ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

2η Εργαστηριακή Άσκηση

Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής

σε δοσμένο διάστημα

ΑΝΤΡΕΑΣ ΧΑΤΖΗΘΩΜΑΣ

ΑΕΜ: 8026

Θέμα 1 (Μέθοδος της διχοτόμου)

Λίγα λόγια για τον κώδικα

Στο 1ο θέμα της εργαστηριακής άσκησης, υλοποιήσαμε στο Matlab τη μέθοδο της διχοτόμου χρησιμοποιώντας τον αλγόριθμο που αναγράφεται στο βιβλίο (5.1.1). Ο κώδικας του αλγόριθμου βρίσκεται στο αρχείο **bisection_alg.m**.

Συγκεκριμένα, η συνάρτηση **bisection_alg(y,a,b,l,e)** με την οποία υλοποιήσαμε τον αλγόριθμο της διχοτόμησης, δέχεται ως ορίσματα τη συνάρτηση **f(x)** στην οποία θέλουμε να υπολογίσουμε το ελάχιστο **x***, τα αρχικά όρια αναζήτησης **a** και **b**, τη μεταβλητή **l** η οποία αντιστοιχεί στο τελικό εύρος αναζήτησης και τη μεταβλητή **e** η οποία αντιστοιχεί στην απόσταση από την διχοτόμο. Τρέχοντας τη συνάρτηση του αλγορίθμου για συγκεκριμένες τιμές, μας επιστρέφει τον αριθμό **k** που είναι ο αριθμός επαναλήψεων, τον αριθμό **n** που αντιστοιχεί στον αριθμό υπολογισμών της αντικειμενικής συνάρτησης και τα εκάστοτε όρια **a(i)** και **b(i)** όπου i=1,2,..,k τα οποία επιστρέφονται υπό την μορφή μονοδιάστατου πίνακα με k στοιχεία.

Επίσης για να βρούμε τον αριθμό υπολογισμών της αντικειμενικής συνάρτησης χρησιμοποιούμε ένα μετρητή εντός του αλγόριθμου που υλοποιήσαμε. Εναλλακτικά, μπορούσαμε να χρησιμοποιήσουμε τον τύπο της σελ. 114 του βιβλίου $\left(\frac{1}{2}\right)^{n/2} \geq \frac{l}{b1-a1}$ ωστόσο θα προτιμήσουμε να τον αποφύγουμε διότι με τον τύπο αυτό παίρνουμε μια εκτίμηση του n και κατά συνέπεια έχουμε σε ορισμένες περιπτώσεις έχουμε μια μικρή απόκλιση.

Εφαρμογή του αλγόριθμου

Εφόσον τελειώσαμε με την υλοποίηση του αλγόριθμου και τον κώδικα, τρέχουμε τον αλγόριθμο. Για αρχή, τρέχουμε τον αλγόριθμο για την κάθε συνάρτηση ξεχωριστά με e=0.001, l=0.01 και αρχικά όρια για την $f_1(x)$: a=1, b=6 και για τις $f_2(x)$, $f_3(x)$: a=0, b=6.

Τα τελικά όρια που παίρνουμε για την κάθε συνάρτηση είναι:

$f_1(x)$: x* ϵ [2.5472, 2.5541]

$f_2(x)$: x* ϵ [2.3488, 2.3567]

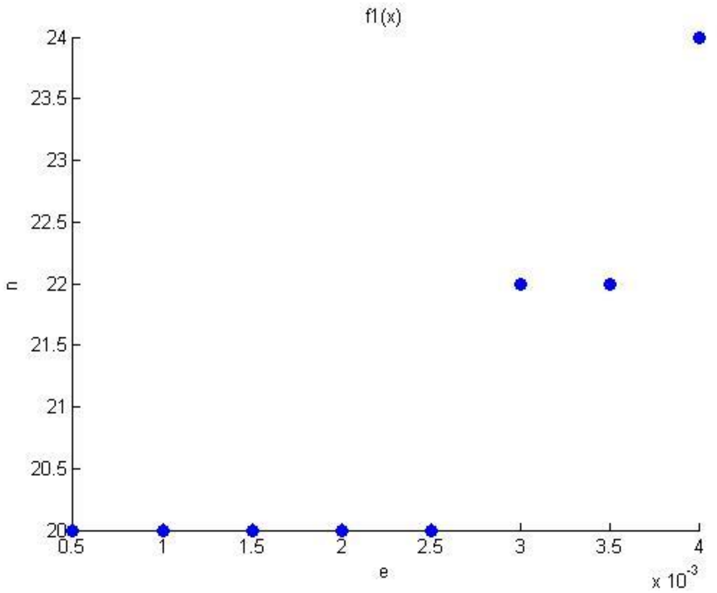
$f_3(x)$: x* ϵ [0.6209, 0.6287]

Από τα τελικά όρια κάθε συνάρτησης αντίστοιχα παρατηρούμε ότι τηρείται η ανισότητα b_k-a_k < l δηλαδή το τελικό εύρος είναι μικρότερο από το l που διαλέξαμε. Οπότε, ο αλγόριθμος επιστρέφει σωστές τιμές βάσει προδιαγραφής.

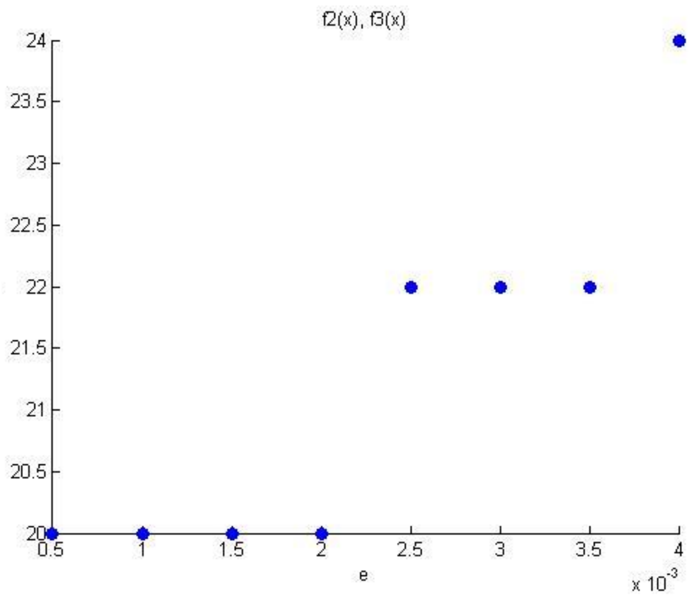
Εφαρμογή του αλγόριθμου για διάφορες τιμές - Παρατηρήσεις και Συμπεράσματα

Α. Για το (α) ερώτημα, τρέχουμε τον αλγόριθμο για την $f_1(x)$ με αρχικά όρια a=1 και b=6. Κρατάμε σταθερό το τελικό εύρος αναζήτησης l=0.01 και μεταβάλλουμε το e ξεκινώντας από 0.0005 μέχρι 0.0040 με βήμα 0.0005. Αφού τρέξουμε τον αλγόριθμο για τις διάφορες τιμές του e, παρατηρούμε την μεταβολή των υπολογισμών της αντικειμενικής συνάρτησης και δημιουργούμε την αντίστοιχη γραφική παράσταση όπως φαίνεται στο Σχήμα 1.α1.

Ακολουθούμε την ίδια διαδικασία και για τις συναρτήσεις $f_2(x)$ και $f_3(x)$ όμως με αρχικά όρια a=0 και b=6 με το βήμα για το e να παραμένει ίδιο. Παρατηρούμε ότι οι γραφικές παραστάσεις για τις $f_2(x)$ και $f_3(x)$ είναι ίδιες γι' αυτό θα τις απεικονίσουμε στο ίδιο σχήμα (Σχ. 1.α2)



Σχήμα 1.α1

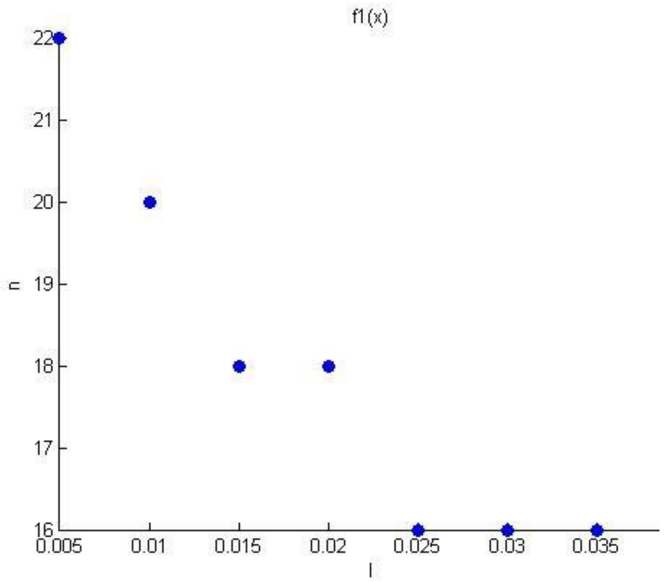


Σχήμα 1.α2

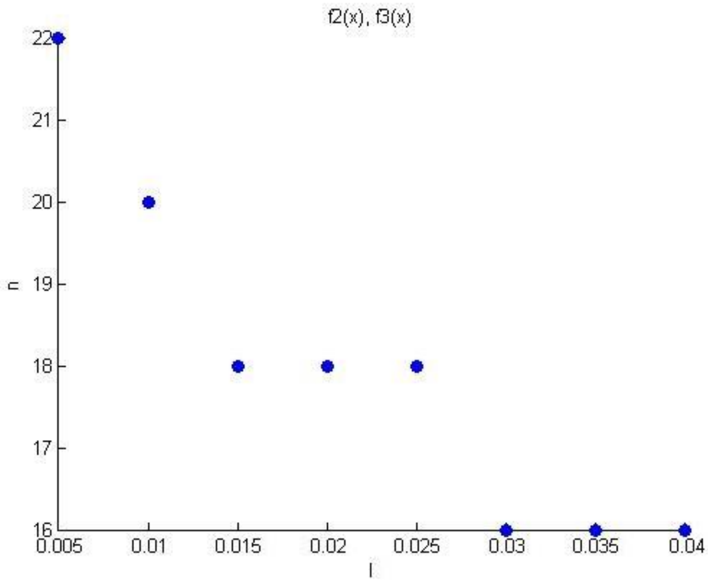
Αυτό που παρατηρούμε από τα Σχήματα 1.α1 - 1.α2 είναι ότι καθώς αυξάνουμε την απόσταση από τη διχοτόμο, αυξάνεται και ο αριθμός υπολογισμών της αντικειμενικής συνάρτησης. Αυτό ήταν αναμενόμενο διότι αυξάνοντας την απόσταση από τη διχοτόμο, αυξάνεται και η απόσταση μεταξύ των σημείων x1 και x2, οπότε για να πετύχουμε τελικό εύρος μικρότερο του l=0.01 απαιτούνται περισσότερα βήματα και αυτό συνεπάγεται σε περισσότερους υπολογισμούς της αντικειμενικής συνάρτησης. Αυτό εύκολα φαίνεται και μέσα από τους τύπους του βιβλίου για τα x1 και x2 όπου $x1,2k = \frac{ak+bk}{2} \pm e$.

Επίσης, ο ρυθμός με τον οποίο αυξάνεται το n συναρτήσει του e είναι ανάλογα με την συνάρτηση στην οποία εφαρμόζουμε τον αλγόριθμο. Στην $1^η$ συνάρτηση βλέπουμε ότι το n αυξάνεται από 20 σε 22 στο e=0.003. Στη $2^η$ και $3^η$ συνάρτηση το n αυξάνεται από 20 σε 22 στο e=0.025.

Β. Ακολούθως, για το ερώτημα (β), εκτελούμε παρόμοια διαδικασία με πριν, κρατώντας τώρα σταθερό το e=0.001 και μεταβάλλοντας το τελικό εύρος αναζήτησης l ξεκινώντας από 0.005 μέχρι 0.040 με βήμα 0.005 Τα αρχικά όρια είναι ίδια με πριν για κάθε συνάρτηση αντίστοιχα. Οι γραφικές παραστάσεις που προκύπτουν είναι οι παρακάτω. (Σχ. 1.β1 και Σχ. 1.β2)



Σχήμα 1.β1

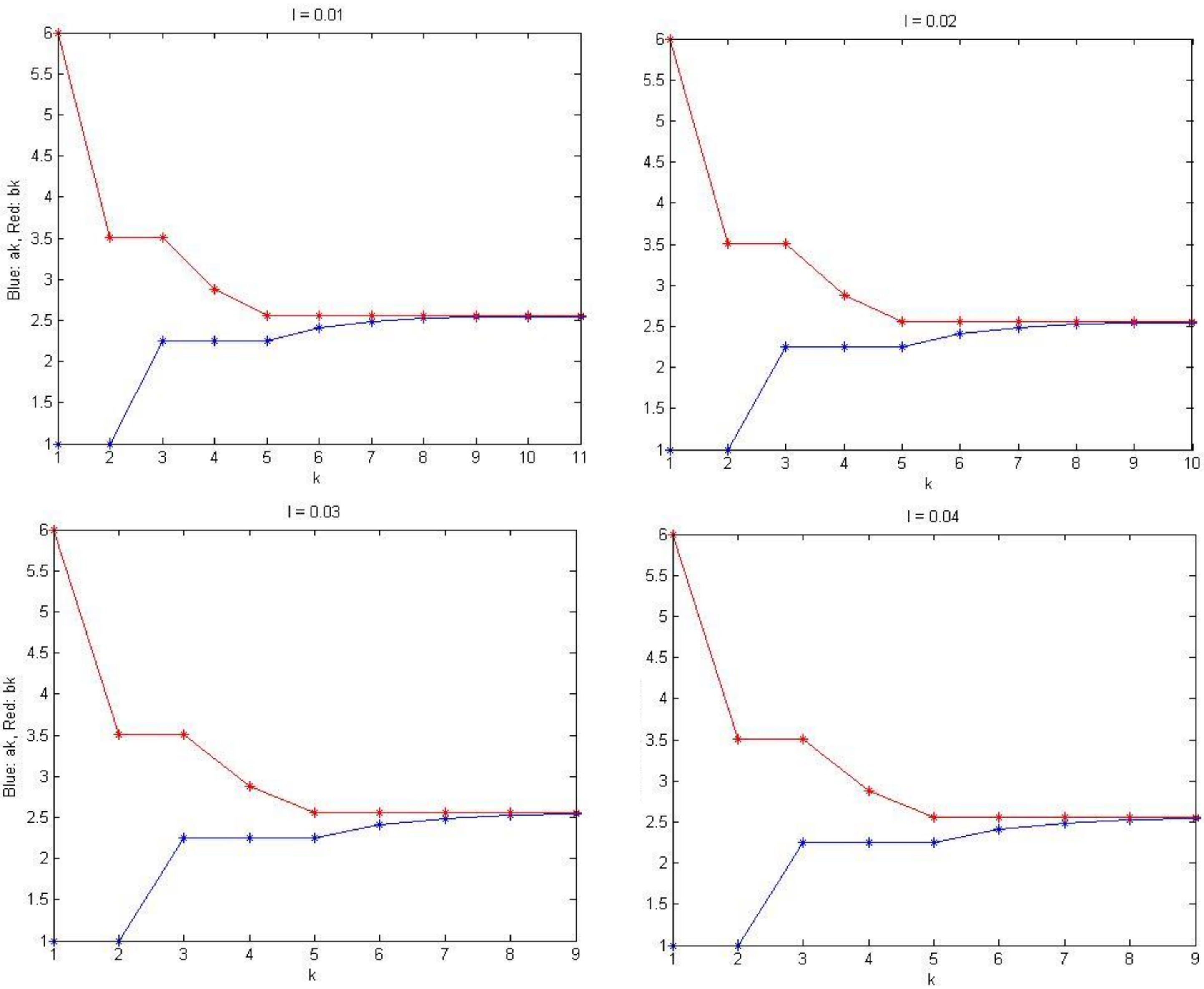


Σχήμα 1.β2

Εδώ παρατηρούμε ότι καθώς αυξάνουμε την ακρίβεια l, οι υπολογισμοί της αντικειμενικής συνάρτησης μειώνονται. Αυτό είναι λογικό διότι αυξάνοντας το l, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την ζητούμενη ακρίβεια που αυτό συνεπάγεται και στη μείωση του αριθμού των υπολογισμών της αντικειμενικής συνάρτησης. Αυτό γίνεται εύκολα αντιληπτό από τις παραπάνω γραφικές παραστάσεις αλλά και από τον τύπο $bk - ak \leq l$.

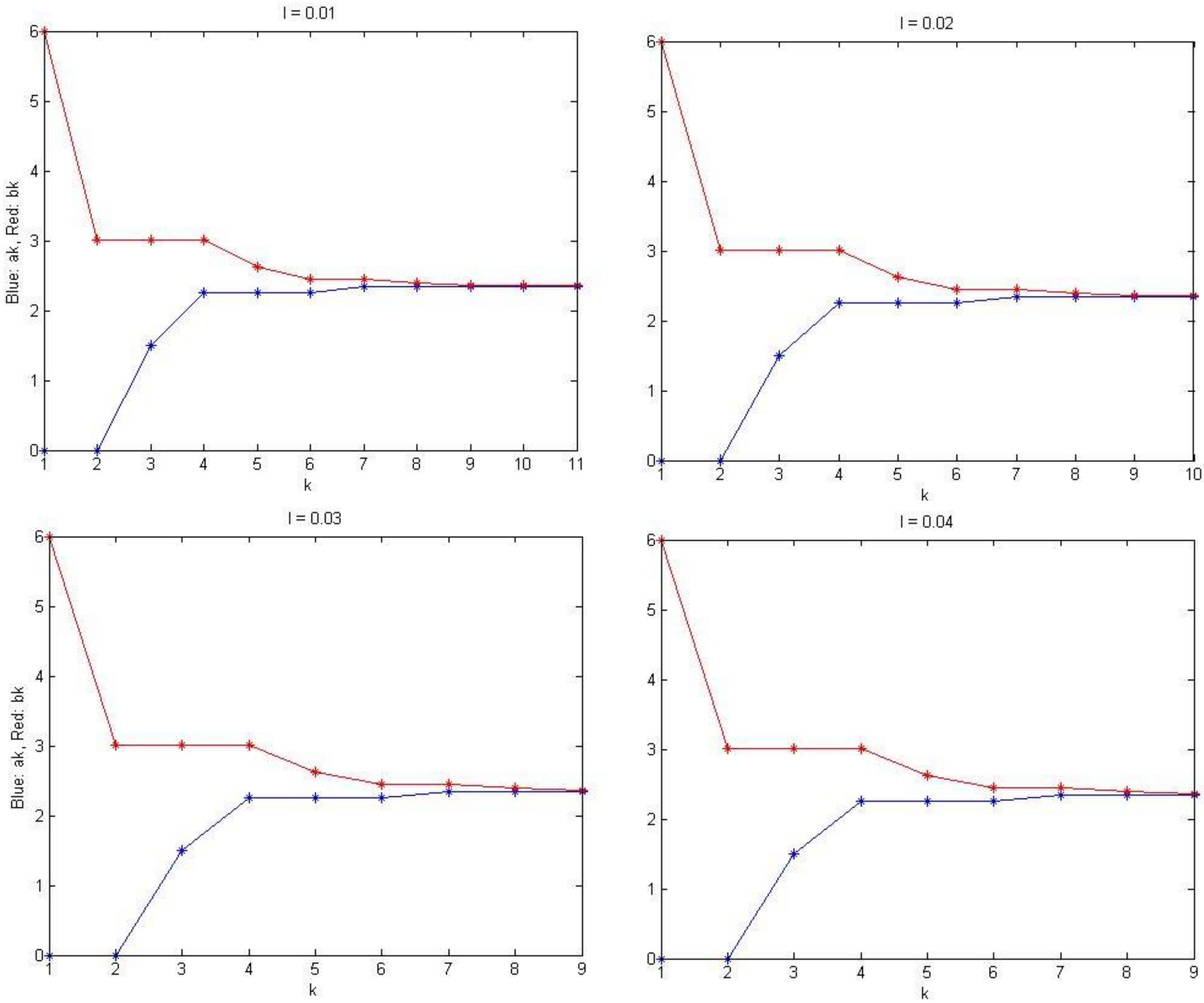
Γ. Στο ερώτημα (γ) μεταβάλλουμε το l από 0.01 μέχρι 0.04 με βήμα 0.01 και κρατάμε σταθερό το e=0.001. Αφού τρέξουμε τον αλγόριθμο για τα διάφορα l, δημιουργούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k, b_k]$ συναρτήσει του βήματος k, για κάθε συνάρτηση ξεχωριστά.

Οι γραφικές παραστάσεις απεικονίζονται παρακάτω. Η $f_1(x)$ απεικονίζεται στο Σχήμα 1.γ1 και οι $f_2(x)$ και $f_3(x)$ στα Σχήματα 1.γ2 και 1.γ3 αντίστοιχα.

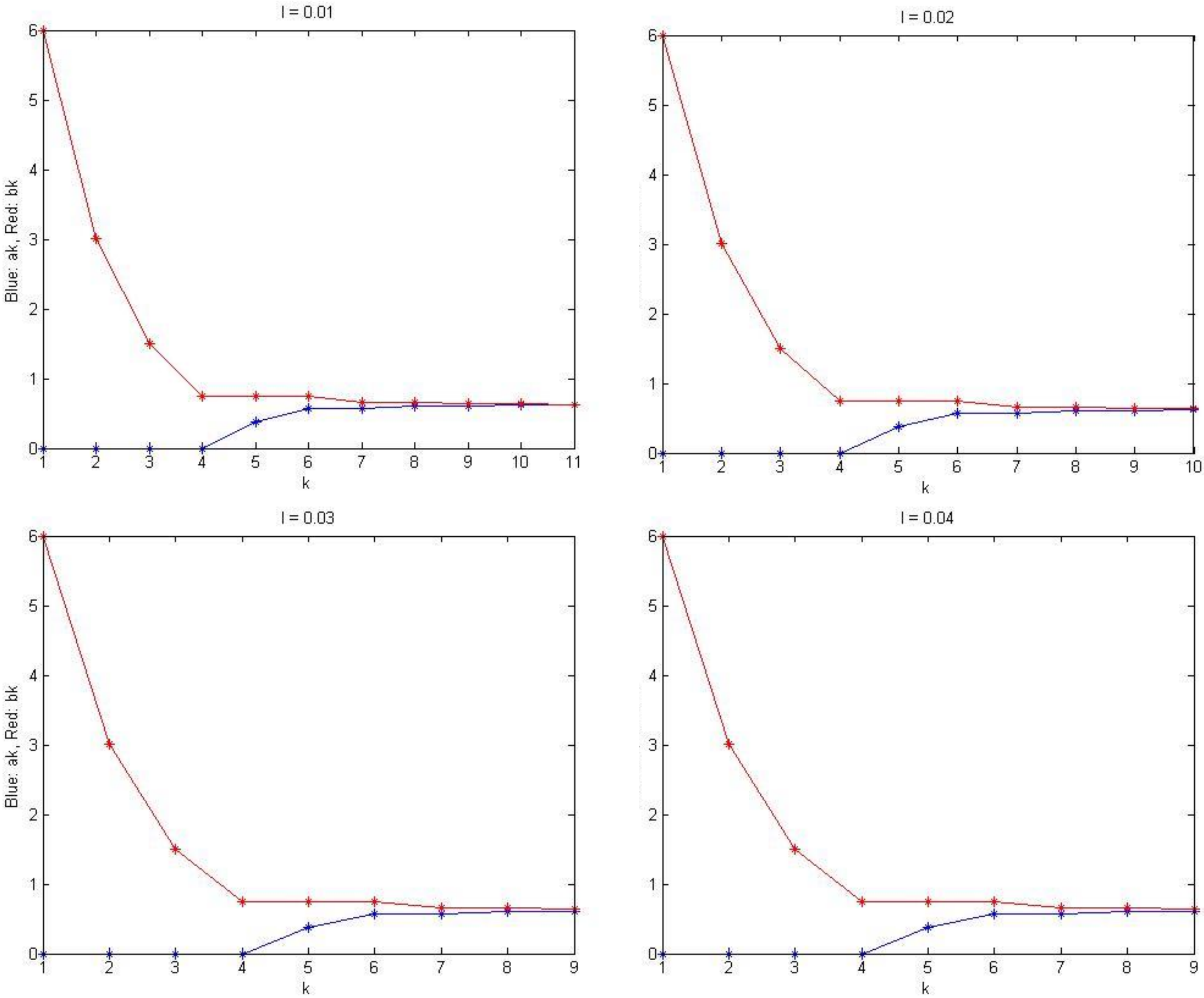


Σχήμα 1.γ1: $f_1(x)$

Εδώ παρατηρούμε ότι αυξάνοντας το l, μειώνεται ο αριθμός επαναλήψεων. Αυτό είναι λογικό διότι αυξάνοντας το l, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την ζητούμενη ακρίβεια, όπως δηλαδή και στο ερώτημα (β) από το σχήμα 1.β1 στο οποίο παρατηρούσαμε τη μείωση του αριθμού των υπολογισμών της αντικειμενικής συνάρτησης.Επίσης παρατηρούμε ότι οι γραφικές παραστάσεις ταυτίζονται μέχρι ένα συγκεκριμένο κ, ανάλογα με την ακρίβεια. Αυτό συμβαίνει διότι για μεγαλύτερη ακρίβεια (δηλ. μικρότερο l) ο αλγόριθμος απαιτεί περισσότερα βήματα, οπότε σε αυτά τα επιπλέον βήματα που απαιτούνται, το διάγραμμα του αλγορίθμου παύει να ταυτίζεται με τα υπόλοιπα διαγράμματα μικρότερης ακρίβειας.



Σχήμα 1.γ2: $f_2(x)$



Σχήμα 1.γ3: $f_3(x)$

Για τις συναρτήσεις $f_2(x)$ και $f_3(x)$ που φαίνονται στα σχήματα 1.γ2 και 1.γ3 αντίστοιχα, καταλήγουμε στα ίδια συμπεράσματα με τα συμπεράσματα της συνάρτησης $f_1(x)$ του Σχ. 1.γ1.

Θέμα 2 (Μέθοδος του χρυσού τομέα)

Λίγα λόγια για τον κώδικα

Όπως και στο 1ο θέμα, έτσι κ εδώ, υλοποιήσαμε στο Matlab τη μέθοδο του χρυσού τομέα χρησιμοποιώντας τον αλγόριθμο που αναγράφεται στο βιβλίο (5.1.2). Ο κώδικας του αλγόριθμου βρίσκεται στο αρχείο **golden_section_alg.m**.

Συγκεκριμένα, η συνάρτηση **golden_section_alg(y,a,b,l)** με την οποία υλοποιήσαμε τον αλγόριθμο του χρυσού τομέα, δέχεται ως ορίσματα τη συνάρτηση **f(x)** στην οποία θέλουμε να υπολογίσουμε το ελάχιστο **x***, τα αρχικά όρια αναζήτησης **a** και **b**, και τη μεταβλητή **l** η οποία αντιστοιχεί στο τελικό εύρος. Τρέχοντας τη συνάρτηση του αλγορίθμου για συγκεκριμένες τιμές, μας επιστρέφει τον αριθμό **k** που είναι ο αριθμός επαναλήψεων, τον αριθμό **n** που αντιστοιχεί στον αριθμό υπολογισμών της αντικειμενικής συνάρτησης και τα εκάστοτε όρια **a(i)** και **b(i)** όπου i=1,2,..,k τα οποία επιστρέφονται υπό τη μορφή μονοδιάστατου πίνακα με k στοιχεία.

Επίσης για να βρούμε τον αριθμό υπολογισμών της αντικειμενικής συνάρτησης χρησιμοποιήσαμε ένα μετρητή. Εναλλακτικά, μπορούσαμε να χρησιμοποιήσουμε τον τύπο της σελ. 114 του βιβλίου $(0{,}618)^{n-1} \geq \frac{l}{b1-a1}$ ωστόσο όπως και στο 1ο θέμα, έτσι κ εδώ θα προτιμήσουμε να τον αποφύγουμε για τον προαναφερθέντα λόγο.

Εφαρμογή του αλγόριθμου

Εφόσον τελειώσαμε με την υλοποίηση του αλγόριθμου και τον κώδικα, τρέχουμε τον αλγόριθμο. Για αρχή, τρέχουμε τον αλγόριθμο για την κάθε συνάρτηση ξεχωριστά με l=0.01 και αρχικά όρια για την $f_1(x)$: a=1, b=6 και για τις $f_2(x)$, $f_3(x)$: a=0, b=6.

Τα τελικά όρια που παίρνουμε για την κάθε συνάρτηση είναι:

$f_1(x)$: x* ϵ [2.5463, 2.5559]

$f_2(x)$: x* ϵ [2.3478, 2.3548]

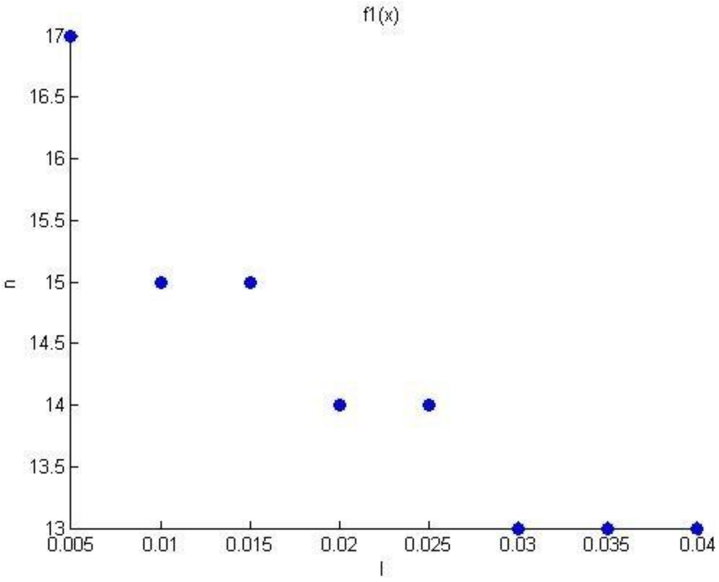
$f_3(x)$: x* ϵ [0.6200, 0.6272]

Από τα τελικά όρια κάθε συνάρτησης αντίστοιχα παρατηρούμε ότι τηρείται η ανισότητα b_k-a_k < l δηλαδή το τελικό εύρος είναι μικρότερο από το l που διαλέξαμε. Οπότε, ο αλγόριθμος επιστρέφει σωστές τιμές βάσει προδιαγραφής

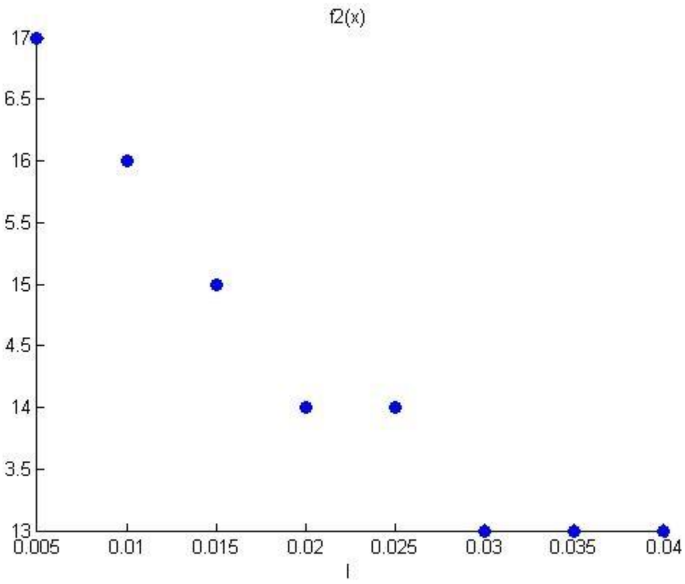
Εφαρμογή του αλγόριθμου για διάφορες τιμές - Παρατηρήσεις και Συμπεράσματα

Α. Για το (α) ερώτημα, τρέχουμε τον αλγόριθμο για την $f_1(x)$ με αρχικά όρια a=1 και b=6. Μεταβάλλουμε το τελικό εύρος αναζήτησης l από 0.005 μέχρι 0.04 και με βήμα 0.005. Αφού τρέξουμε τον αλγόριθμο για τις διάφορες τιμές του l, παρατηρούμε την μεταβολή των υπολογισμών της αντικειμενικής συνάρτησης και δημιουργούμε την αντίστοιχη γραφική παράσταση όπως φαίνεται στο Σχήμα 2.α1.

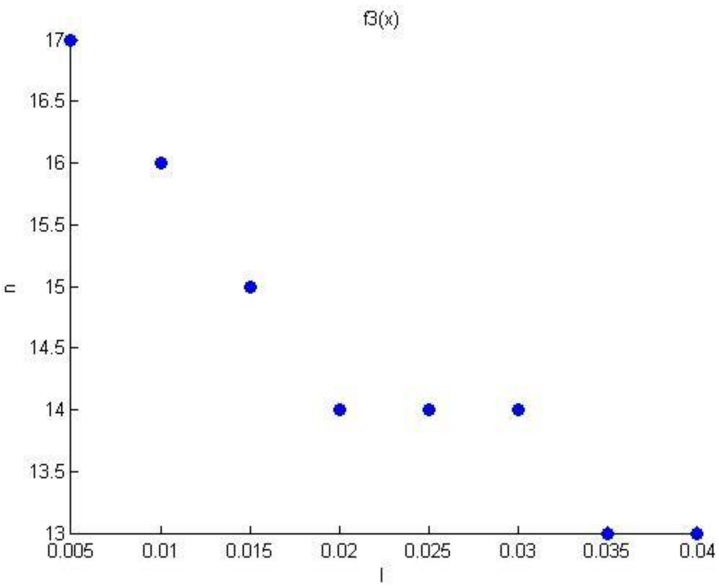
Ακολουθούμε την ίδια διαδικασία και για τις συναρτήσεις $f_2(x)$ και $f_3(x)$ όμως με αρχικά όρια a=0 και b=6. (Σχήματα 2.α2 και 2.α3).



Σχήμα 2.α1



Σχήμα 2.α2

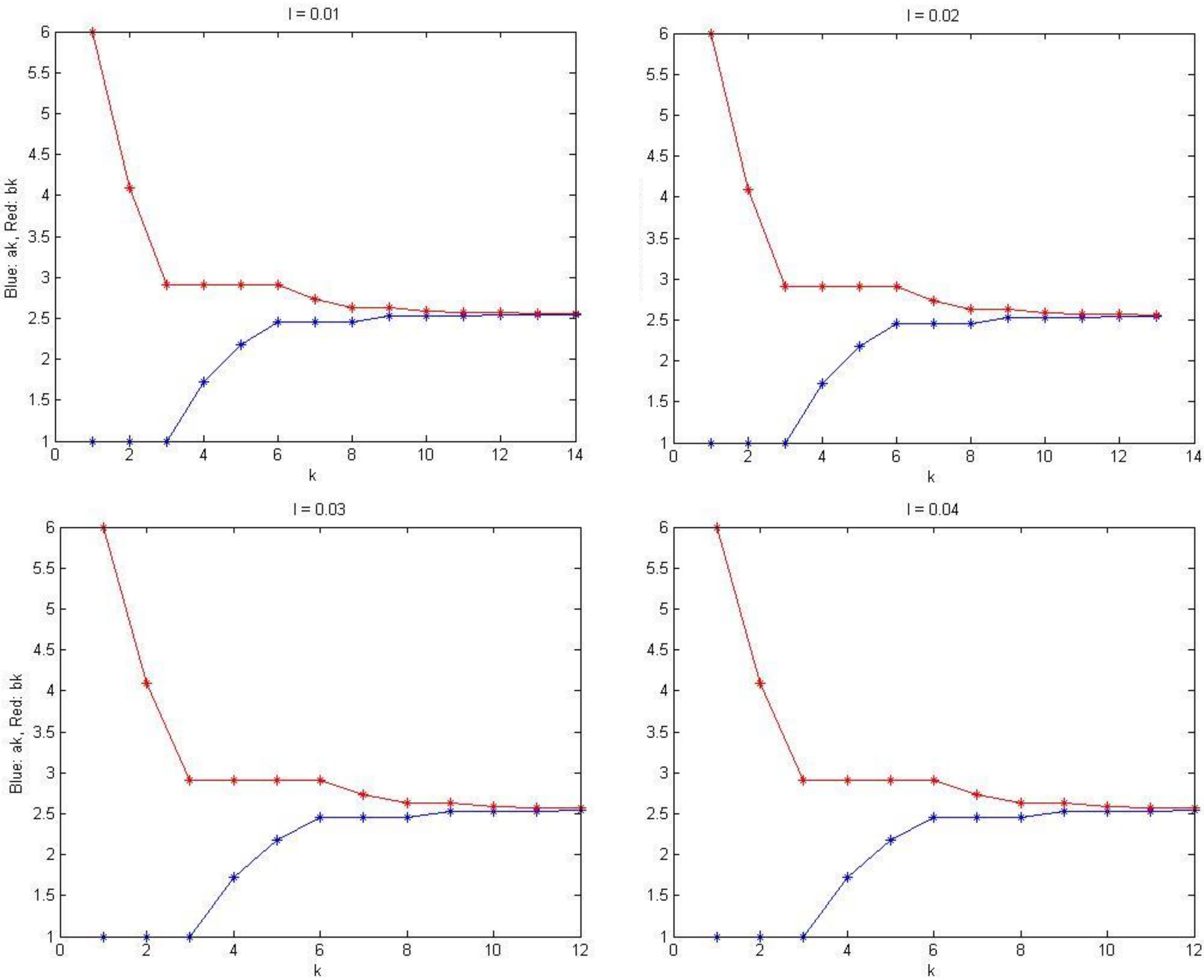


Σχήμα 2.α3

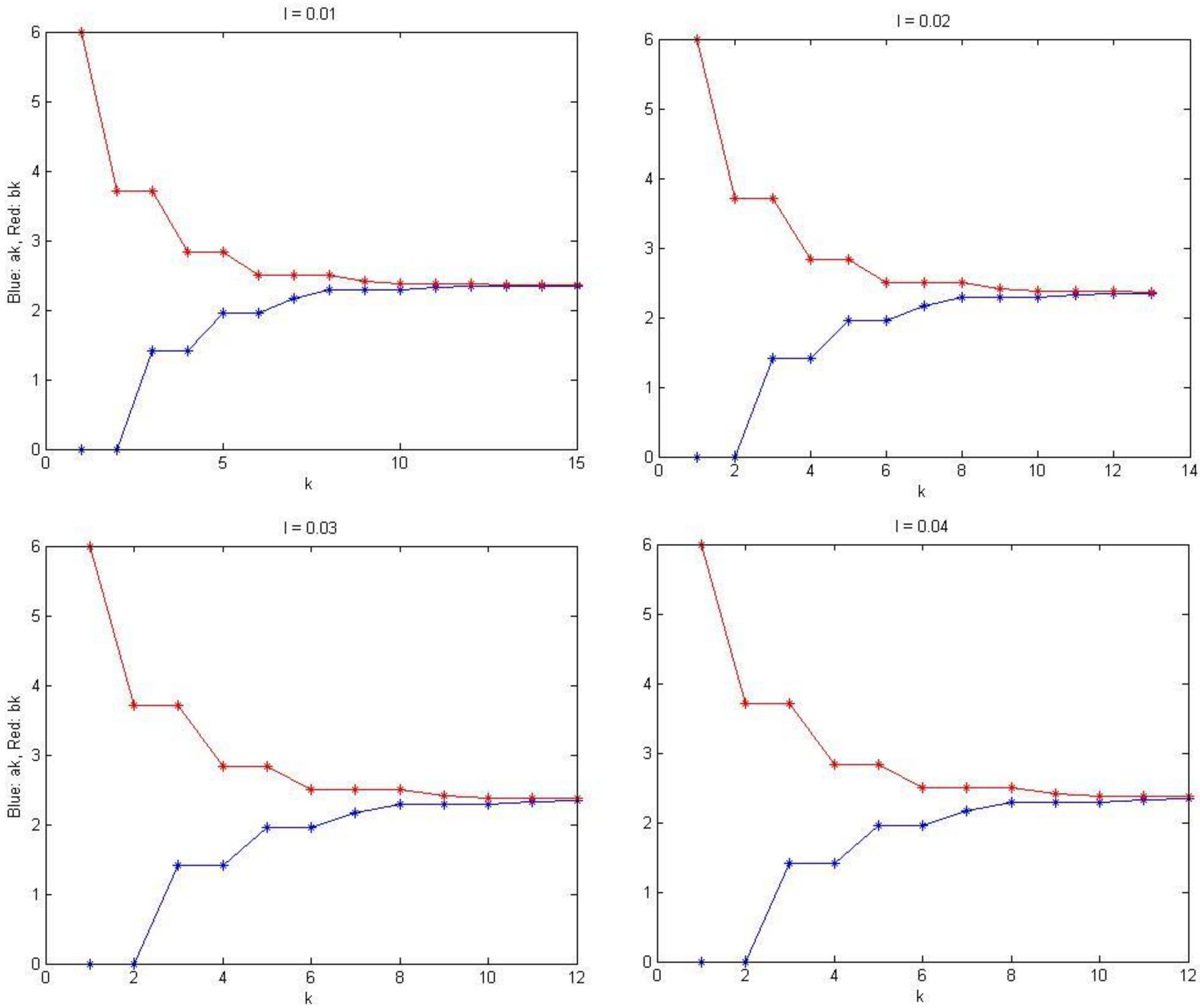
Εδώ παρατηρούμε ότι καθώς αυξάνουμε την ακρίβεια l, οι υπολογισμοί της αντικειμενικής συνάρτησης μειώνονται. Αυτό είναι λογικό διότι αυξάνοντας το l, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την προδιαγεγραμμένη ακρίβεια που αυτό συνεπάγεται και στη μείωση του αριθμού των υπολογισμών της αντικειμενικής συνάρτησης. Καταλήγουμε δηλαδή, στο ίδιο συμπέρασμα με το ερώτημα (β) του 1ου θέματος, ωστόσο, ο αριθμός υπολογισμών της αντικειμενικής συνάρτησης είναι μικρότερος από αυτόν της προηγούμενης μεθόδου για τα αντίστοιχα

Β. Στο ερώτημα (β) μεταβάλλουμε το l από 0.01 μέχρι 0.04 με βήμα 0.01. Αφού τρέξουμε τον αλγόριθμο για τα διάφορα l, δημιουργούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k, b_k]$ συναρτήσει του βήματος k, για κάθε συνάρτηση ξεχωριστά.

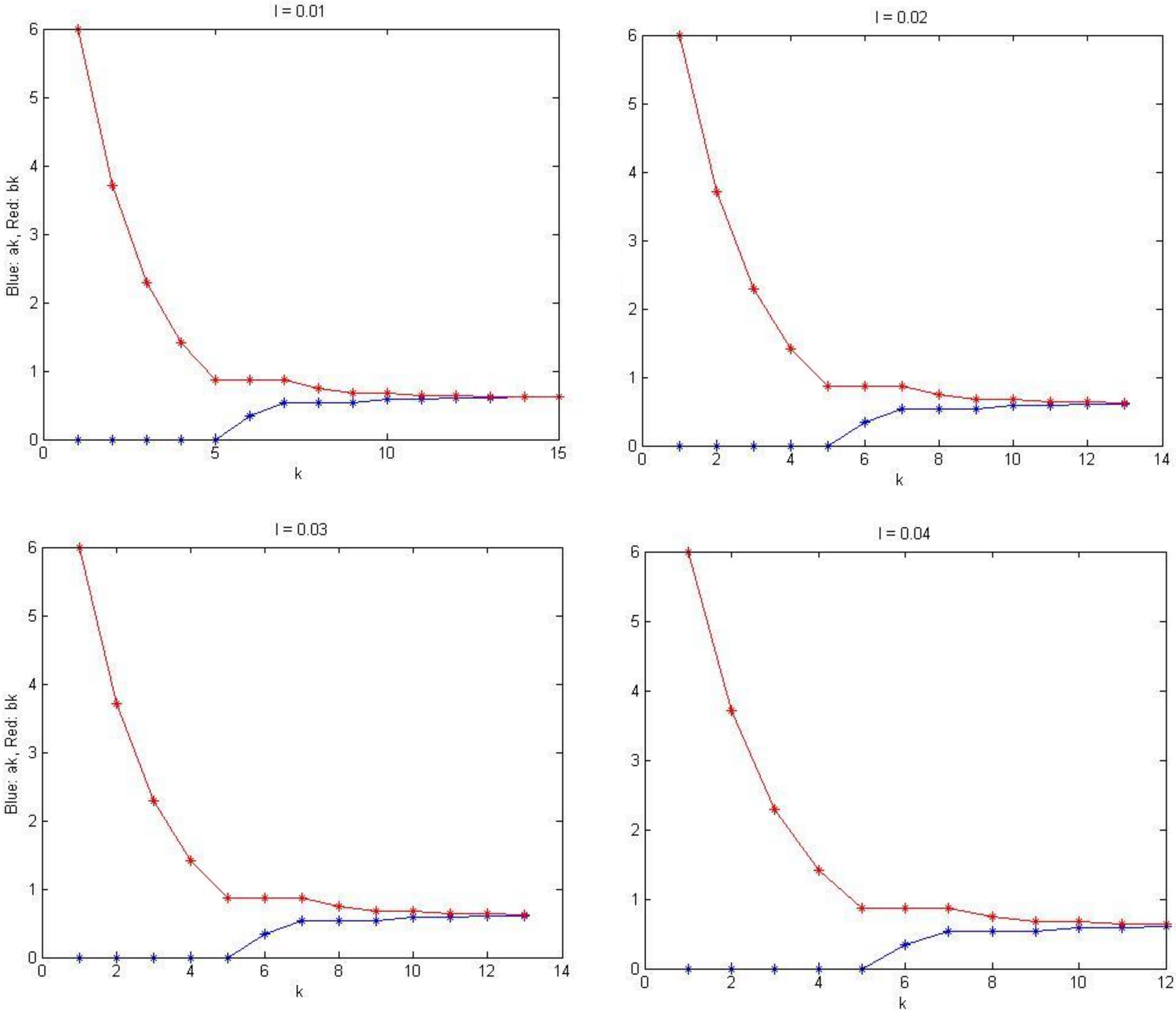
Οι γραφικές παραστάσεις απεικονίζονται παρακάτω. Η $f_1(x)$ απεικονίζεται στο Σχήμα 2.β1 και οι $f_2(x)$ και $f_3(x)$ στα Σχήματα 2.β2 και 2.β3 αντίστοιχα.



Σχήμα 2.β1: $f_1(x)$



Σχήμα 2.β2: $f_2(x)$



Σχήμα 2.β3: $f_3(x)$

Και στις 3 συναρτήσεις παρατηρούμε ότι αυξάνοντας το l, μειώνεται ο αριθμός επαναλήψεων, διότι αυξάνοντας το l η απαιτούμενη ακρίβεια μειώνεται, επομένως ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την επιλεγμένη ακρίβεια.

Παρατηρούμε ακόμη, ότι οι γραφικές παραστάσεις κάθε συνάρτησης, ταυτίζονται μέχρι ένα συγκεκριμένο κ, ανάλογα με την ακρίβεια. Αυτό συμβαίνει διότι για μεγαλύτερη ακρίβεια (δηλ. μικρότερο l) ο αλγόριθμος απαιτεί περισσότερα βήματα, οπότε σε αυτά τα επιπλέον βήματα που απαιτούνται το διάγραμμα του αλγορίθμου παύει να ταυτίζεται με τα υπόλοιπα διαγράμματα μικρότερης ακρίβειας.

Επίσης, για l=0.01 η $f_1(x)$ διαφέρει κατά μια (1) επανάληψη από τις συναρτήσεις $f_2(x)$ και $f_3(x)$, πράγμα το οποίο δεν παρατηρήθηκε στη μέθοδο της διχοτόμου. Επομένως, ίσως να μπορούσαμε να πούμε ότι ο αριθμός k σε ορισμένες περιπτώσεις εξαρτάται και από τη συνάρτηση στην οποία εφαρμόζεται ο αλγόριθμος της χρυσής τομής.

Θέμα 3 (Μέθοδος Fibonacci)

Λίγα λόγια για τον κώδικα

Όπως και στα προηγούμενα θέματα, έτσι κ εδώ, υλοποιήσαμε στο Matlab τη μέθοδο Fibonacci με τη διαφορά ότι δεν χρησιμοποιήσαμε τον αλγόριθμο του βιβλίου αλλά χρησιμοποιήσαμε την θεωρητική περιγραφή της μεθόδου. Ο κώδικας του αλγόριθμου βρίσκεται στο αρχείο **golden_section_alg.m**.

Συγκεκριμένα, η συνάρτηση **fibonacci_alg(y,a,b,l)** με την οποία υλοποιήσαμε τη μέθοδο Fibonacci, δέχεται ως ορίσματα τη συνάρτηση **f(x)** στην οποία θέλουμε να υπολογίσουμε το ελάχιστο **x***, τα αρχικά όρια αναζήτησης **a** και **b**, και τη μεταβλητή **l** η οποία αντιστοιχεί στο τελικό εύρος. Τρέχοντας τη συνάρτηση του αλγορίθμου για συγκεκριμένες τιμές, μας επιστρέφει τον αριθμό k που είναι ο αριθμός επαναλήψεων, τον αριθμό **n** που αντιστοιχεί στον αριθμό υπολογισμών της αντικειμενικής συνάρτησης και τα εκάστοτε όρια **a(i)** και **b(i)** όπου i=1,2,..,k τα οποία επιστρέφονται υπό τη μορφή μονοδιάστατου πίνακα με k στοιχεία.

Για να βρούμε τον αριθμό επαναλήψεων χρησιμοποιήσαμε την ανισότητα $Fn \leq \frac{l}{b1-a1}$, όπου F ο n-οστός αριθμός Fibonacci. Έτσι από την ανισότητα αυτή βρίσκουμε και τον αριθμό υπολογισμών της αντικειμενικής συνάρτησης ο οποίος αντιστοιχεί με το δείκτη n. Επίσης από την ανισότητα αυτή βρίσκουμε και τον αριθμό επαναλήψεων ο οποίος θα είναι κ=n-2 έτσι ώστε να ισχύει ο αριθμός Fibonacci $F_{(n-k-2)}$ ο οποίος θα χρησιμοποιηθεί αργότερα στην υλοποίηση της μεθόδου.

Εφόσον βρήκαμε το n και το k, θέτουμε:

$x1 = a_{(1)}+(F_{(n-1-1)}/F_{(n-1+1)})*(b_{(1)}-a_{(1)})$

$x2 = a_{(1)}+(F_{(n-1)}/F_{(n-1+1)})*(b_{(1)}-a_{(1)})$

Ακολούθως, αντικαθιστούμε τα x1 και x2 στη συνάρτηση και ελέγχουμε αν f(x1) > f(x2) ή αν f(x1) < f(x2).

Για την $1^η$ περίπτωση:

$a_{(k+1)} = x1$

$b_{(k+1)} = b_{(k)}$

$x1 = x2 \rightarrow y(x1) = y(x2)$

$x2 = a_{(k+1)}+(F_{(n-k-1)}/F_{(n-k)})*(b_{(k+1)}-a_{(k+1)})$

Για την $2^η$ περίπτωση:

$a_{(k+1)} = a_{(k)}$

$b_{(k+1)} = x2$

$x2 = x1 \rightarrow y(x2) = y(x1)$

$x1 = a_{(k+1)}+(F_{(n-k-2)}/F_{(n-k)})*(b_{(k+1)}-a_{(k+1)})$

Εφόσον εκτελεστούν αυτά, αντικαθιστούμε τα νέα x1 και x2 στη συνάρτηση και ελέγχουμε πάλι τις 2 περιπτώσεις. Αυτό γίνεται για κ=n-2 φορές.

Εφαρμογή του αλγόριθμου

Εφόσον τελειώσαμε με την υλοποίηση της μεθόδου και τον κώδικα, τρέχουμε τον αλγόριθμο. Για αρχή, τρέχουμε τον αλγόριθμο για την κάθε συνάρτηση ξεχωριστά με e=0.001, l=0.01 και αρχικά όρια για την $f_1(x)$: a=1, b=6 και για τις $f_2(x)$, $f_3(x)$: a=0, b=6.

Τα τελικά όρια που παίρνουμε για την κάθε συνάρτηση είναι:

$f_1(x)$: $x^* \in [2.5410, 2.5592]$

$f_2(x)$: $x^* \in [2.3410, 2.3508]$

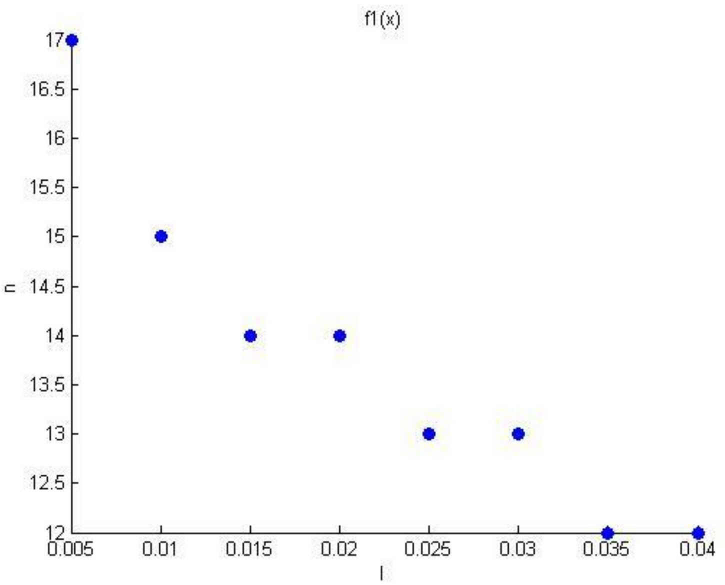
$f_3(x)$: $x^* \in [0.6098, 0.6197]$

Από τα τελικά όρια κάθε συνάρτησης αντίστοιχα παρατηρούμε ότι τηρείται η ανισότητα $b_k - a_k < l$ δηλαδή το τελικό εύρος είναι μικρότερο από το l που διαλέξαμε. Οπότε, ο αλγόριθμος επιστρέφει σωστές τιμές βάσει προδιαγραφής.

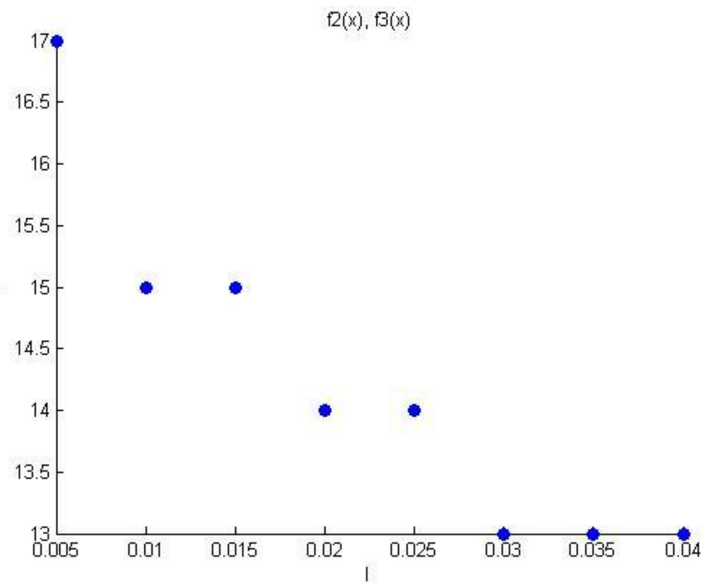
Εφαρμογή του αλγόριθμου για διάφορες τιμές - Παρατηρήσεις και Συμπεράσματα

Α. Για το (α) ερώτημα, τρέχουμε τον αλγόριθμο για την $f_1(x)$ με αρχικά όρια a=1 και b=6. Μεταβάλλουμε το τελικό εύρος αναζήτησης l από 0.005 μέχρι 0.040 και με βήμα 0.005. Αφού τρέξουμε τον αλγόριθμο για τις διάφορες τιμές του l, παρατηρούμε την μεταβολή των υπολογισμών της αντικειμενικής συνάρτησης και δημιουργούμε την αντίστοιχη γραφική παράσταση όπως φαίνεται στο Σχήμα 3.α1.

Ακολουθούμε την ίδια διαδικασία και για τις συναρτήσεις $f_2(x)$ και $f_3(x)$ όμως με αρχικά όρια a=0 και b=6. Οι γραφικές παραστάσεις για την $f_2(x)$ και $f_3(x)$ είναι ίδιες(Σχήμα 3.α2).



Σχήμα 3.α1

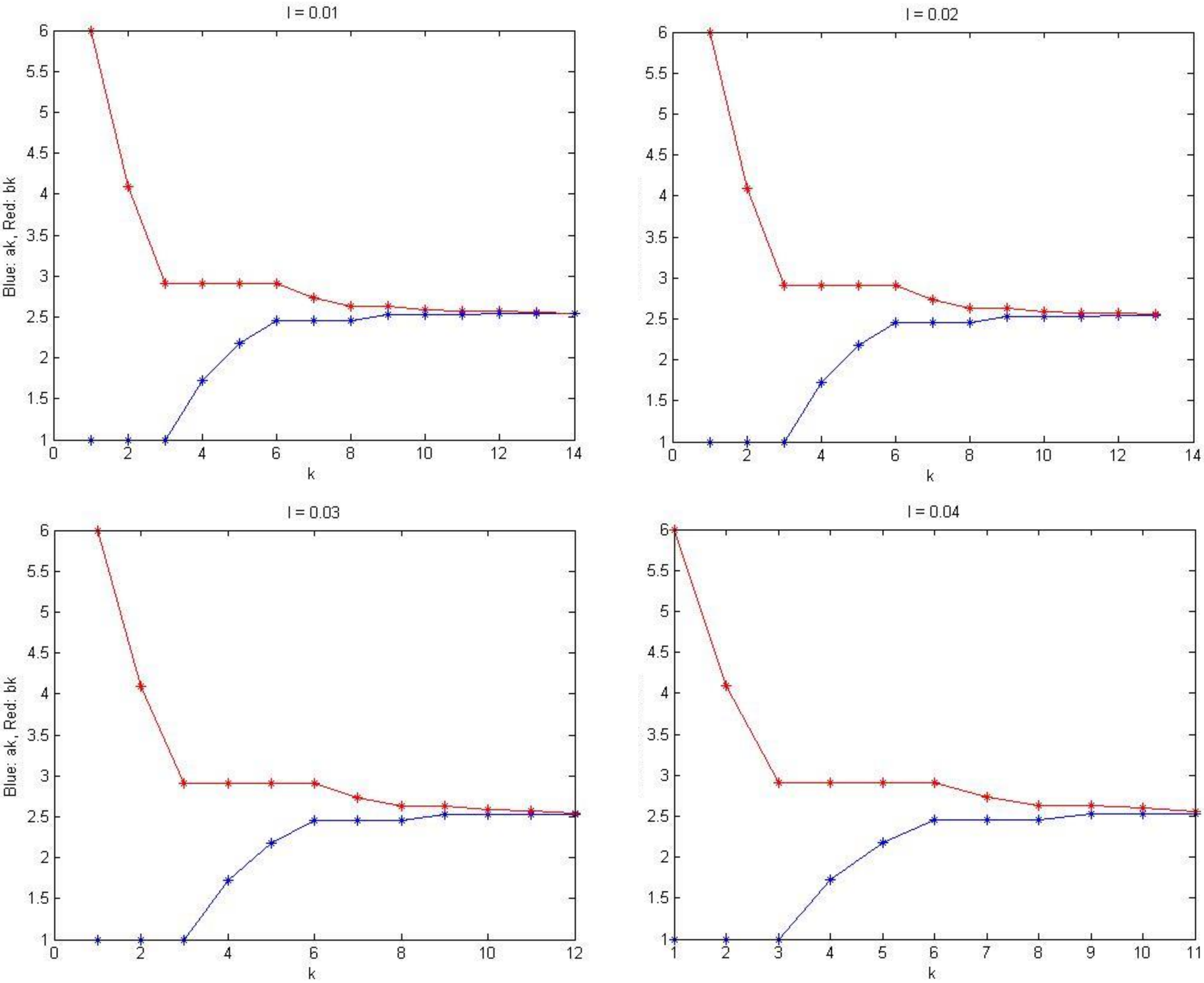


Σχήμα 3.α2

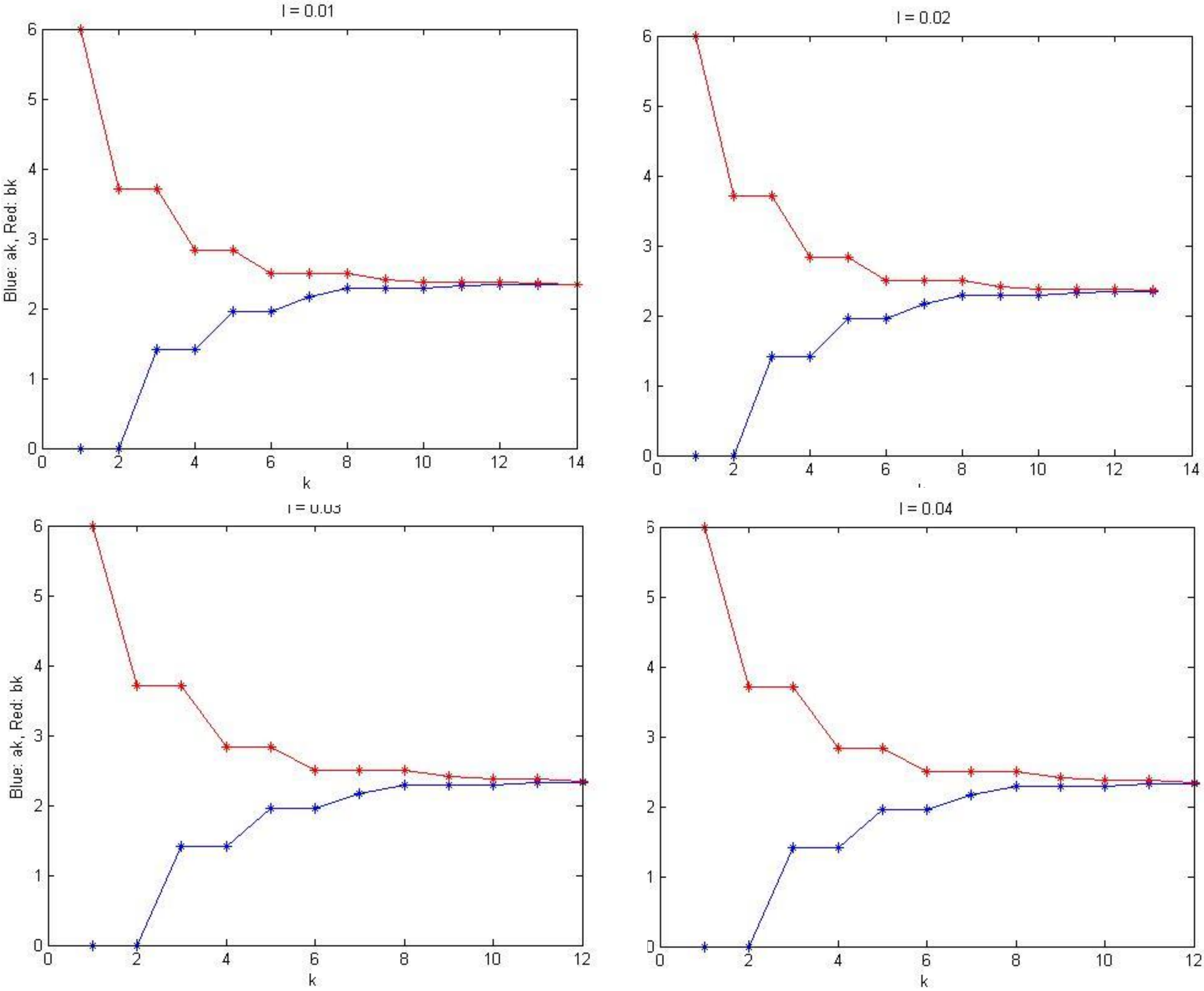
Εδώ παρατηρούμε ότι καθώς αυξάνουμε την ακρίβεια l, οι υπολογισμοί της αντικειμενικής συνάρτησης μειώνονται. Αυτό είναι λογικό διότι αυξάνοντας το l, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την προδιαγεγραμμένη ακρίβεια που αυτό συνεπάγεται και στη μείωση του αριθμού των υπολογισμών της αντικειμενικής συνάρτησης. Καταλήγουμε δηλαδή, στο ίδιο συμπέρασμα με το ερώτημα (β) του 1^{ου} θέματος και (α) του 2^{ου} θέματος, , ωστόσο, ο αριθμός n, είναι μικρότερος από αυτόν της μεθόδου της διχοτόμου και ίδιος με αυτόν της μεθόδου του χρυσού τομέα, για τα αντίστοιχα l.

B. Στο ερώτημα (β) μεταβάλλουμε το l από 0.01 μέχρι 0.04 με βήμα 0.01. Αφού τρέξουμε τον αλγόριθμο για τα διάφορα l, δημιουργούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k, b_k]$ συναρτήσει του βήματος k, για κάθε συνάρτηση ξεχωριστά.

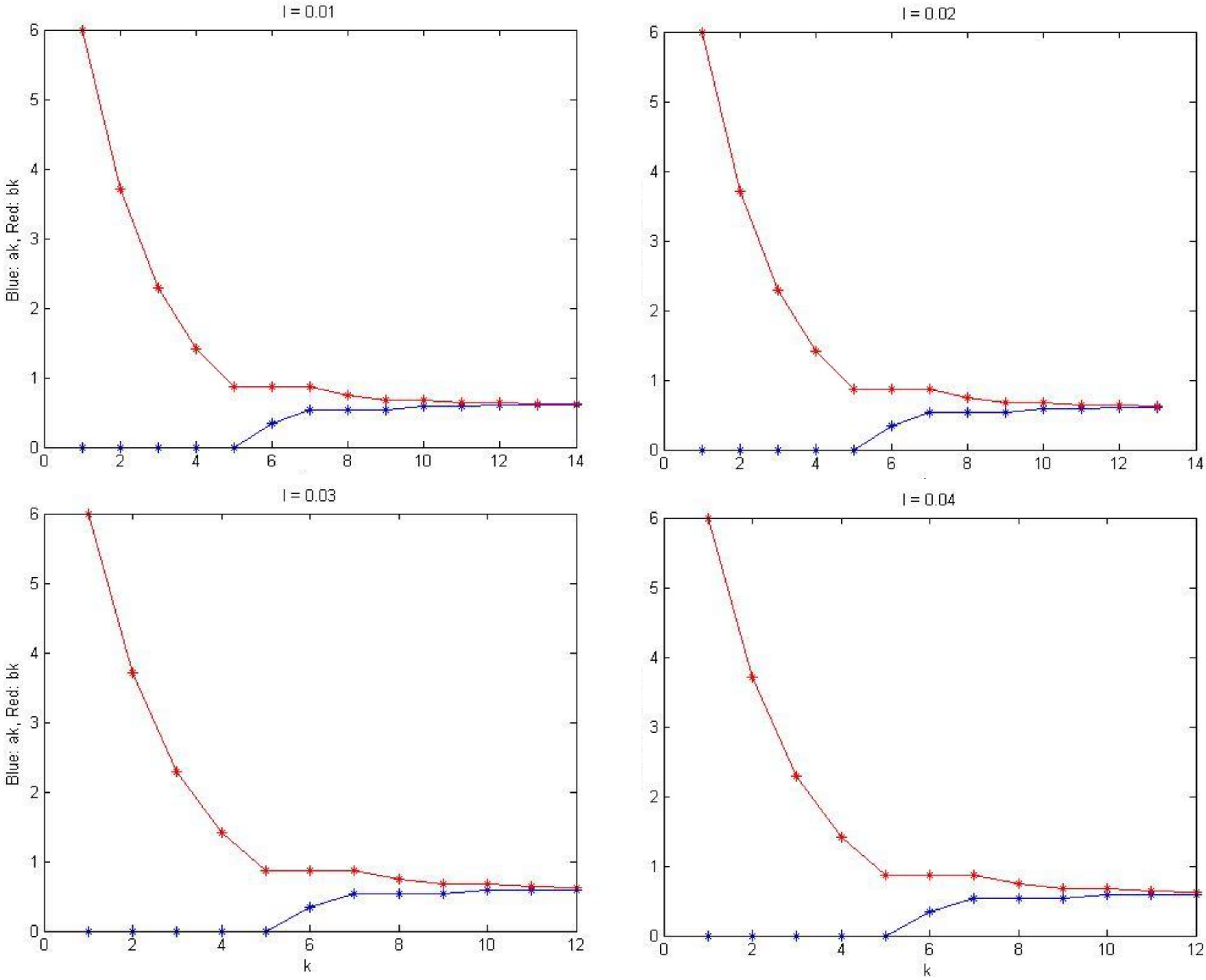
Οι γραφικές παραστάσεις απεικονίζονται παρακάτω. Η $f_1(x)$ απεικονίζεται στο Σχήμα 3.β1 και οι $f_2(x)$ και $f_3(x)$ στα Σχήματα 3.β2 και 3.β3 αντίστοιχα.



Σχήμα 3.β1: $f_1(x)$



Σχήμα 3.β2: $f_2(x)$



Σχήμα 3.β3: $f_3(x)$

Όπως και στις προηγούμενες μεθόδους παρατηρούμε ότι και στις 3 συναρτήσεις, αν αυξήσουμε το l, μειώνεται ο αριθμός επαναλήψεων, διότι αυξάνοντας το l η απαιτούμενη ακρίβεια μειώνεται, επομένως ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την επιλεγμένη ακρίβεια.

Παρατηρούμε ακόμη, ότι οι γραφικές παραστάσεις κάθε συνάρτησης, ταυτίζονται μέχρι ένα συγκεκριμένο κ, ανάλογα με την ακρίβεια. Αυτό συμβαίνει διότι για μεγαλύτερη ακρίβεια (δηλ. μικρότερο l) ο αλγόριθμος απαιτεί περισσότερα βήματα, οπότε σε αυτά τα επιπλέον βήματα που απαιτούνται το διάγραμμα του αλγορίθμου παύει να ταυτίζεται με τα υπόλοιπα διαγράμματα μικρότερης ακρίβειας.

Επίσης, παρατηρούμε ότι για ίδιο l, οι συναρτήσεις έχουν τον ίδιο αριθμό επαναλήψεων με εξαίρεση το l=0.04 της $f_1(x)$ (Σχ. 3.β1) που διαφέρει κατά μια (1) επανάληψη από το αντίστοιχο l των συναρτήσεων $f_2(x)$ και $f_3(x)$. (Σχ. 3.β2 και 3.β3)

Θέμα 4 (Μέθοδος της διχοτόμου με χρήση παραγώγων)

Λίγα λόγια για τον κώδικα

Όπως και στο 1ο θέμα, έτσι κ εδώ, υλοποιήσαμε στο Matlab τη μέθοδο της διχοτόμου αλλά με χρήση παραγώγου, χρησιμοποιώντας τον αλγόριθμο που αναγράφεται στο βιβλίο (5.1.4). Ο κώδικας του αλγόριθμου βρίσκεται στο αρχείο **bisection_der_alg.m**.

Συγκεκριμένα, η συνάρτηση **bisection_der_alg(y,a,b,l)** με την οποία υλοποιήσαμε τον αλγόριθμο της διχοτόμου με παραγώγους, δέχεται ως ορίσματα τη συνάρτηση **f(x)** στην οποία θέλουμε να υπολογίσουμε το ελάχιστο **x***, τα αρχικά όρια αναζήτησης **a** και **b**, και τη μεταβλητή **l** η οποία αντιστοιχεί στο τελικό εύρος. Τρέχοντας τη συνάρτηση του αλγορίθμου για συγκεκριμένες τιμές, μας επιστρέφει τον αριθμό **k** που είναι ο αριθμός επαναλήψεων, τον αριθμό **n** που αντιστοιχεί στον αριθμό υπολογισμών της αντικειμενικής συνάρτησης και τα εκάστοτε όρια **a(i)** και **b(i)** όπου i=1,2,..,k τα οποία επιστρέφονται υπό τη μορφή μονοδιάστατου πίνακα με k στοιχεία.

Επίσης για να βρούμε τον αριθμό υπολογισμών της αντικειμενικής συνάρτησης, χρησιμοποιήσαμε τον τύπο της σελ. 114 του βιβλίου $\left(\frac{1}{2}\right)^n > \frac{l}{b(1)-a(1)}$

Εφαρμογή του αλγόριθμου

Εφόσον τελειώσαμε με την υλοποίηση του αλγόριθμου και τον κώδικα, τρέχουμε τον αλγόριθμο. Για αρχή, τρέχουμε τον αλγόριθμο για την κάθε συνάρτηση ξεχωριστά με l=0.01 και αρχικά όρια για την $f_1(x)$: a=1, b=6 και για τις $f_2(x)$, $f_3(x)$: a=0, b=6.

Τα τελικά όρια που παίρνουμε για την κάθε συνάρτηση είναι:

$f_1(x)$: x* ϵ [2.5479, 2.5527]

$f_2(x)$: x* ϵ [2.3496, 2.3525]

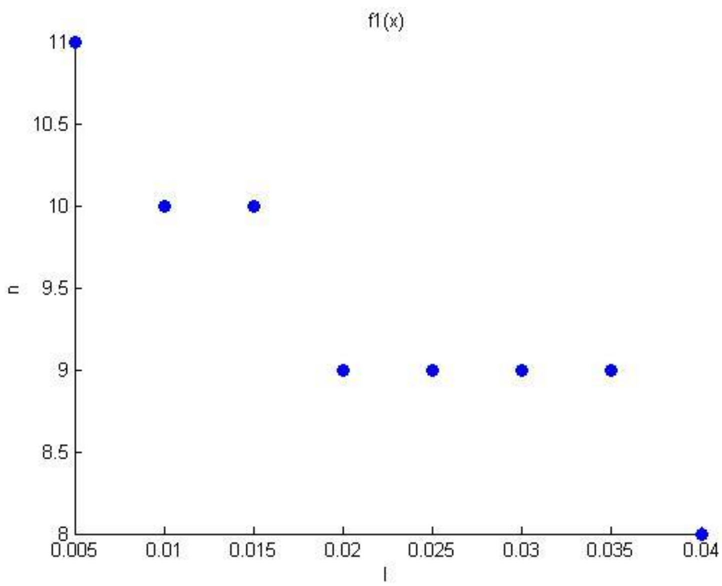
$f_3(x)$: x* ϵ [0.6211, 0.6240]

Από τα τελικά όρια κάθε συνάρτησης αντίστοιχα παρατηρούμε ότι τηρείται η ανισότητα b_k-a_k < l δηλαδή το τελικό εύρος είναι μικρότερο από το l που διαλέξαμε. Οπότε, ο αλγόριθμος επιστρέφει σωστές τιμές βάσει προδιαγραφής

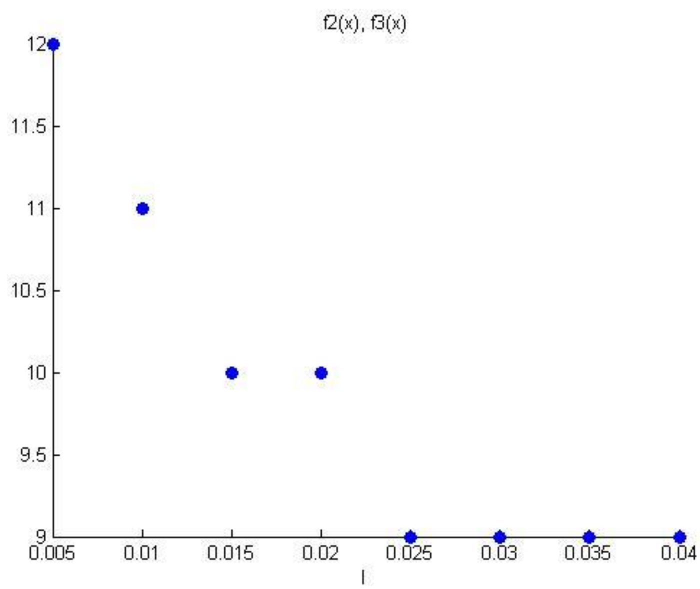
Εφαρμογή του αλγόριθμου για διάφορες τιμές -
Παρατηρήσεις και Συμπεράσματα

Α. Για το (α) ερώτημα, τρέχουμε τον αλγόριθμο για την $f_1(x)$ με αρχικά όρια a=1 και b=6. Μεταβάλλουμε το τελικό εύρος αναζήτησης l από 0.005 μέχρι 0.04 και με βήμα 0.005. Αφού τρέξουμε τον αλγόριθμο για τις διάφορες τιμές του l, παρατηρούμε την μεταβολή των υπολογισμών της αντικειμενικής συνάρτησης και δημιουργούμε την αντίστοιχη γραφική παράσταση όπως φαίνεται στο Σχήμα 4.α1.

Ακολουθούμε την ίδια διαδικασία και για τις συναρτήσεις $f_2(x)$ και $f_3(x)$ όμως με αρχικά όρια a=0 και b=6. Οι γραφικές παραστάσεις για την $f_2(x)$ και $f_3(x)$ είναι ίδιες (Σχήμα 4.α2).



Σχήμα 4.α1

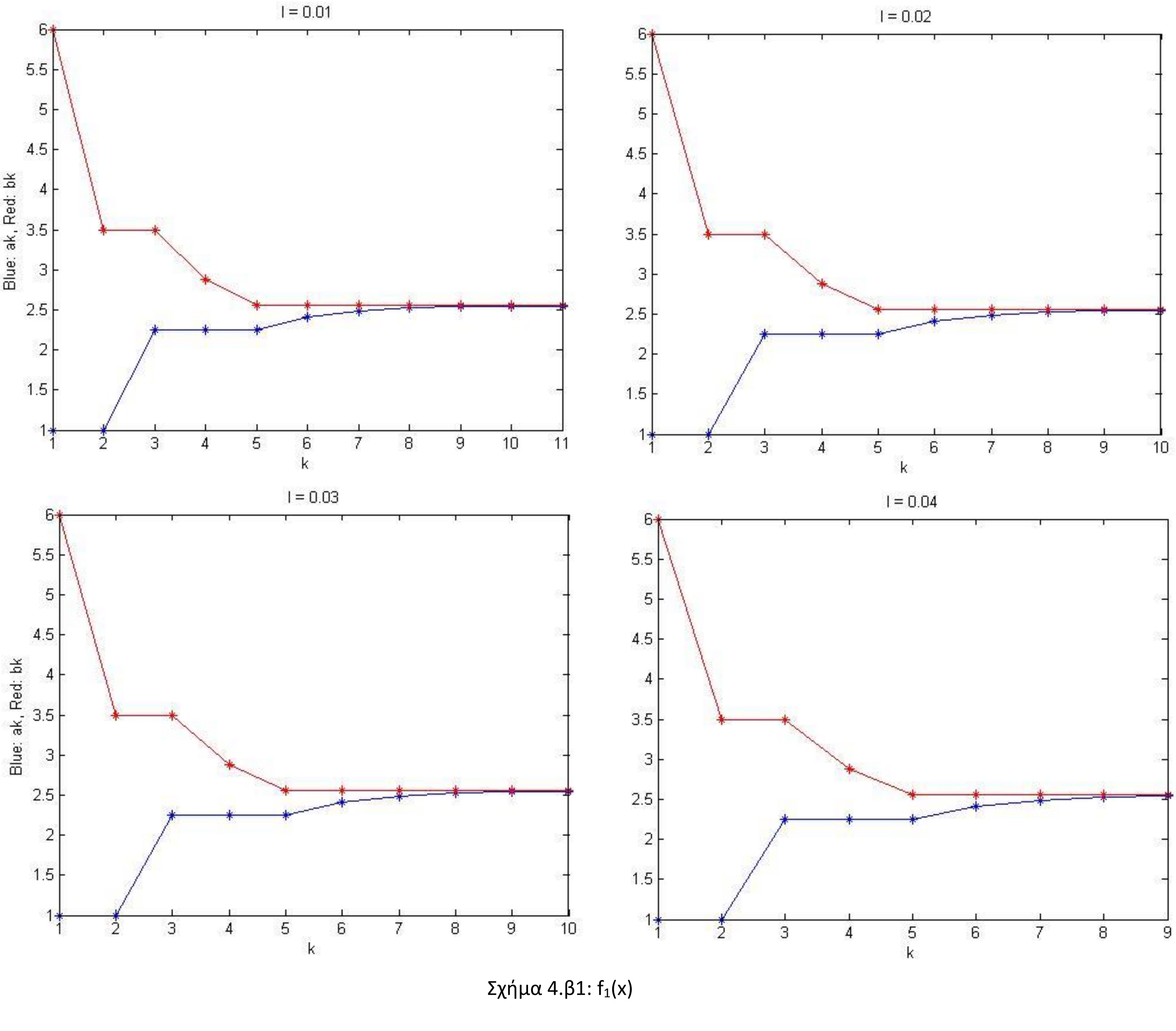


Σχήμα 4.α2

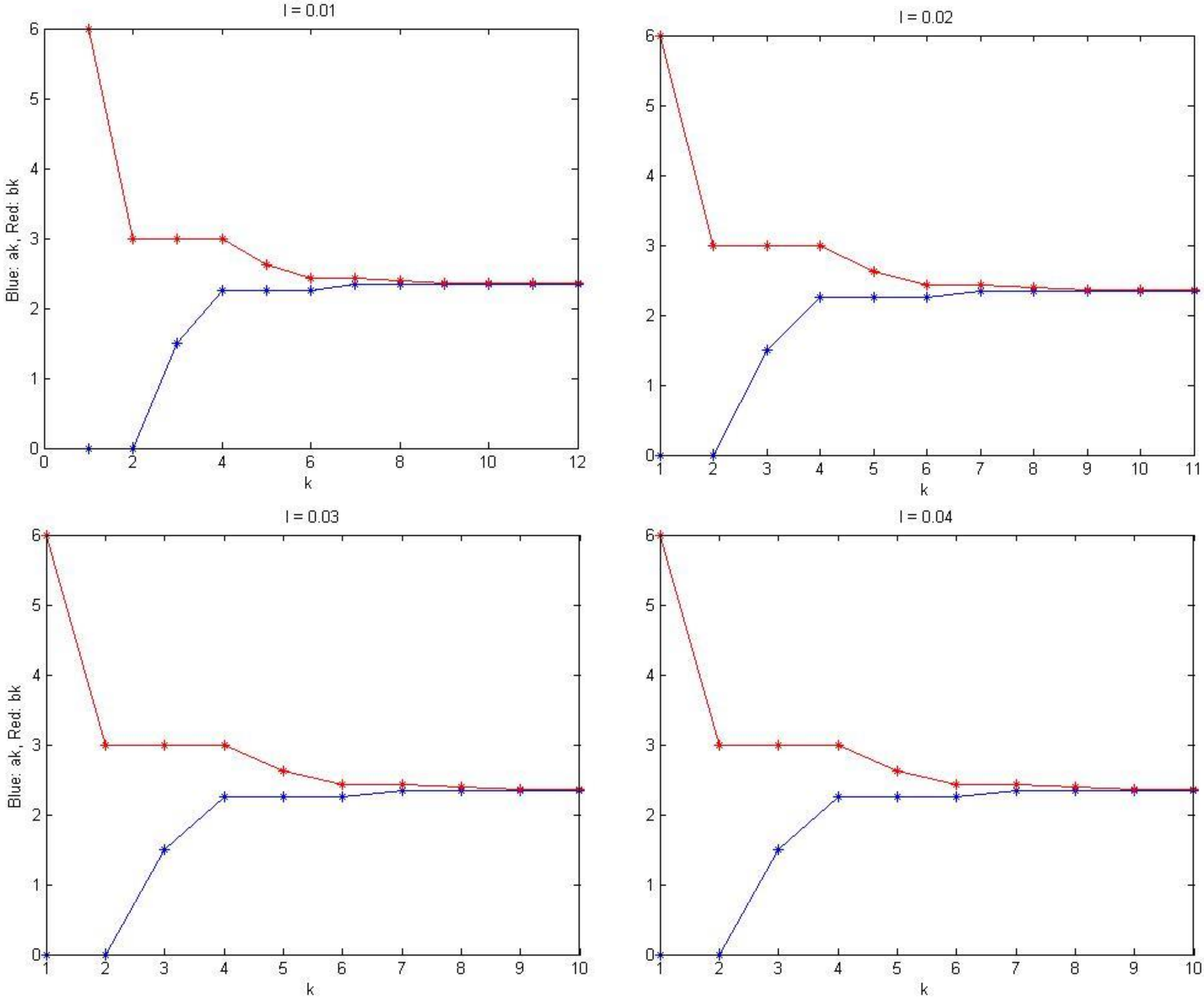
Παρατηρούμε ότι καθώς αυξάνουμε την ακρίβεια l, οι υπολογισμοί της αντικειμενικής συνάρτησης μειώνονται. Αυτό είναι λογικό διότι αυξάνοντας το l, ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την προδιαγεγραμμένη ακρίβεια που αυτό συνεπάγεται και στη μείωση του αριθμού των υπολογισμών της αντικειμενικής συνάρτησης. Καταλήγουμε δηλαδή, στο ίδιο συμπέρασμα με το ερώτημα (β) του $1^{ου}$ θέματος και (α) του $2^{ου}$ και $3^{ου}$ θέματος, ωστόσο ο αριθμός υπολογισμών της αντικειμενικής συνάρτησης σε αυτή την μέθοδο είναι αρκετά χαμηλότερος από τις προηγούμενες μεθόδους για τα αντίστοιχα l, με αποκορύφωμα τη μέθοδο της διχοτόμου όπου ο αριθμός n, είναι σχεδόν διπλάσιος.

Β. Στο ερώτημα (β) μεταβάλλουμε το l από 0.01 μέχρι 0.04 με βήμα 0.01. Αφού τρέξουμε τον αλγόριθμο για τα διάφορα l, δημιουργούμε τις γραφικές παραστάσεις των άκρων του διαστήματος $[a_k, b_k]$ συναρτήσει του βήματος k, για κάθε συνάρτηση ξεχωριστά.

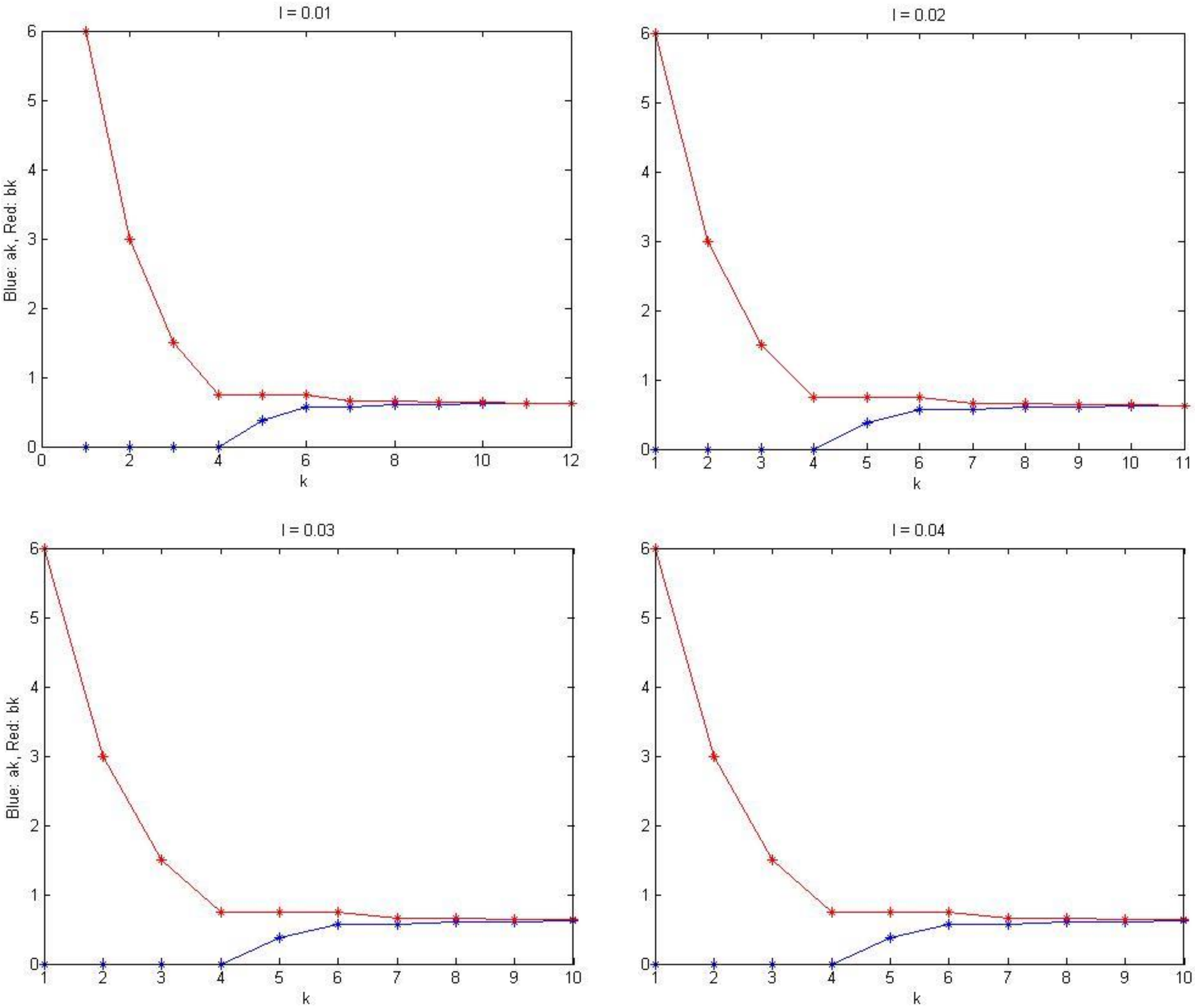
Οι γραφικές παραστάσεις απεικονίζονται παρακάτω. Η $f_1(x)$ απεικονίζεται στο Σχήμα 4.β1 και οι $f_2(x)$ και $f_3(x)$ στα Σχήματα 4.β2 και 4.β3 αντίστοιχα.



Σχήμα 4.β1: $f_1(x)$



Σχήμα 4.β2: $f_2(x)$



Σχήμα 4.β3: $f_3(x)$

Όπως και στις προηγούμενες μεθόδους παρατηρούμε ότι και στις 3 συναρτήσεις, αν αυξήσουμε το l, μειώνεται ο αριθμός επαναλήψεων, διότι αυξάνοντας το l η απαιτούμενη ακρίβεια μειώνεται, επομένως ο αλγόριθμος χρειάζεται λιγότερες επαναλήψεις για να πετύχει την επιλεγμένη ακρίβεια.

Παρατηρούμε ακόμη, ότι οι γραφικές παραστάσεις κάθε συνάρτησης, ταυτίζονται μέχρι ένα συγκεκριμένο κ, ανάλογα με την ακρίβεια. Αυτό συμβαίνει διότι για μεγαλύτερη ακρίβεια (δηλ. μικρότερο l) ο αλγόριθμος απαιτεί περισσότερα βήματα, οπότε σε αυτά τα επιπλέον βήματα που απαιτούνται το διάγραμμα του αλγορίθμου παύει να ταυτίζεται με τα υπόλοιπα διαγράμματα μικρότερης ακρίβειας.

Επίσης, παρατηρούμε ότι ο αλγόριθμος συγκλίνει νωρίτερα στα τελικά όρια συγκριτικά με τις προηγούμενες μεθόδους.

Γενικό Συμπέρασμα

Τρέχοντας όλους τους αλγόριθμους για διάφορες τιμές, καταλήγουμε σε κάποια συμπεράσματα τα οποία αποδεικνύονται από τις γραφικές παραστάσεις που παραθέσαμε παραπάνω αλλά και από τη θεωρία του βιβλίου.

Βασιζόμενοι στις γραφικές παραστάσεις που προκύπτουν για τον κάθε αλγόριθμο, βλέπουμε ότι ο αλγόριθμος της διχοτόμου με παράγωγο, συγκριτικά με τους υπόλοιπους αλγόριθμους που υλοποιήσαμε, επιτυγχάνει την εύρεση του τελικού διαστήματος με την ζητούμενη ακρίβεια, με λίγες επαναλήψεις αλλά και με μικρό αριθμό υπολογισμών της αντικειμενικής συνάρτησης. Καταλήγουμε λοιπόν, στο συμπέρασμα ότι ο αλγόριθμος της διχοτόμου με παράγωγο είναι ο πιο αποδοτικός από τους υπόλοιπους 3.

Από την άλλη μεριά, ο αλγόριθμος της διχοτόμου (χωρίς παράγωγο), παρόλο που και αυτός καταλήγει σε ένα αποδεκτό τελικό διάστημα, ωστόσο δεν είναι αρκετά αποδοτικός όσο είναι οι υπόλοιποι αλγόριθμοι διότι σε κάθε επανάληψη, χρειάζεται ο υπολογισμός της αντικειμενικής συνάρτησης 2 φορές, με συνέπεια ο υπολογισμός του τελικού διαστήματος να απαιτεί περισσότερο χρόνο.

Με την ίδια λογική καταλήγουμε στην εξής κατάταξη των αλγορίθμων, βασιζόμενοι στην απόδοση τους, με $1^{ον}$ τον αποδοτικότερο αλγόριθμο:

1. Μέθοδος της διχοτόμου με χρήση παραγώγου
2. Μέθοδος Fibonacci
3. Μέθοδος του Χρυσού τομέα
4. Μέθοδος της διχοτόμου

Επίσης, αξίζει να σημειωθεί ότι για μικρό l (μεγάλη ακρίβεια), όλοι οι αλγόριθμοι κατέληξαν σε ένα καλό τελικό διάστημα έχοντας περίπου τα ίδια αποτελέσματα, άσχετα με το χρόνο εκτέλεσης τους.