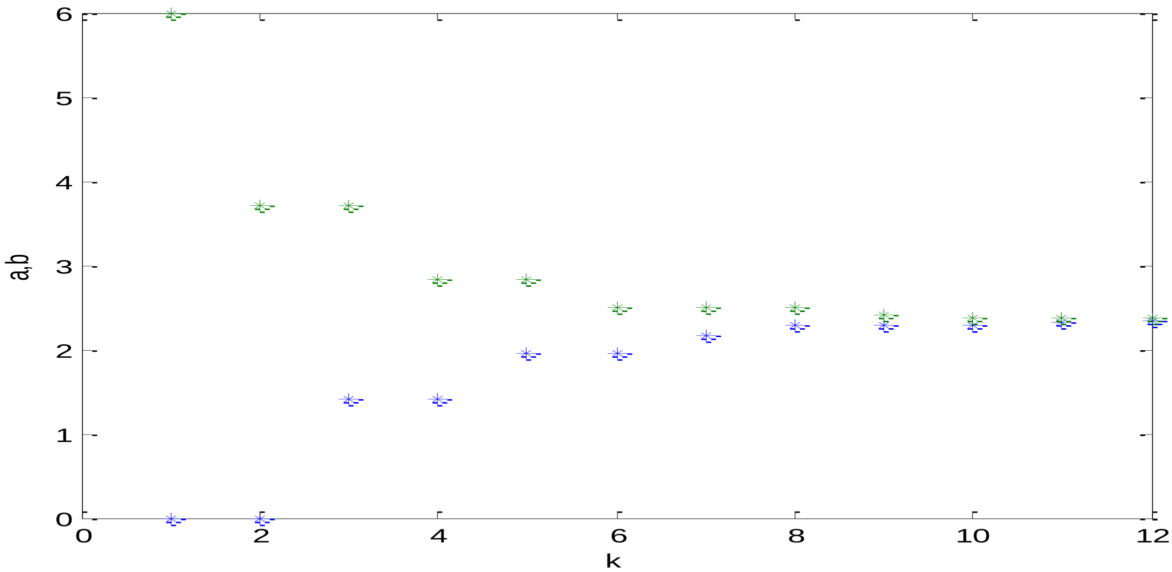
Παρατηρώντας τα διαγράμματα βλέπουμε πως και στα τέσσερα ο αλγόριθμος συγκλίνει στο τελικό διάστημα μετά από πεπερασμένο αριθμό βημάτων. Επίσης παρατηρούμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, μειώνεται ο αριθμός k των επαναλήψεων, διότι μειώνοντας την ακρίβεια ο αλγόριθμος υπολογίζει σε μικρότερο αριθμό βημάτων το τελικό διάστημα.

Για την f2 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:



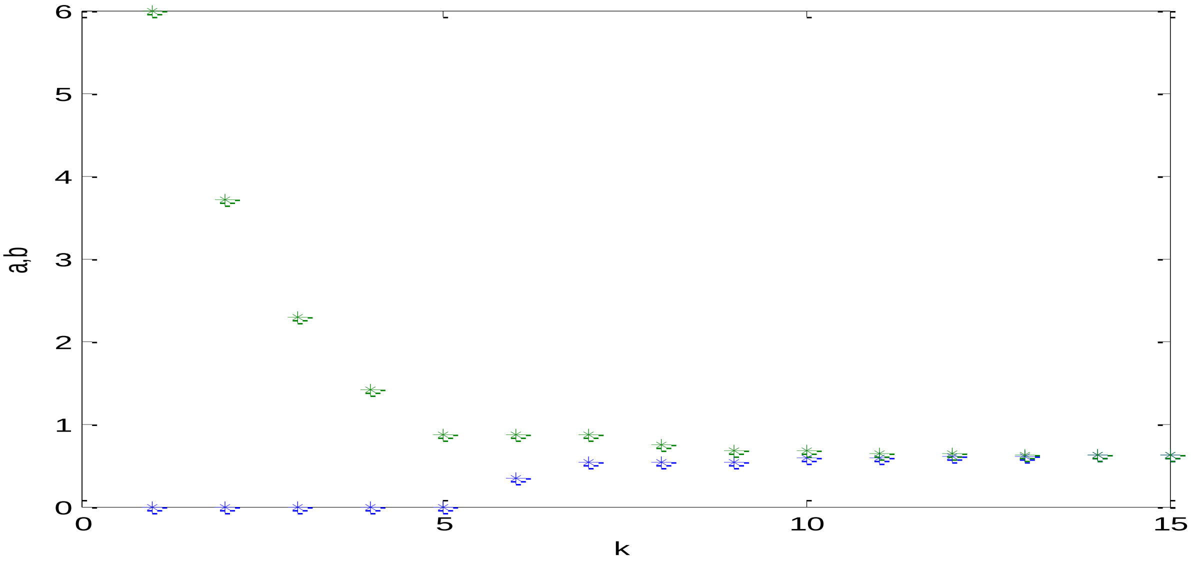


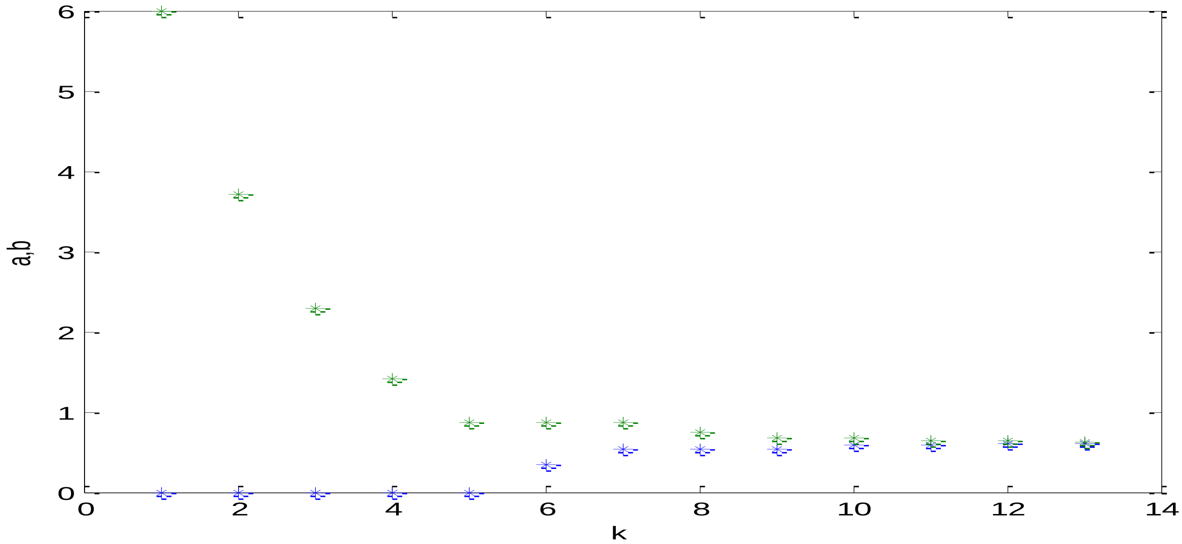


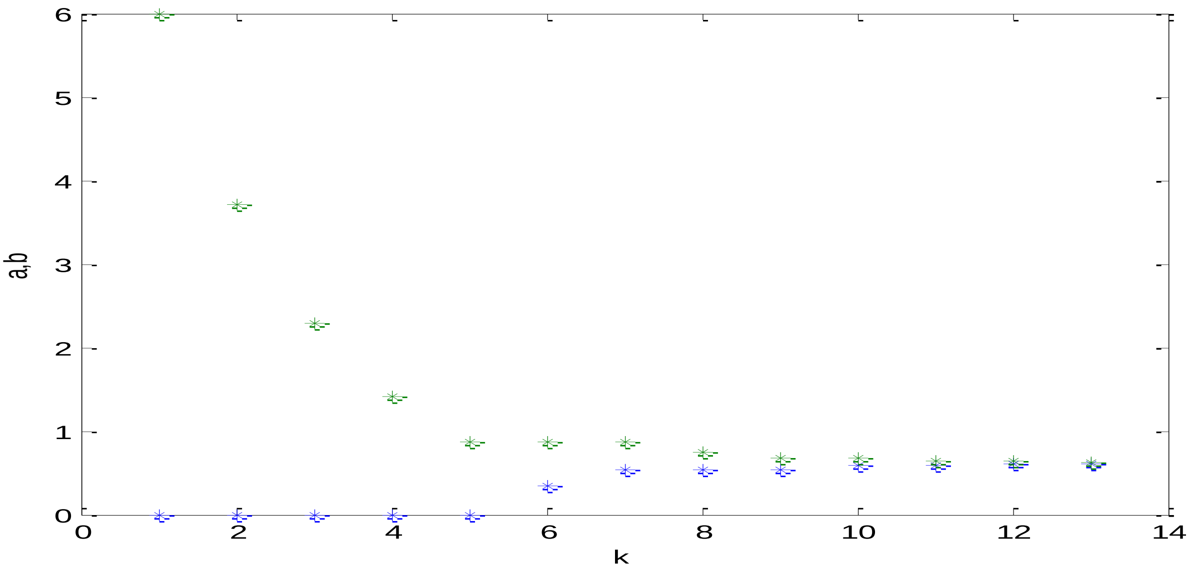


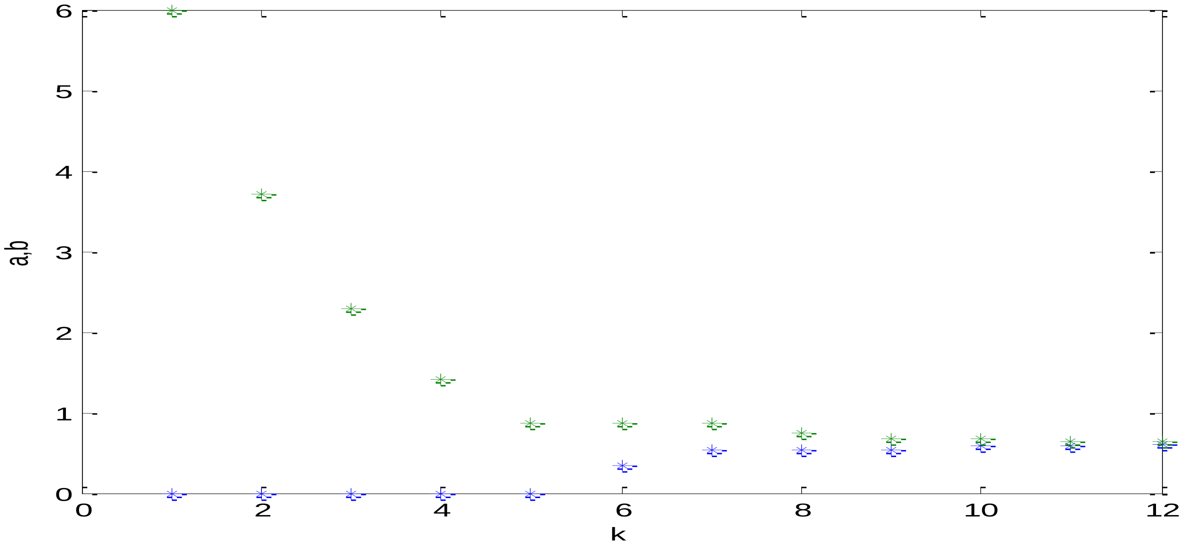
Παρατηρώντας και τα τέσσερα διαγράμματα, καταλήγουμε στα ίδια συμπεράσματα με την f1, δηλαδή ότι ο αλγόριθμος συγκλίνει έπειτα από πεπερασμένο αριθμό βημάτων και ότι αυξάνοντας το l μειώνεται ο αριθμός βημάτων k.

Για την f3 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:


a,b
k


a,b
k


a,b
k



Για την f3 προκύπτουν τα ίδια συμπεράσματα όπως και για τις f1,f2.

Για τη μέθοδο του χρυσού τομέα παρατηρούμε πως έχουμε έναν αυξημένο αριθμό βημάτων για τον υπολογισμό του τελικού διαστήματος, αλλά μικρότερο αριθμό υπολογισμών της αντικειμενικής συνάρτησης, με συνέπεια την αύξηση της ταχύτητας του αλγορίθμου.

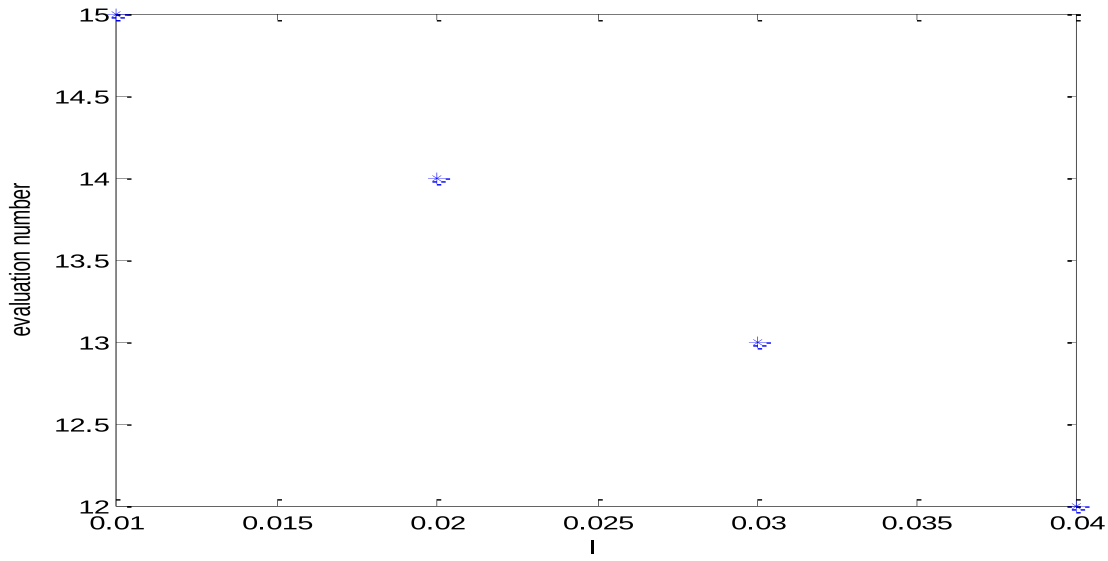
Θέμα 3.

Με τη βοήθεια της θεωρητικής περιγραφής του βιβλίου, υλοποιήσαμε τον αλγόριθμο Fibonacci στο Matlab. Ο αλγόριθμος για προκαθορισμένο αριθμό βημάτων n, υπολογίζει σε κάθε επανάληψη εάν $f(x_{1k})>f(x_{2k})$ ή εάν $f(x_{1k})<f(x_{2k})$. Στην πρώτη περίπτωση θέτει $a_{k+1}=x_{1k}$, $b_{k+1}=b_k$, $x_{1k+1}=x_{2k}$ και $x_{2k+1}=a_{k+1}+(\text{fibonacci}(n-k-1)/\text{fibonacci}(n-k))(b_{k+1}-a_{k+1})$, ενώ στην δεύτερη $a_{k+1}=a_k$, $b_{k+1}=x_{2k}$, $x_{2k+1}=x_{1k}$ και $x_{1k+1}=a_{k+1}+(\text{fibonacci}(n-k-2)/\text{fibonacci}(n-k))(b_{k+1}-a_{k+1})$. Ο αλγόριθμος τερματίζει έπειτα από πεπερασμένο αριθμό βημάτων n-2, όπου ο αριθμός n καθορίζεται από τη σχέση $\text{fibonacci}(n)>=(b_1-a_1)/l$. Το όνομα του αλγορίθμου είναι alg_fibonacci(f,l,a,b) και έχει ως ορίσματα τη συνάρτηση f για την οποία ψάχνουμε το ελάχιστο, το τελικό εύρος του διαστήματος αναζήτησης l και τα αρχικά όρια του διαστήματος αναζήτησης a,b.

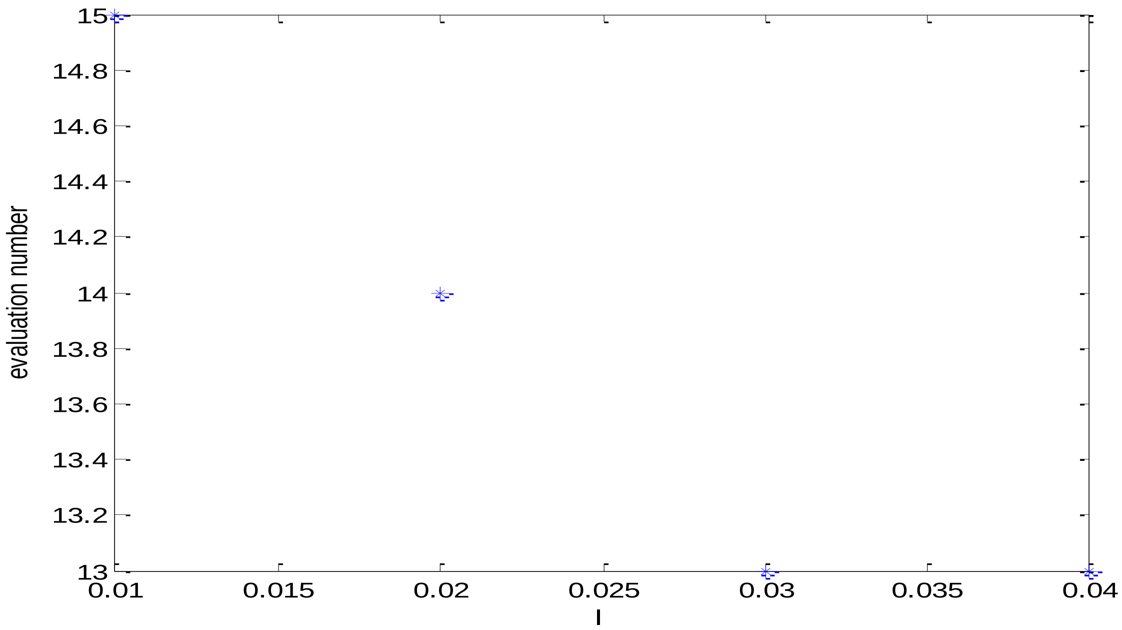
Εφαρμόζοντας τον αλγόριθμο για τη συνάρτηση f1, με l=0.01, καταλήγουμε σε ένα τελικό διάστημα [2.5410 2.5492], το οποίο βλέπουμε πως είναι αποδεκτό, αφού το τελικό εύρος του είναι μικρότερο από l, όπως προβλέπει και ο αλγόριθμος. Εφαρμόζοντάς τον και για τις f2 και f3 καταλήγουμε στα διαστήματα [2.3410 2.3508] και [0.6098 0.6197] αντίστοιχα, τα οποία βλέπουμε πως είναι αποδεκτά και τα δύο, αφού το τελικό τους εύρος είναι μικρότερο από το l.

Μεταβάλλοντας το τελικό εύρος αναζήτησης l από 0.01 μέχρι 0.04 με βήμα 0.01, παρατηρούμε τις μεταβολές των υπολογισμών c των αντικειμενικών συναρτήσεων και για τις 3 συναρτήσεις, δημιουργώντας και τις αντίστοιχες γραφικές παραστάσεις.

Επομένως για την f1 έχουμε:



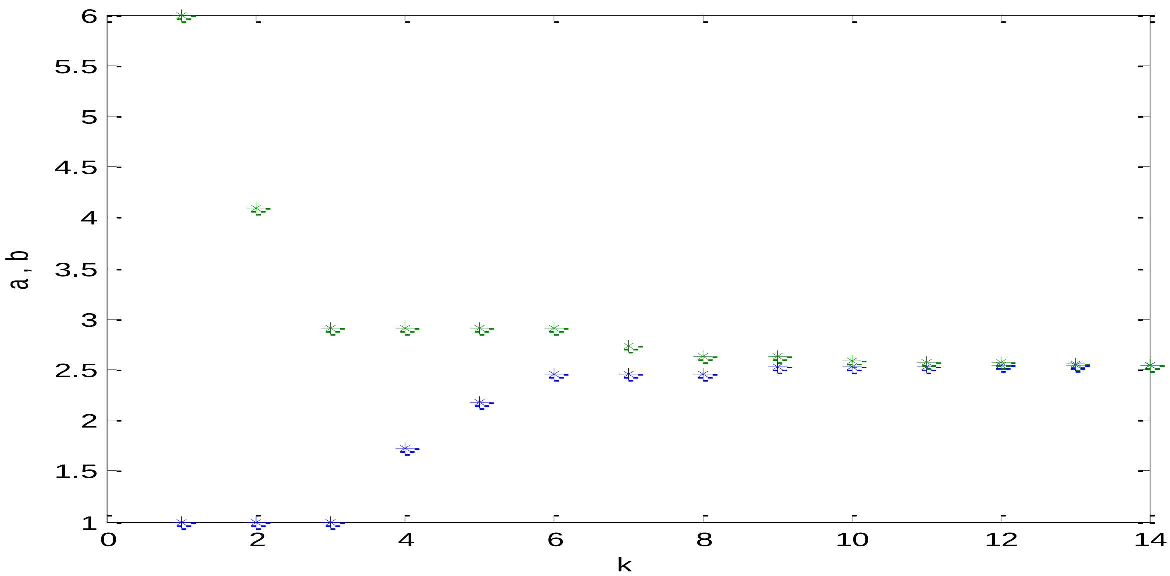
Και για τις f2,f3:

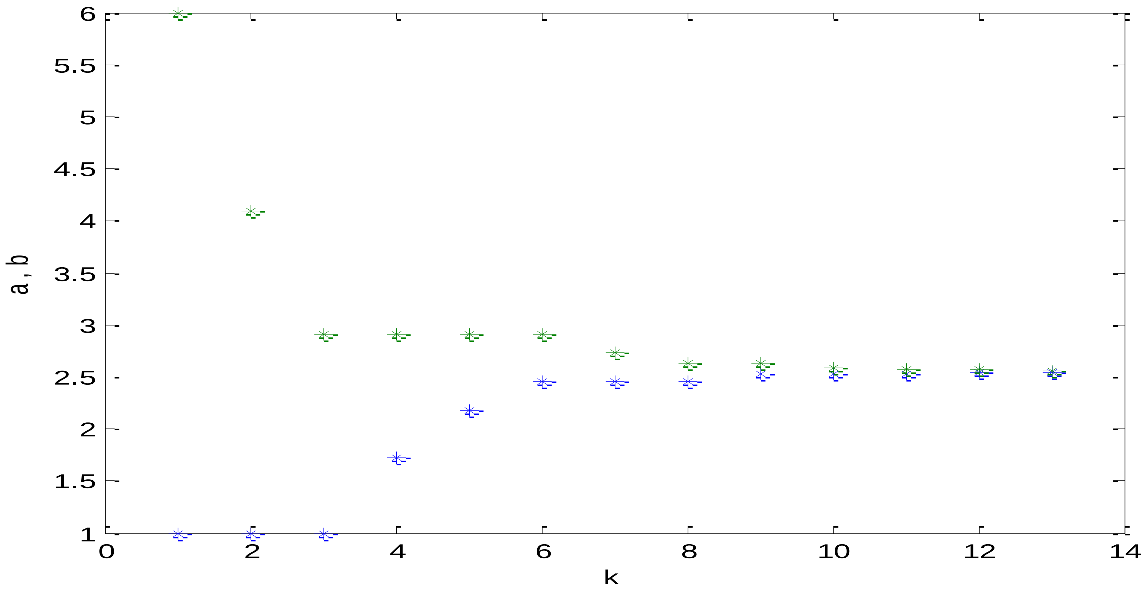


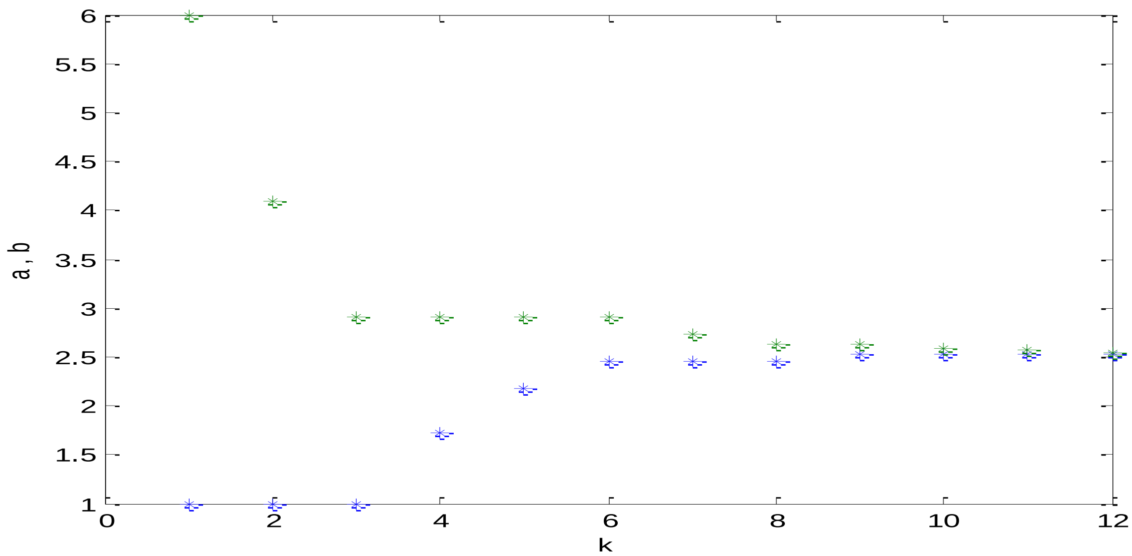
Παρατηρώντας τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, ο αριθμός c των υπολογισμών της αντικειμενικής συνάρτησης μειώνεται. Αυτό συμβαίνει διότι μειώνοντας το τελικό εύρος αναζήτησης, μειώνουμε ουσιαστικά την τελική ακρίβεια που θέλουμε να έχουμε στον υπολογισμό του ελάχιστου της συνάρτησης και έτσι μειώνεται και ο αριθμός επαναλήψεων της μεθόδου, με συνέπεια να μειωθούν και οι υπολογισμοί της αντικειμενικής συνάρτησης.

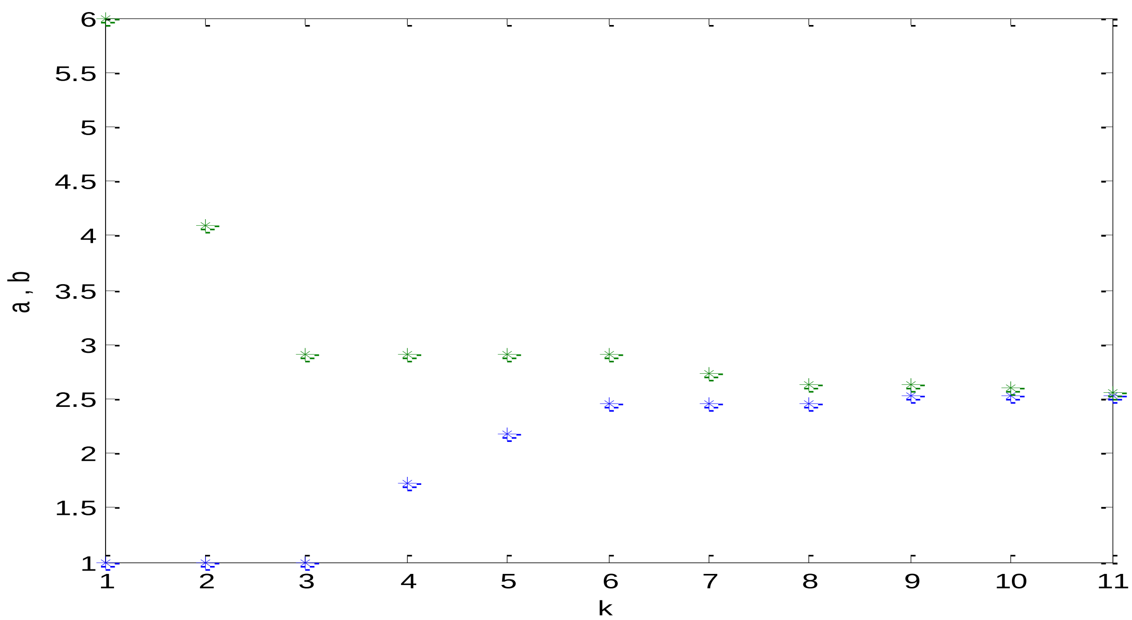
Μεταβάλλοντας τώρα το τελικό εύρος αναζήτησης l, υλοποιούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k,b_k]$ συναρτήσει του δείκτη βήματος k.

Επομένως για την f1 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:



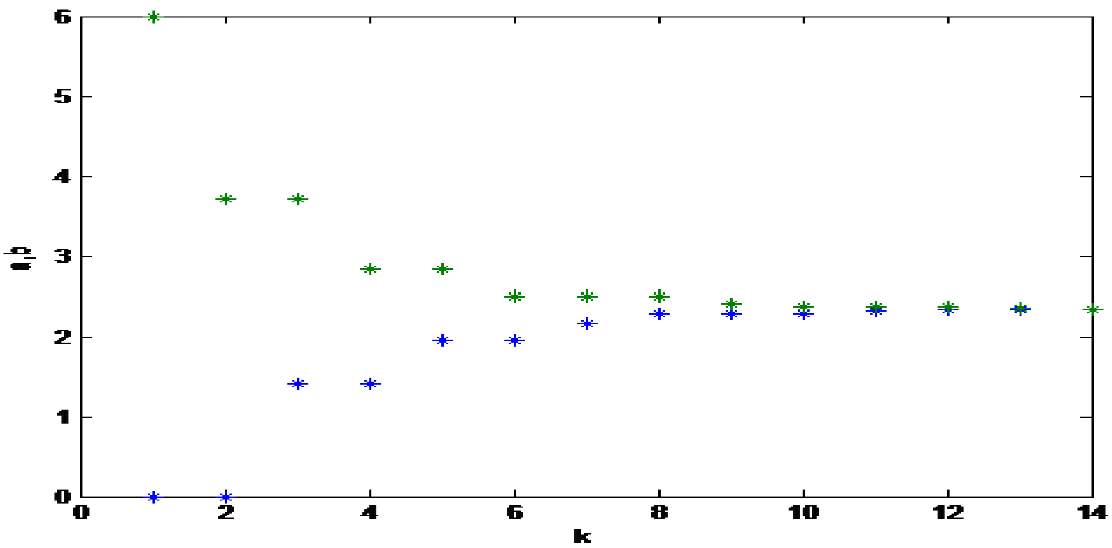

a , b
k

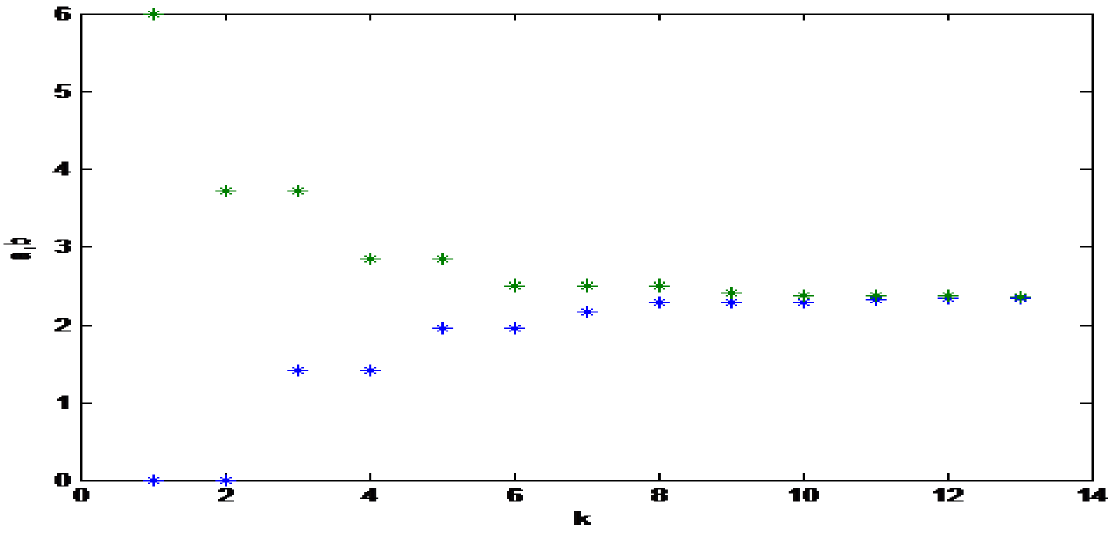

a , b
k

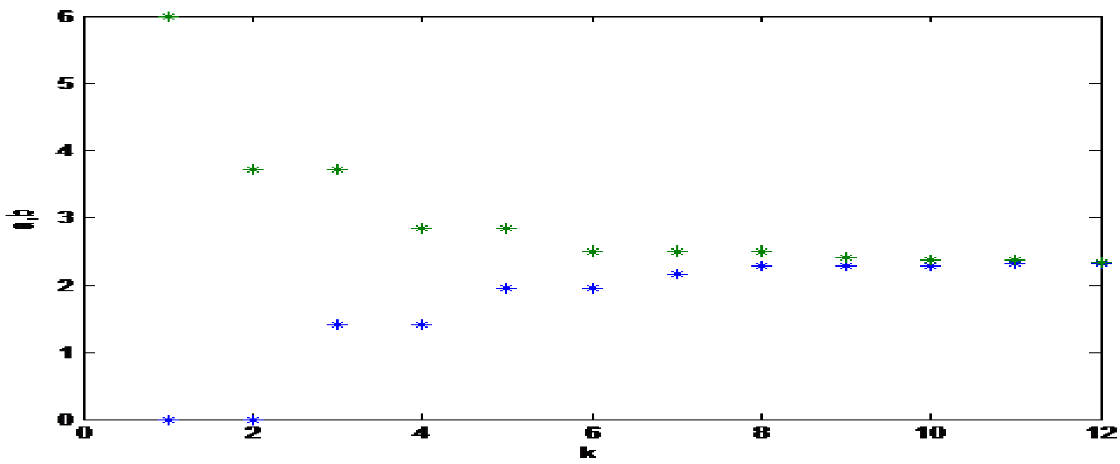

a , b
k

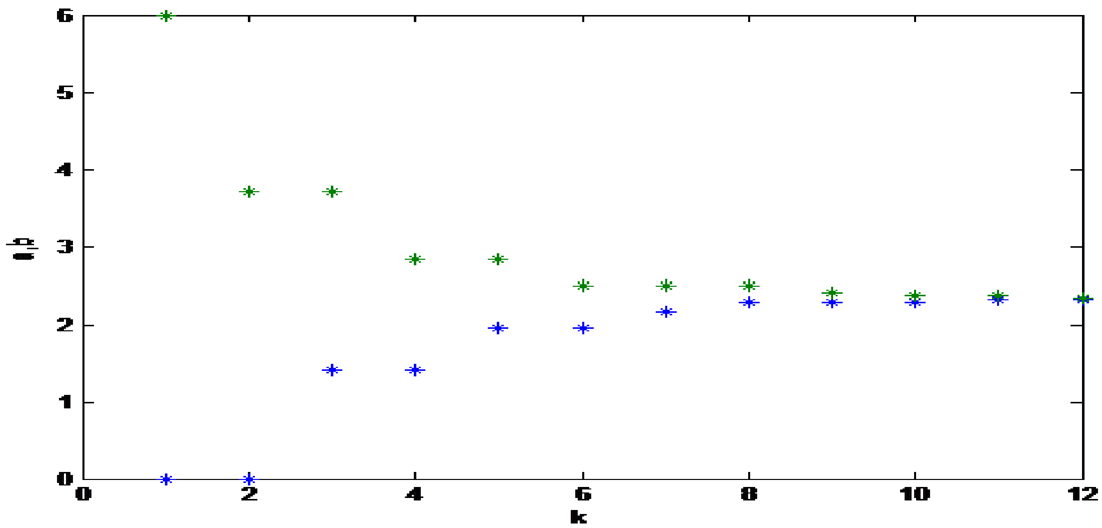
Παρατηρώντας τα διαγράμματα βλέπουμε πως και στα τέσσερα ο αλγόριθμος συγκλίνει στο τελικό διάστημα μετά από πεπερασμένο αριθμό βημάτων. Επίσης παρατηρούμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, μειώνεται ο αριθμός k των επαναλήψεων, διότι μειώνοντας την ακρίβεια ο αλγόριθμος υπολογίζει σε μικρότερο αριθμό βημάτων το τελικό διάστημα.

Για την f2 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:



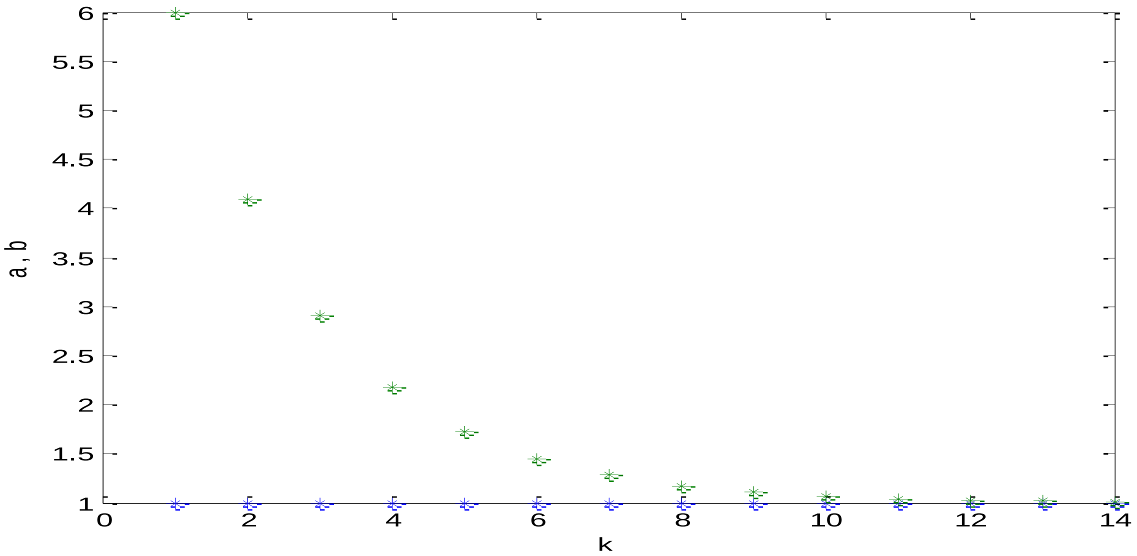


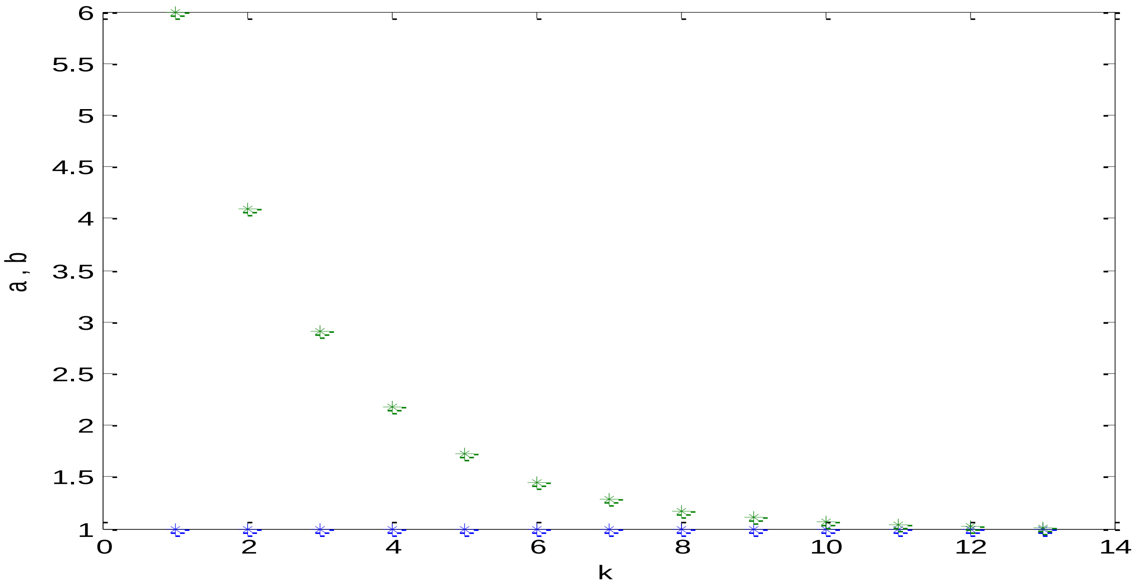


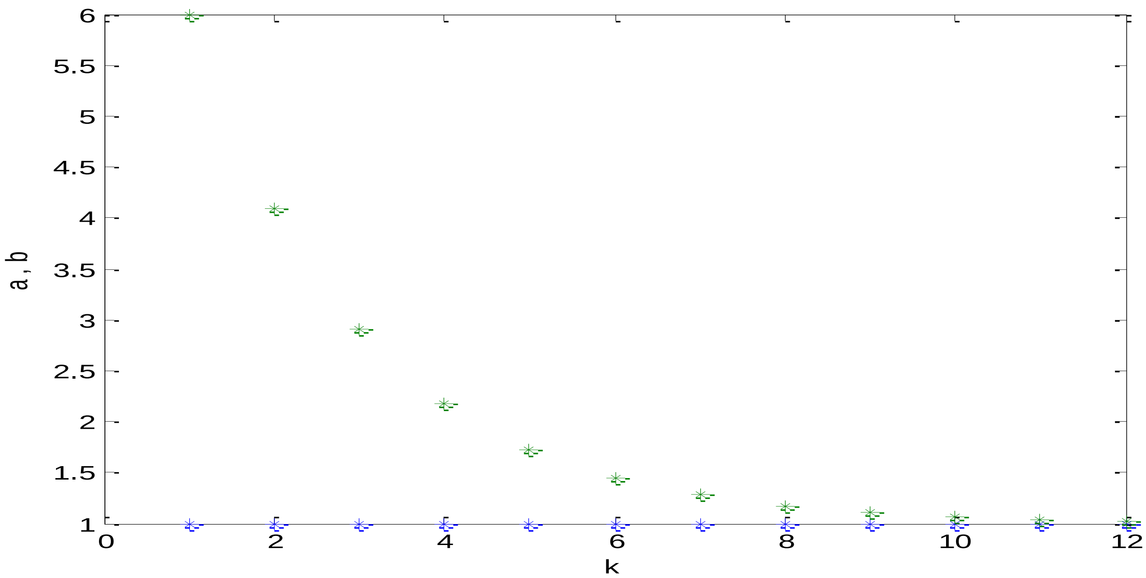


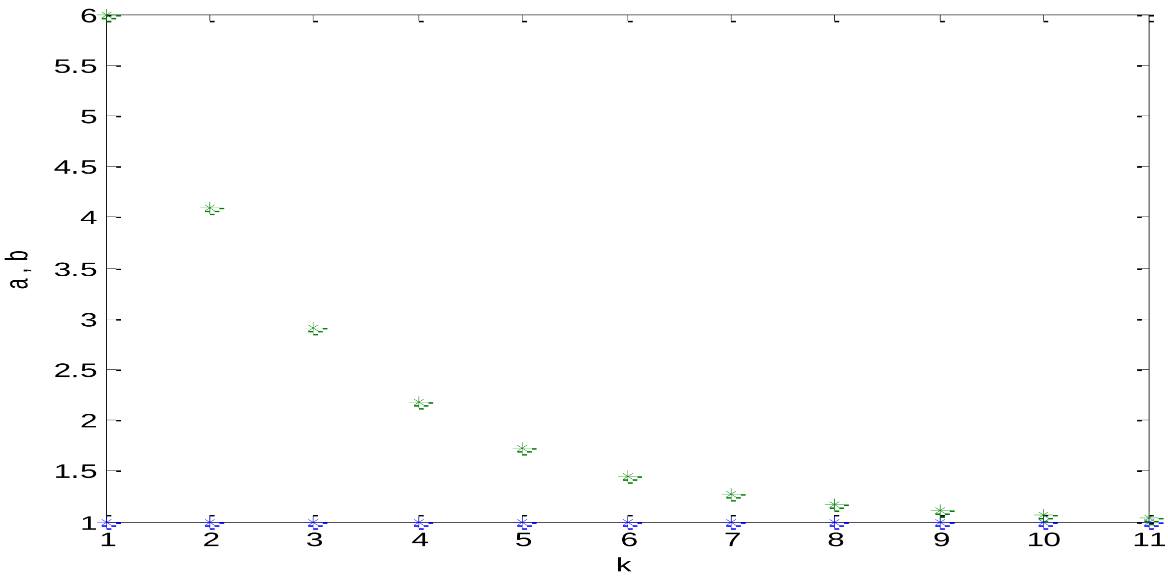
Παρατηρώντας και τα τέσσερα διαγράμματα, καταλήγουμε στα ίδια συμπεράσματα με την f1, δηλαδή ότι ο αλγόριθμος συγκλίνει έπειτα από πεπερασμένο αριθμό βημάτων και ότι αυξάνοντας το l μειώνεται ο αριθμός βημάτων k.

Για την f3 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:


a, b
k
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12
14


a, b
k
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12
14


a, b
k
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12



Για την f3 προκύπτουν τα ίδια συμπεράσματα όπως και για τις f1,f2.

Στην πράξη βλέπουμε πως η μέθοδος Fibonacci έχει περίπου ίδια απόδοση με τη μέθοδο του χρυσού τομέα, έχοντας παρόμοιο αριθμό υπολογισμών την αντικειμενικής συνάρτησης και παρόμοιο αριθμό βημάτων k.

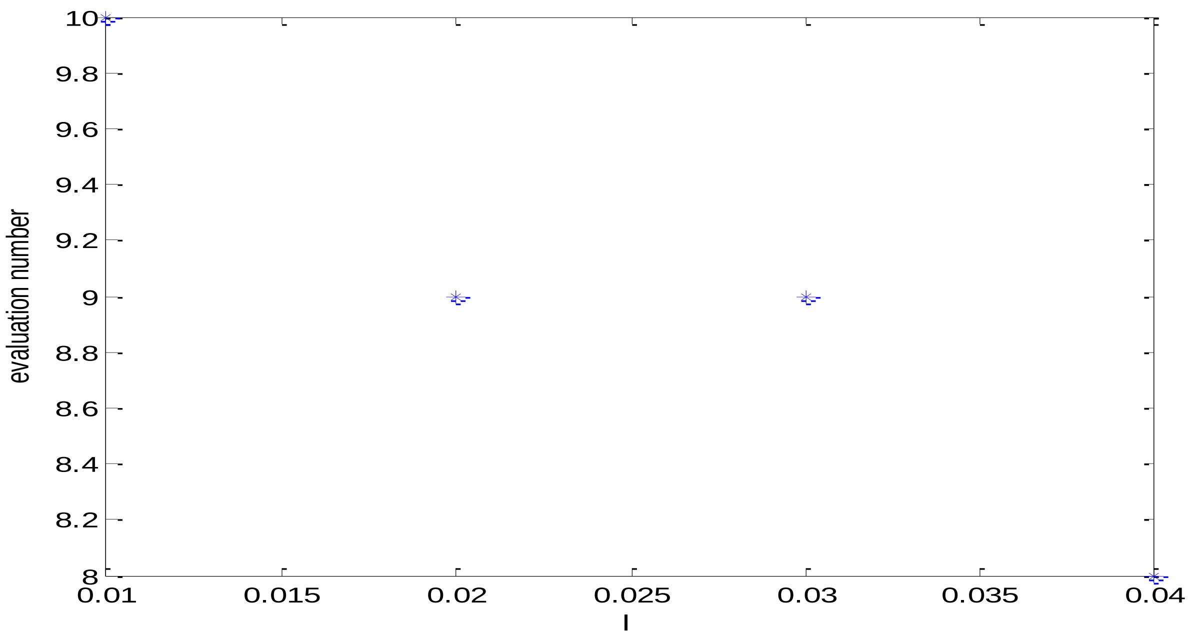
Θέμα 4.

Με τη βοήθεια του αλγορίθμου 5.1.4 του βιβλίου, υλοποιήσαμε τον αλγόριθμο της διχοτόμου με χρήση παραγώγων στο Matlab. Ο αλγόριθμος ονομάζεται alg_dixotomou_me_paragwgo(f,l,a,b) και έχει ως ορίσματα τη συνάρτηση f για την οποία ψάχνουμε το ελάχιστο, το τελικό εύρος του διαστήματος αναζήτησης l και τα αρχικά όρια του διαστήματος αναζήτησης a,b.

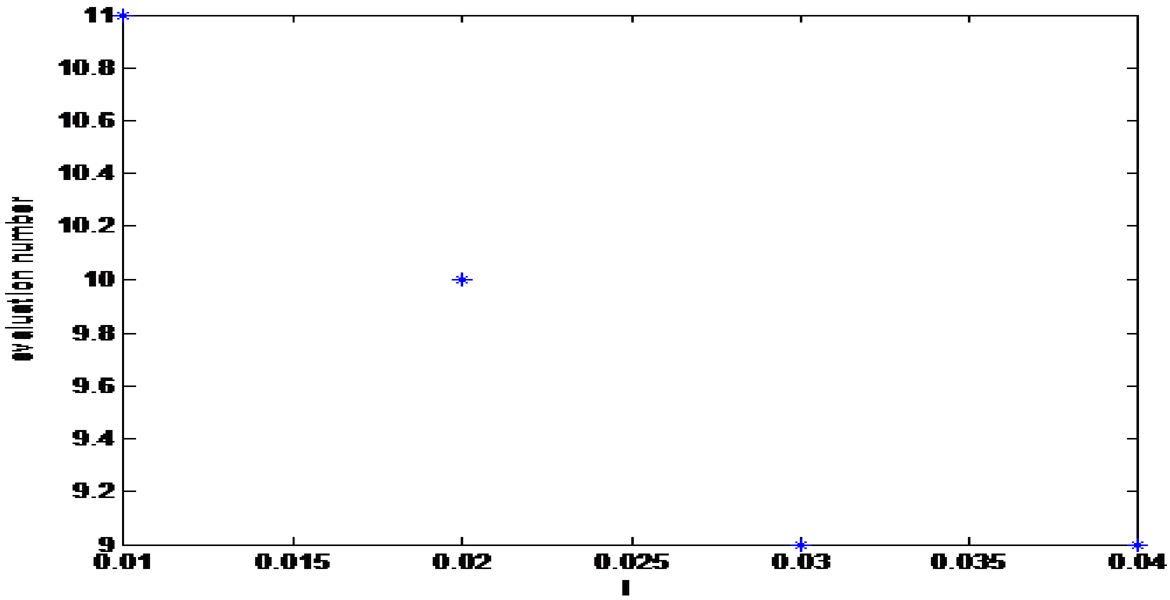
Εφαρμόζοντας τον αλγόριθμο για τη συνάρτηση f1, με l=0.01, καταλήγουμε σε ένα τελικό διάστημα [2.5479 2.5527], το οποίο βλέπουμε πως είναι αποδεκτό, αφού το τελικό εύρος του είναι μικρότερο από l, όπως προβλέπει και ο αλγόριθμος. Εφαρμόζοντάς τον και για τις f2 και f3 καταλήγουμε στα διαστήματα [2.3496 2.3525] και [0.6211 0.6240] αντίστοιχα, τα οποία βλέπουμε πως είναι αποδεκτά και τα δύο, αφού το τελικό τους εύρος είναι μικρότερο από το l.

Μεταβάλλοντας το τελικό εύρος αναζήτησης l από 0.01 μέχρι 0.04 με βήμα 0.01, παρατηρούμε τις μεταβολές των υπολογισμών c των αντικειμενικών συναρτήσεων και για τις 3 συναρτήσεις, δημιουργώντας και τις αντίστοιχες γραφικές παραστάσεις.

Επομένως για την f1 έχουμε:



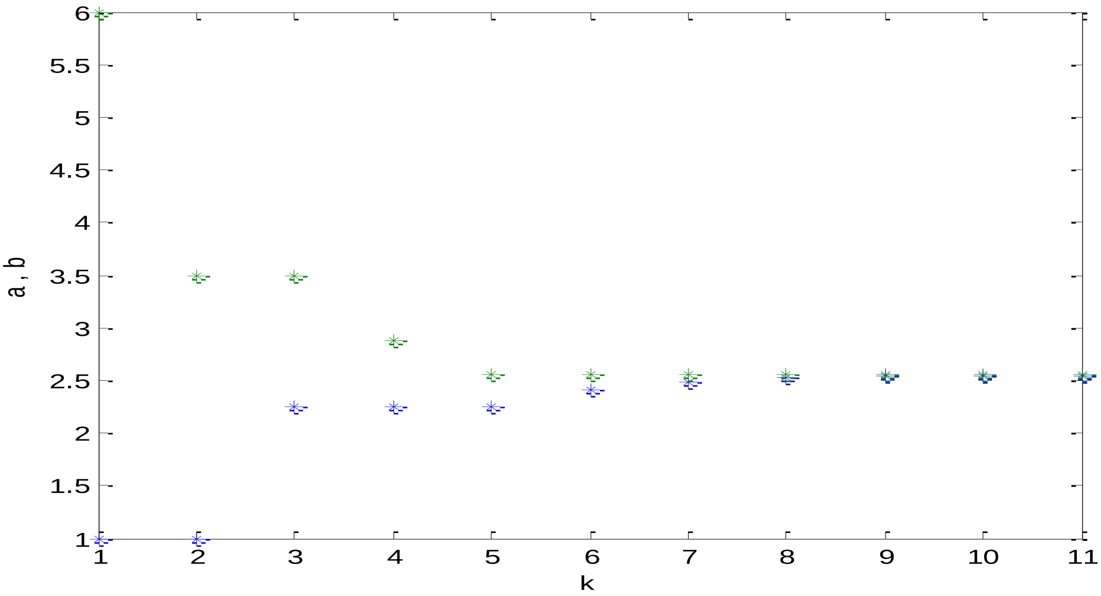
Και για τις f2,f3:

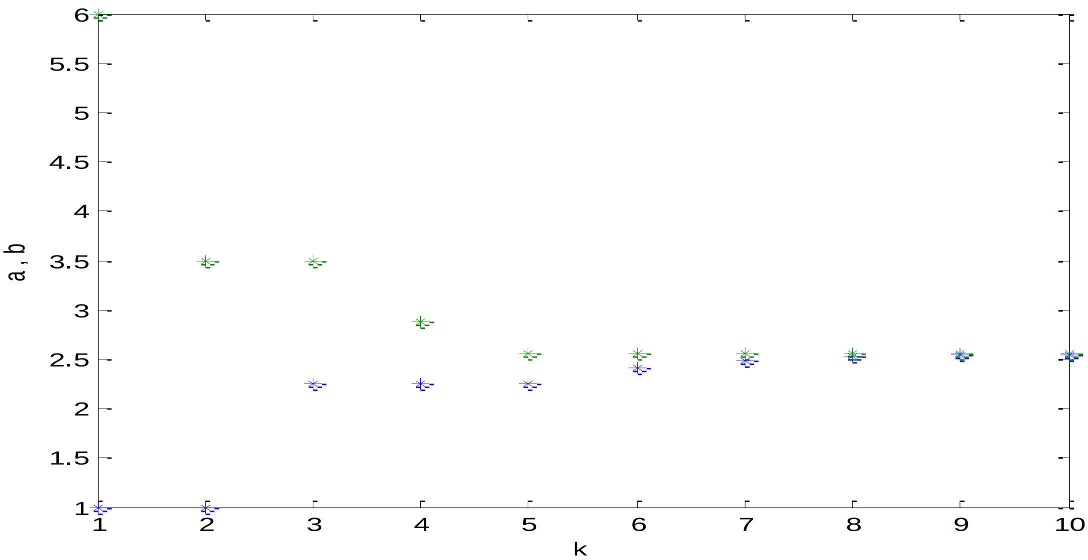


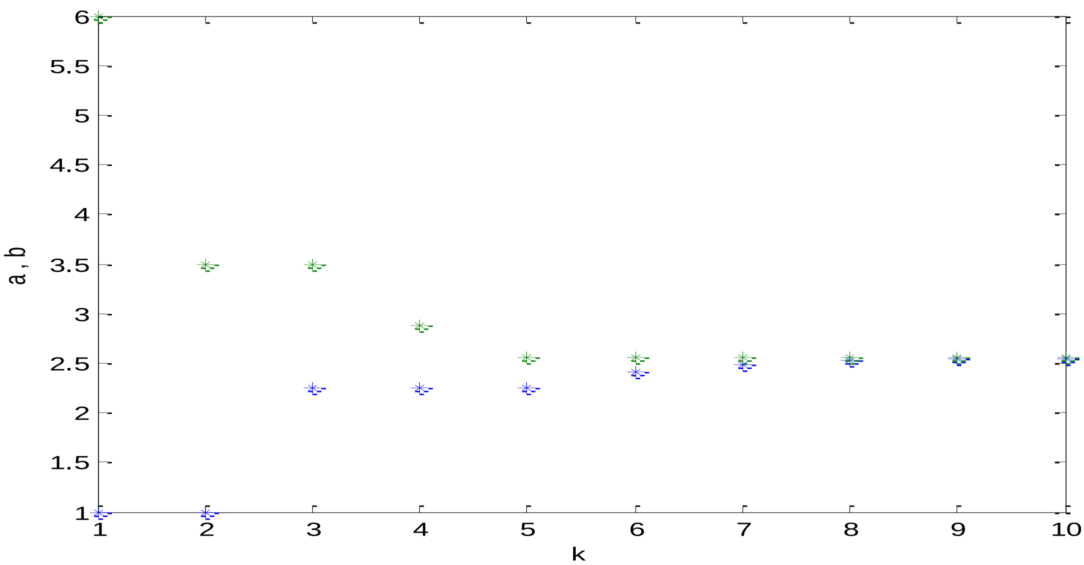
Παρατηρώντας τα αποτελέσματα βλέπουμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, ο αριθμός c των υπολογισμών της αντικειμενικής συνάρτησης μειώνεται. Αυτό συμβαίνει διότι μειώνοντας το τελικό εύρος αναζήτησης, μειώνουμε ουσιαστικά την τελική ακρίβεια που θέλουμε να έχουμε στον υπολογισμό του ελάχιστου της συνάρτησης και έτσι μειώνεται και ο αριθμός επαναλήψεων της μεθόδου, με συνέπεια να μειωθούν και οι υπολογισμοί της αντικειμενικής συνάρτησης.

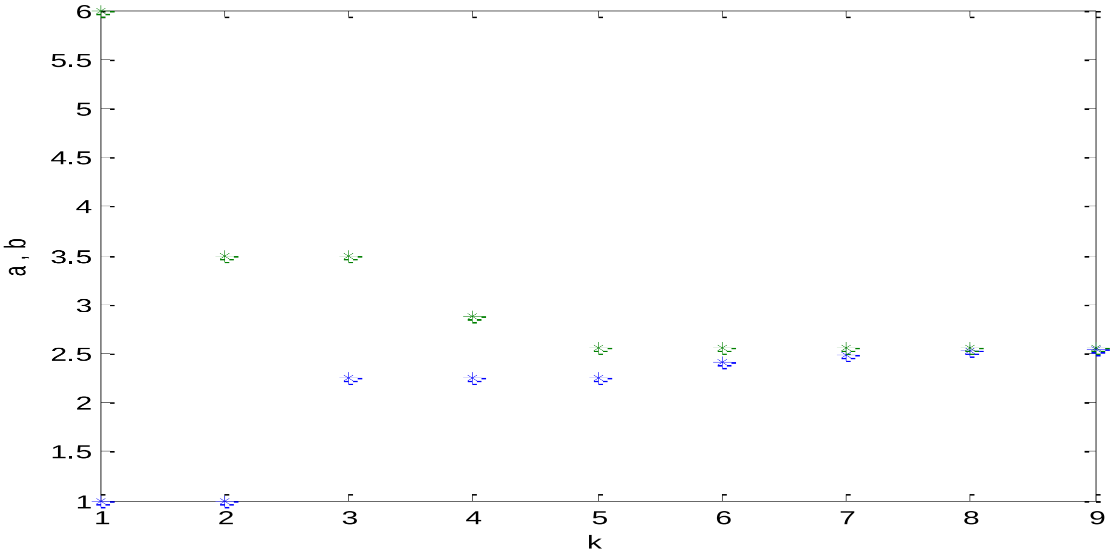
Μεταβάλλοντας τώρα το τελικό εύρος αναζήτησης l, υλοποιούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k,b_k]$ συναρτήσει του δείκτη βήματος k.

Επομένως για την f1 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:


6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
a , b
1
2
3
4
5
6
7
8
9
10
11
k

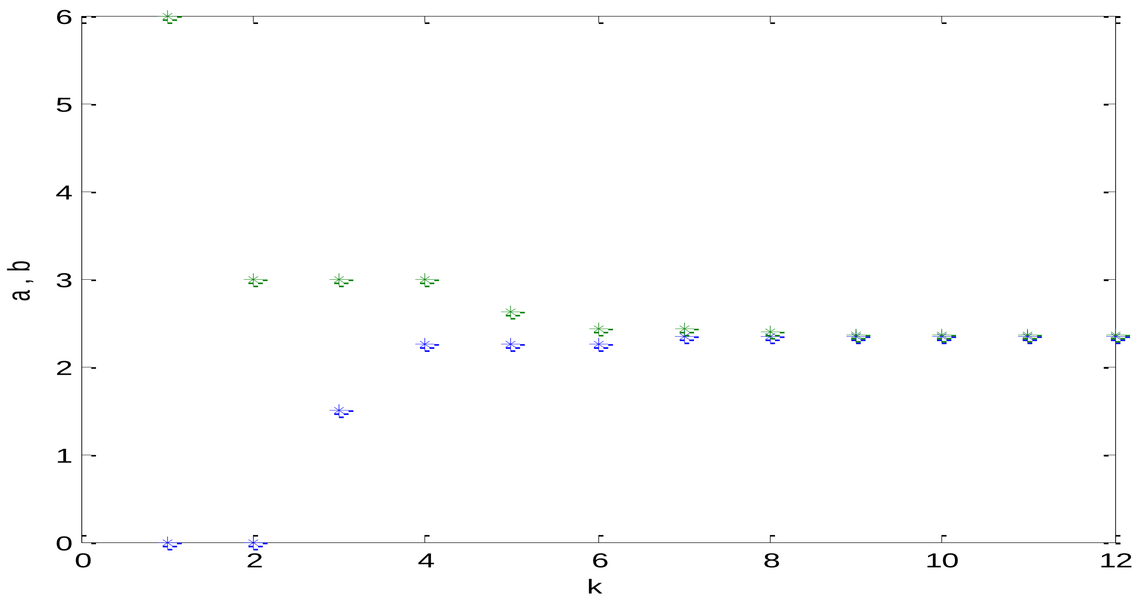

6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
a , b
1
2
3
4
5
6
7
8
9
10
k

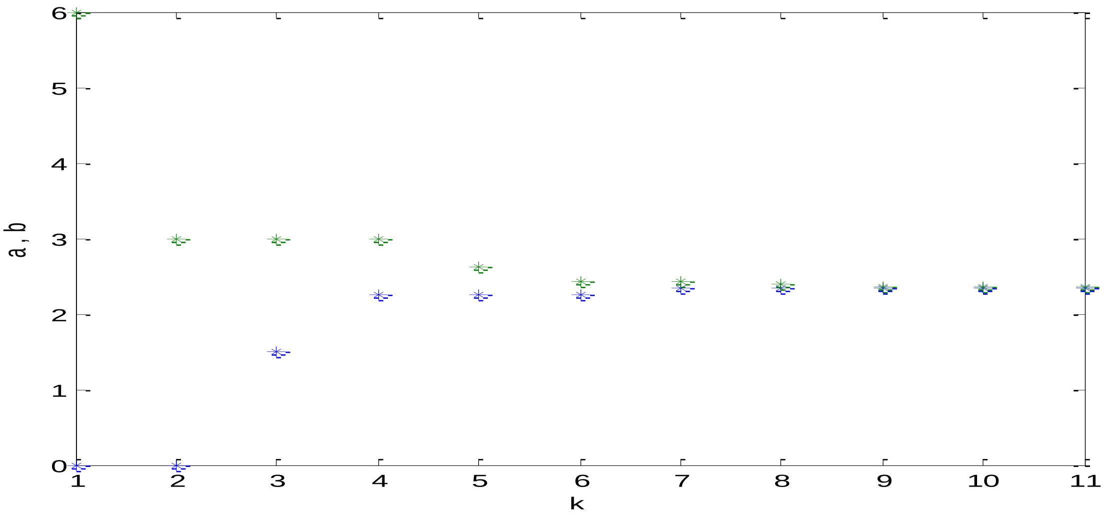

6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
a , b
1
2
3
4
5
6
7
8
9
10
k



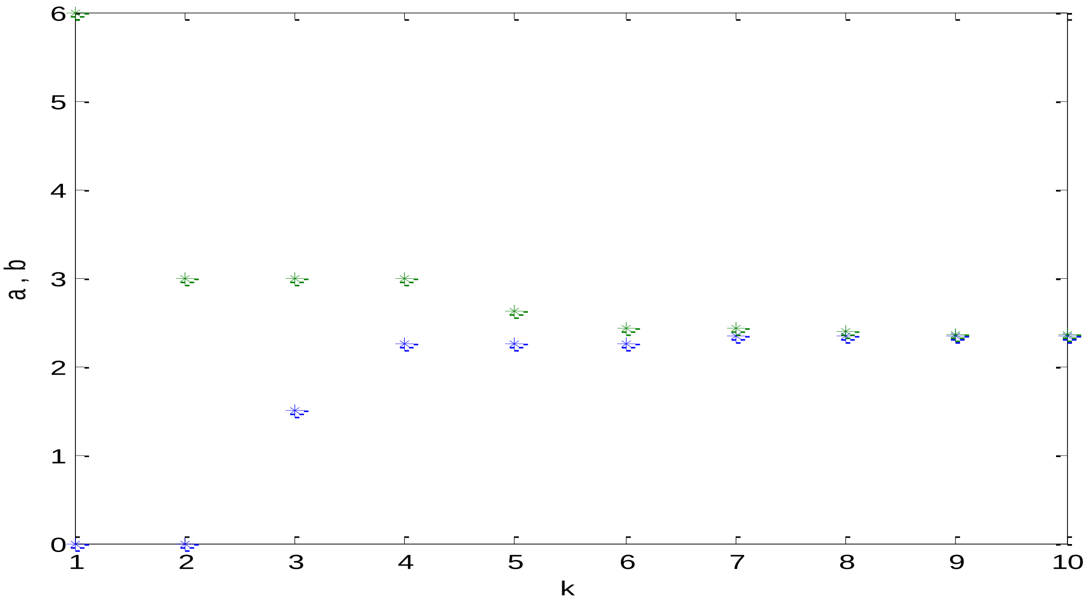
Παρατηρώντας τα διαγράμματα βλέπουμε πως και στα τέσσερα ο αλγόριθμος συγκλίνει στο τελικό διάστημα μετά από πεπερασμένο αριθμό βημάτων. Επίσης παρατηρούμε πως καθώς αυξάνεται το τελικό εύρος αναζήτησης l, μειώνεται ο αριθμός k των επαναλήψεων, διότι μειώνοντας την ακρίβεια ο αλγόριθμος υπολογίζει σε μικρότερο αριθμό βημάτων το τελικό διάστημα.

Για την f2 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:

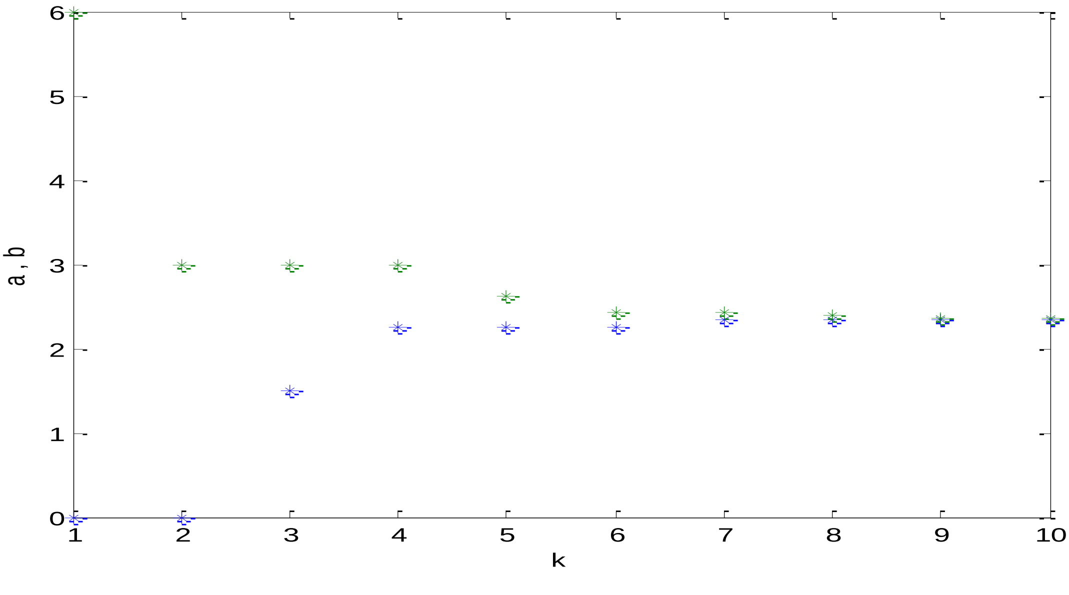




6
5
4
3
2
1
0
a, b
1
2
3
4
5
6
7
8
9
10
11
k



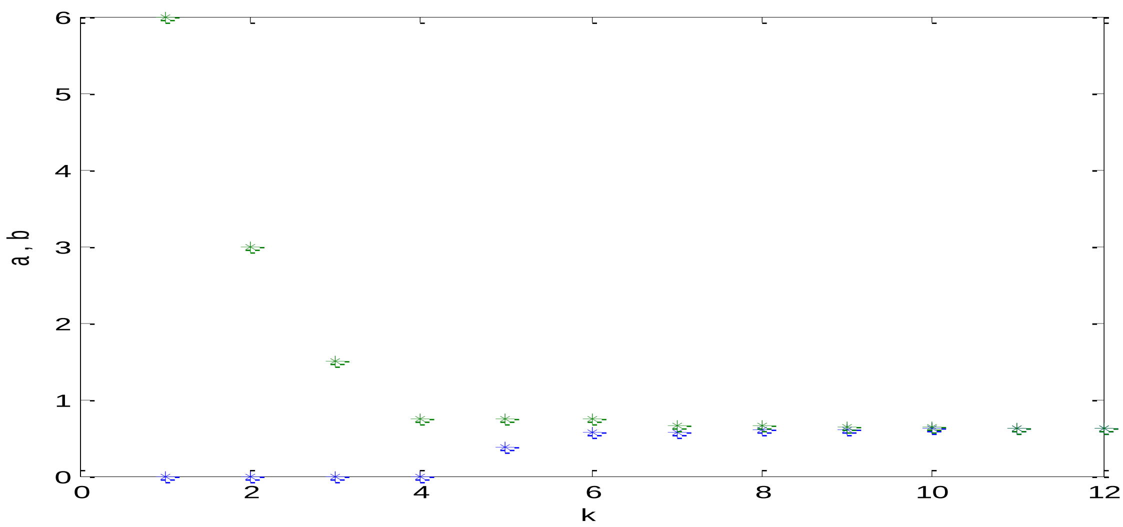
6
5
4
3
2
1
0
a, b
1
2
3
4
5
6
7
8
9
10
k

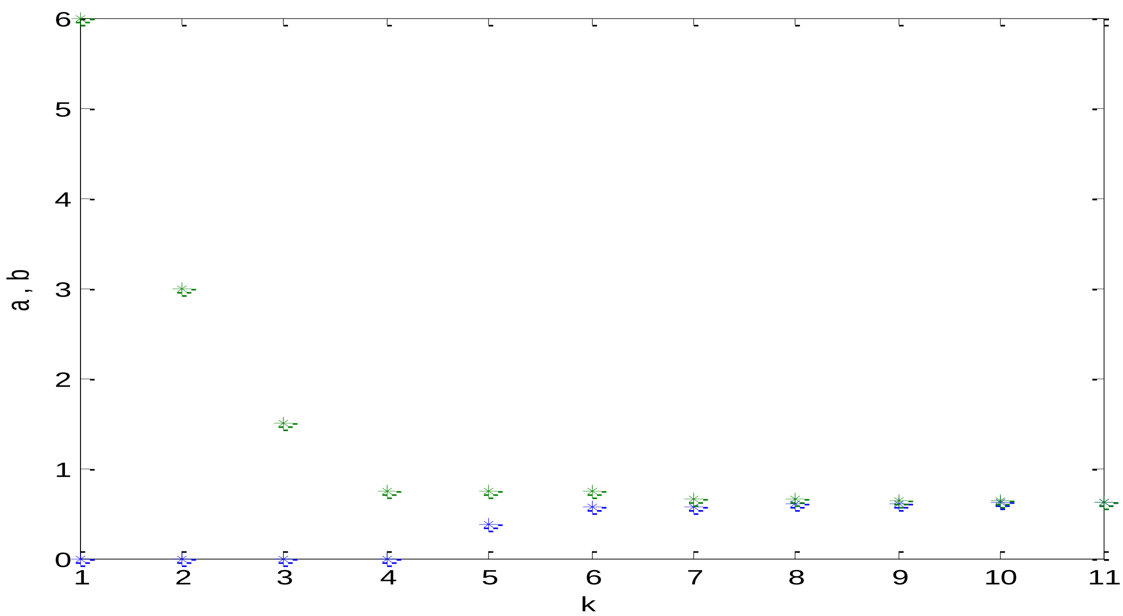


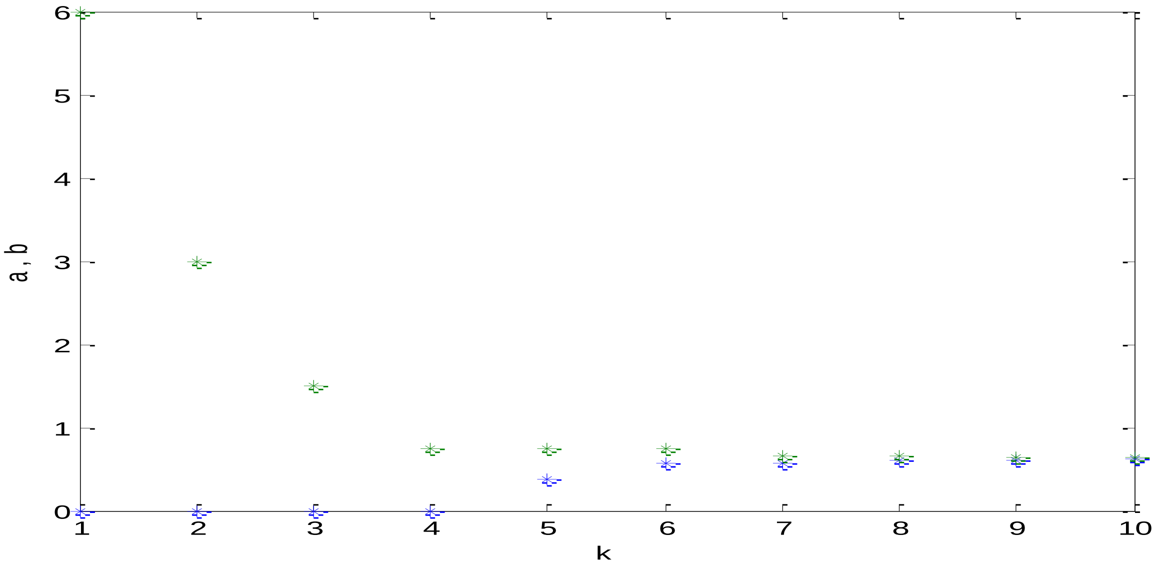
6
5
4
3
2
1
0
a, b
1
2
3
4
5
6
7
8
9
10
k

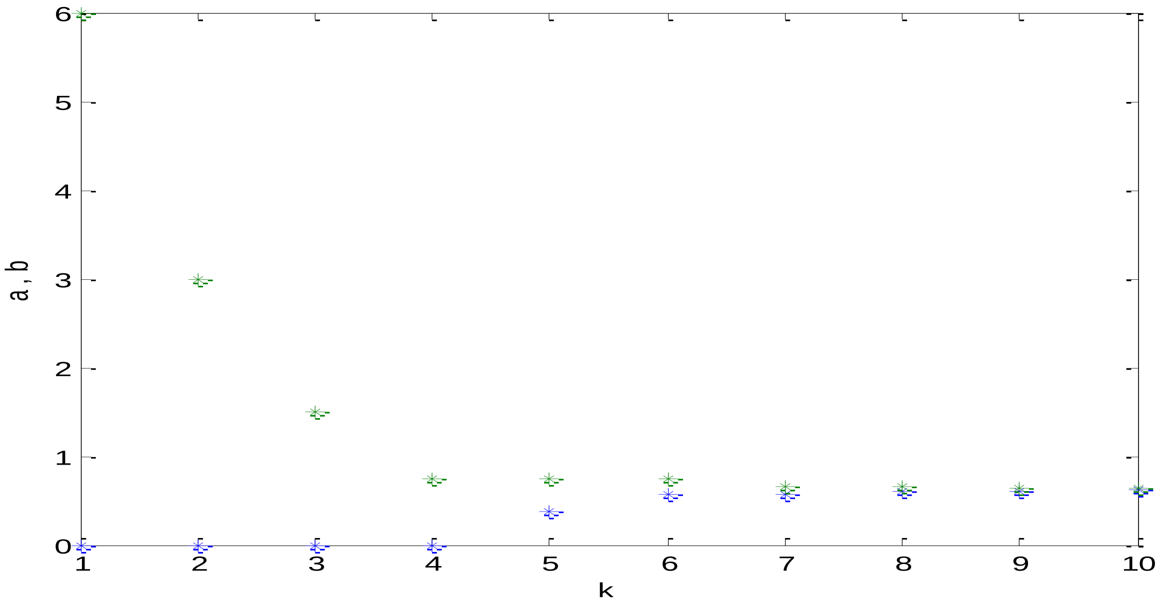
Παρατηρώντας και τα τέσσερα διαγράμματα, καταλήγουμε στα ίδια συμπεράσματα με την f1, δηλαδή ότι ο αλγόριθμος συγκλίνει έπειτα από πεπερασμένο αριθμό βημάτων και ότι αυξάνοντας το l μειώνεται ο αριθμός βημάτων k.

Για την f3 και για l=0.01, l=0.02, l=0.03 και l=0.04 προκύπτουν αντίστοιχα οι εξής τέσσερις γραφικές παραστάσεις:









Για την f3 προκύπτουν τα ίδια συμπεράσματα όπως και για τις f1,f2.

Για τη μέθοδο της διχοτόμου με τη χρήση παραγώγων, παρατηρούμε πως υπολογίζει σε έναν μικρό αριθμό βημάτων το τελικό διάστημα και έχει έναν μικρό αριθμό υπολογισμών της αντικειμενικής συνάρτησης, αυξάνοντας έτσι την ταχύτητα του αλγορίθμου.

Συμπέρασμα:

Συγκρίνοντας και τις τέσσερις μεθόδους ως προς την αποδοτικότητα τους καταλήγουμε στο συμπέρασμα πως ο αλγόριθμος της διχοτόμου με τη χρήση παραγώγων είναι ο πιο αποδοτικός, καθώς με μικρό αριθμό βημάτων και υπολογισμών της αντικειμενικής συνάρτησης, συγκλίνει στο τελικό διάστημα έχοντας την μεγαλύτερη ακρίβεια. Έπειτα ακολουθούν οι αλγόριθμοι Fibonacci και χρυσού τομέα, καθώς είναι και αυτοί γρήγοροι αφού δεν απαιτούν μεγάλο αριθμό υπολογισμών της αντικειμενικής συνάρτησης και τελευταίος έρχεται ο αλγόριθμος της διχοτόμου, ο οποίος είναι ο πιο αργός και από τους 3 έχοντας ωστόσο την ίδια περίπου ακρίβεια με τους προηγούμενους.