Τεχνικές Βελτιστοποίησης

<u>Πρώτη εργαστηριακή άσκηση</u>

-

<u>Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής σε δοσμένο διάστημα</u>

Κωνσταντίνος Κωνσταντινίδης,

ΑΕΜ: 9162 | *konkonstantinidis@ece.auth.gr*

Το παρόν είναι μία αναφορά που παρουσιάζει και σχολιάζει τα ζητούμενα διαγράμματα και έπειτα καταλήγει στα συμπεράσματα της έρευνας.

Οι συναρτήσεις προς ελαχιστοποίηση είναι οι:

- $f_1(x) = (x-2)^2 - \sin(x+3)$
- $f_2(x) = e^{-5x} + (x+2)\cos^2(0.5x)$
- $f_3(x) = x^2\sin(x+2) - (x+1)^2$

στο δοσμένο διάστημα [2,5].

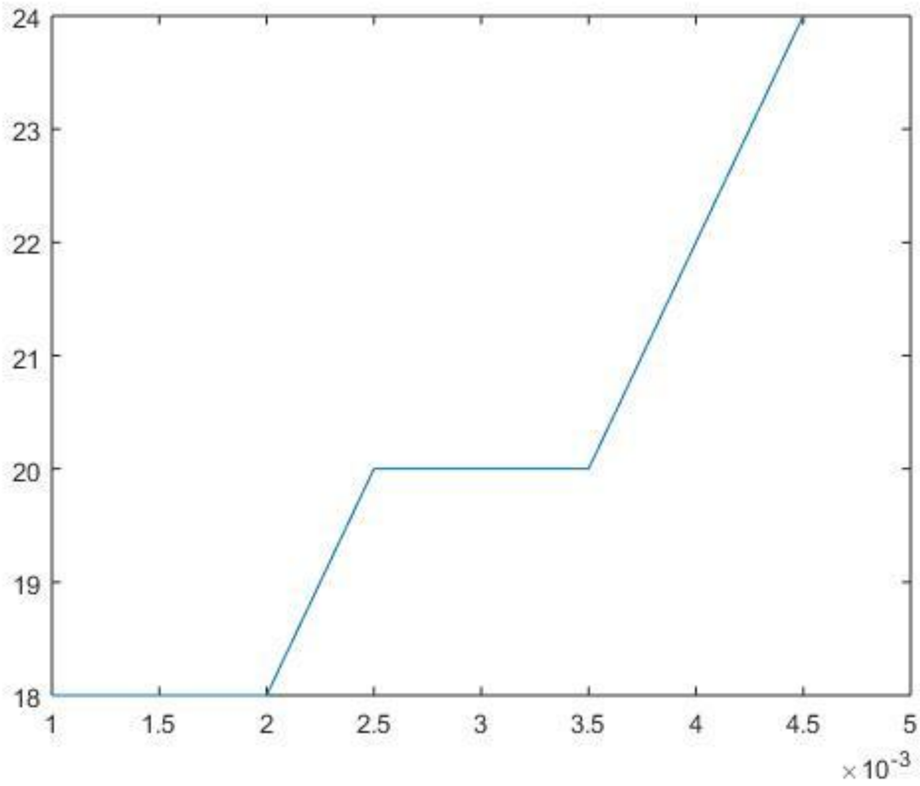
Θέμα 1

Η μέθοδος της διχοτόμου

Έχει παραδοθεί η υλοποίηση της μεθόδου της διχοτόμου στο Matlab. Ο κώδικας είναι αρκετά απλός, δεν χρειάζονται επιπλέον σχόλια ως προς την λειτουργία του.

Κρατώντας σταθερό το τελικό εύρος αναζήτησης $\lambda = 0.01$, μελετάμε το πόσες φορές υπολογίζεται τιμή της συνάρτησης f για διάφορες τιμές της σταθεράς ε (απόσταση από τη διχοτόμο) (const_e στον κώδικα). Η σταθερά ε μεταβάλλεται από 0.001 με βήμα step = 0.0005 έως την τιμή λ/2 – step = 0.0045, καθώς πρέπει να είναι $\varepsilon<\lambda/2$, ειδάλλως η μέθοδος δεν συγκλίνει.

Ακολουθεί μία γραφική παράσταση από τις τιμές που προκύπτουν (όλες οι συναρτήσεις f(x) έδωσαν την ίδια γραφική παράσταση για τη συγκεκριμένη μελέτη) .

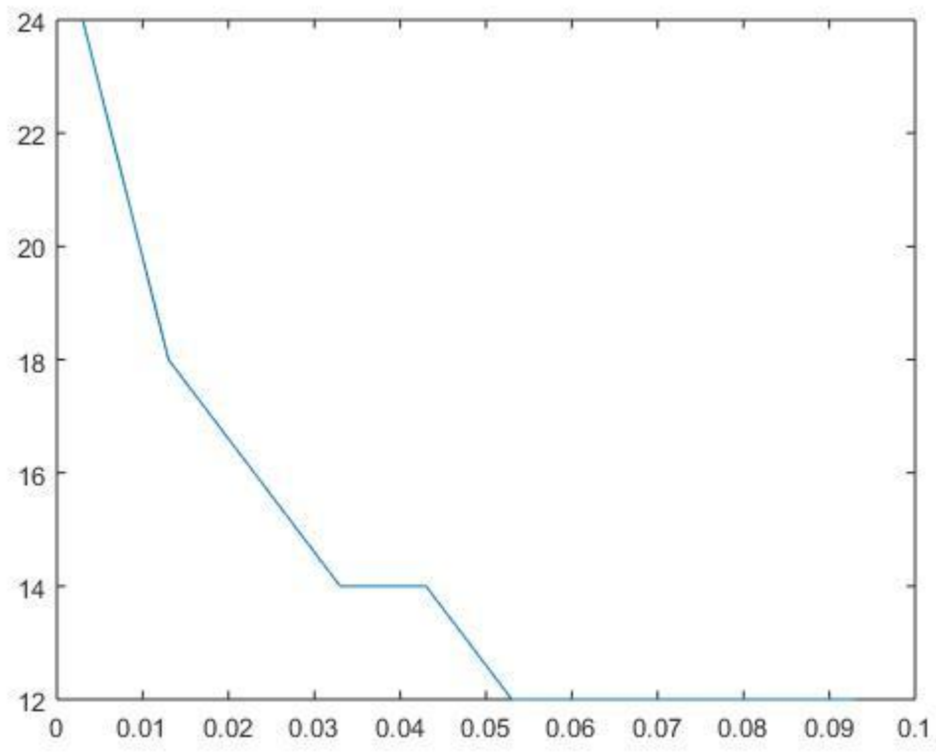


Γραφική παράσταση 1, (κλήσεις f(x) , ε) = (y,x) .

Όπως βλέπουμε από την γραφική παράσταση λ, όσο αυξάνεται η τιμή της σταθεράς ε, τόσο αυξάνονται και οι κλήσεις της εκάστοτε συνάρτησης f(x). Αυτό συμβαίνει γιατί με την αύξηση της σταθεράς ε, τα $x_1(k)$ και $x_2(k)$ υπολογίζονται πιο μακριά από τη διχοτόμο του διαστήματος [a(k),b(k)] και άρα κατά συνέπεια πιο κοντά στα άκρα του, με αποτέλεσμα το διάστημα [a(k+1),b(k+1)] που προκύπτει μετά από κάθε επανάληψη k να είναι μεγαλύτερο. Έτσι, χρειάζονται παραπάνω βήματα του αλγορίθμου για να φτάσουμε στο επιθυμητό εύρος διαστήματος λ, άρα και περισσότερες κλήσεις της εκάστοτε συνάρτησης f(x).

Κρατώντας σταθερό το ε = 0.001, μελετάμε το πόσες φορές υπολογίζεται τιμή της συνάρτησης f για διάφορες τιμές του τελικού εύρους διαστήματος λ. Το τελικό εύρος διαστήματος λ μεταβάλλεται από 0.003 μέχρι 0.1, με step = 0.01.

Ακολουθεί μία γραφική παράσταση από τις τιμές που προκύπτουν (όλες οι συναρτήσεις f(x) έδωσαν την ίδια γραφική παράσταση για τη συγκεκριμένη μελέτη) .

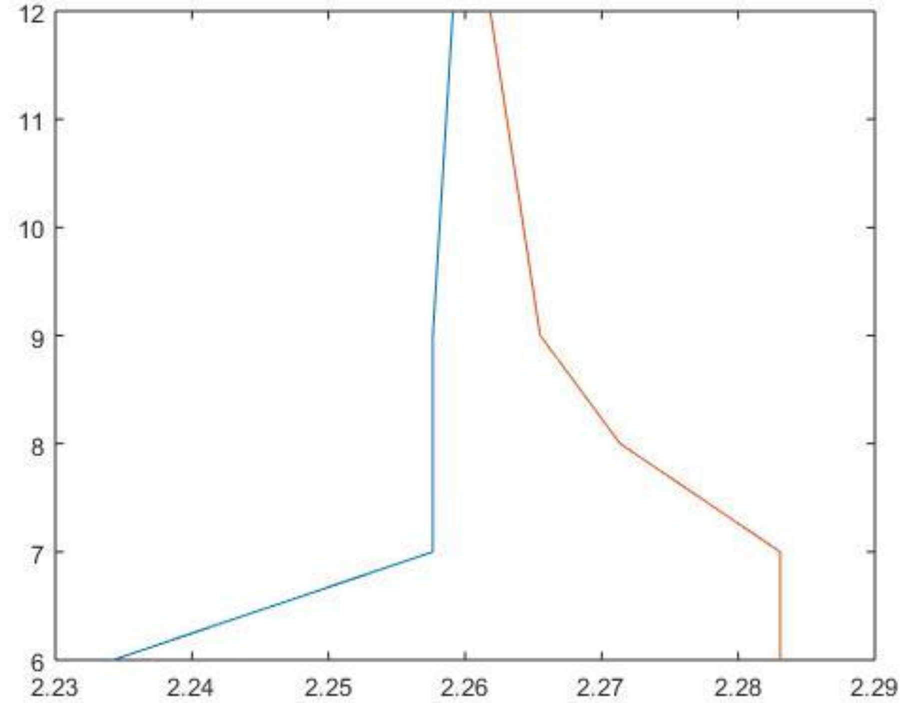


Γραφική παράσταση 2, (κλήσεις f(x) , λ) = (y,x) .

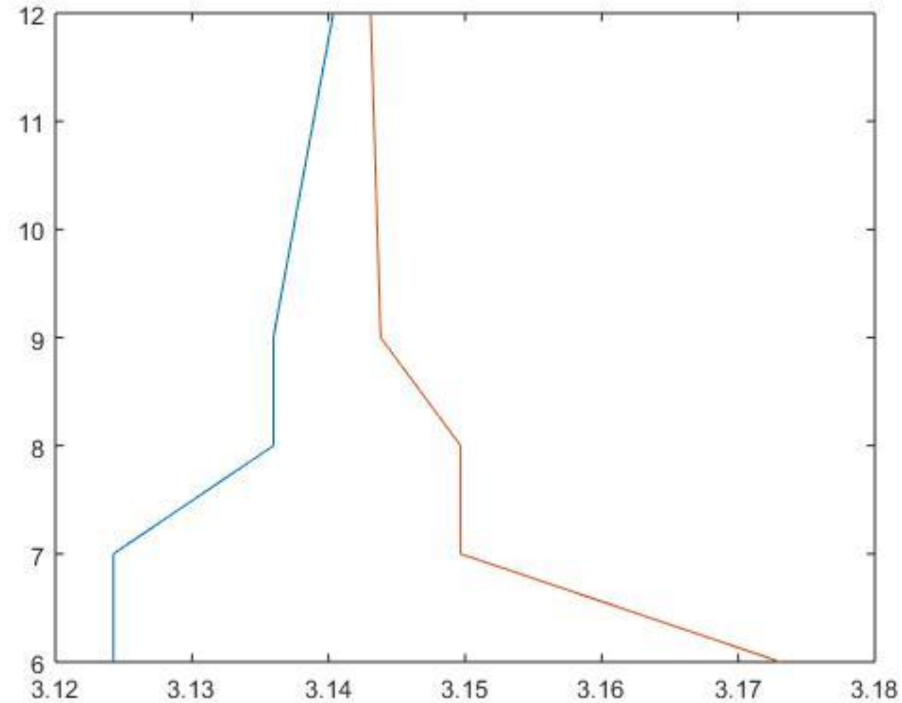
Από την γραφική παράσταση 2, έχουμε πως όσο αυξάνεται το τελικό εύρος διαστήματος λ, τόσο μειώνονται οι κλήσεις της εκάστοτε συνάρτησης f(x). Αυτό συμβαίνει επειδή όσο αυξάνουμε το τελικό εύρος διαστήματος λ, τόσα λιγότερα βήματα – διαμερίσεις θα χρειαστούν να γίνουν στο αρχικό διάστημα αναζήτησης ώστε να φτάσουμε σε λύση επιθυμητής ακρίβειας. Ως εκ τούτου, θα έχουμε και λιγότερες κλήσεις της συνάρτησης.

Τέλος, μελετάμε τα τελικά άκρα του διαστήματος $[a_k , b_k]$ συναρτήσει του δείκτη βήματος k, για τις ίδιες τιμές επιλογής του τελικού εύρους αναζήτησης λ με πάνω.

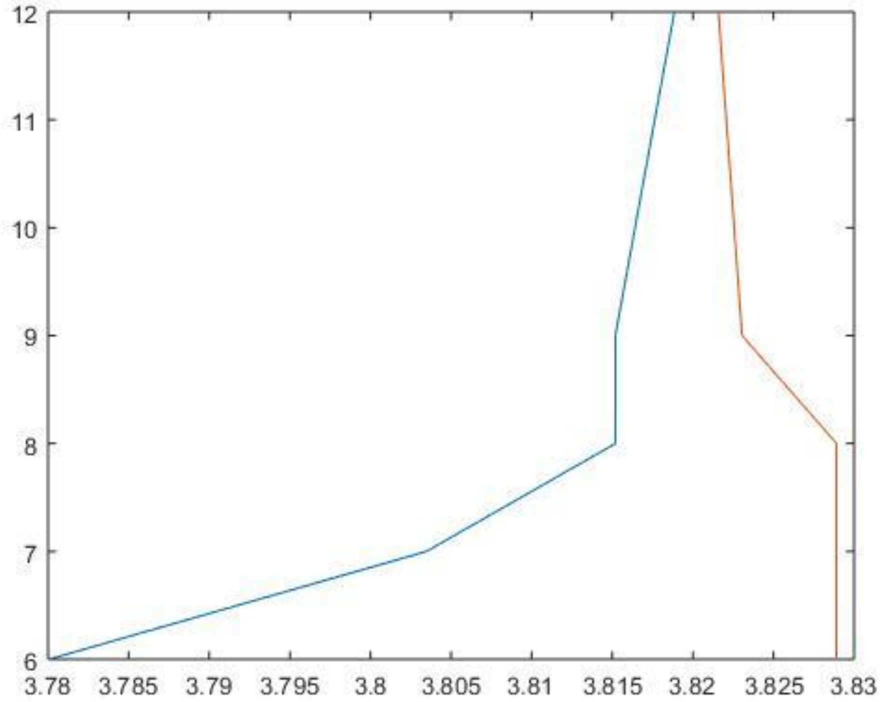
Ακολουθούν τρία διαγράμματα, ένα για κάθε συνάρτηση, που παρουσιάζουν τις γραφικές παραστάσεις (k , a_k) και (k , b_k).



Διάγραμμα 1, f_1(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 2, f_2(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 3, $f_3(x)$: (k , a(k)) , (k , b(k)) = (y,x) .

Όπως βλέπουμε στα διαγράμματα 1,2 και 3, και όπως λογικά περιμέναμε, όσο παραπάνω μειώνεται το εύρος του τελικού διαστήματος [a(k),b(k)], χρειάζονται παραπάνω βήματα του αλγορίθμου για να το φτάσουμε.

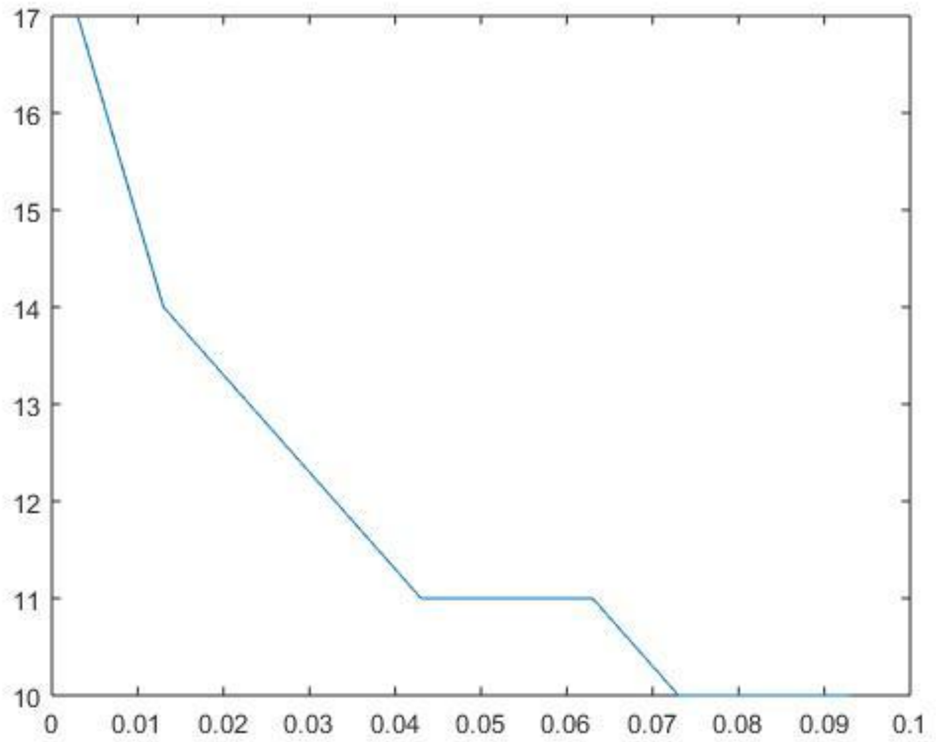
Θέμα 2

Η μέθοδος του Χρυσού Τομέα

Έχει παραδοθεί η υλοποίηση της μεθόδου του Χρυσού Τομέα στο Matlab. Ο κώδικας είναι αρκετά απλός, δεν χρειάζονται επιπλέον σχόλια ως προς την λειτουργία του.

Μελετάμε το πόσες φορές υπολογίζεται τιμή της συνάρτησης f για διάφορες τιμές του τελικού εύρους διαστήματος λ. Το τελικό εύρος διαστήματος λ μεταβάλλεται από 0.003 μέχρι 0.1, με step = 0.01.

Ακολουθεί μία γραφική παράσταση από τις τιμές που προκύπτουν (όλες οι συναρτήσεις f(x) έδωσαν την ίδια γραφική παράσταση για τη συγκεκριμένη μελέτη) .

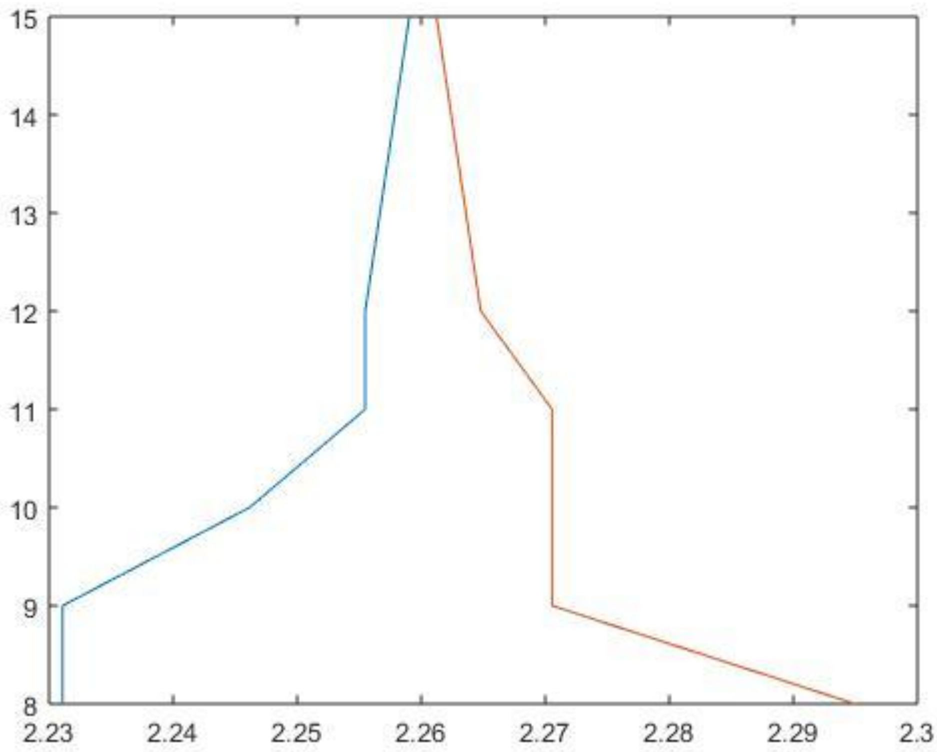


Γραφική παράσταση 3, (κλήσεις f(x) , λ) = (y,x) .

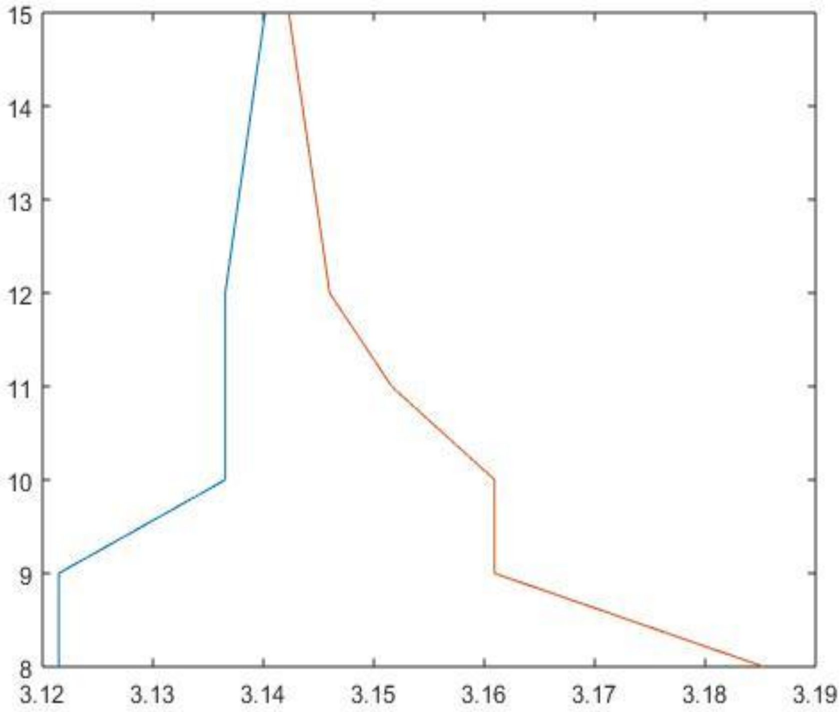
Από την γραφική παράσταση 3, έχουμε πως όσο αυξάνεται το τελικό εύρος διαστήματος λ, τόσο μειώνονται οι κλήσεις της εκάστοτε συνάρτησης f(x). Αυτό συμβαίνει επειδή όσο αυξάνουμε το τελικό εύρος διαστήματος λ, τόσα λιγότερα βήματα – διαμερίσεις θα χρειαστούν να γίνουν στο αρχικό διάστημα αναζήτησης ώστε να φτάσουμε σε λύση επιθυμητής ακρίβειας. Ως εκ τούτου, θα έχουμε και λιγότερες κλήσεις της συνάρτησης.

Τέλος, μελετάμε τα τελικά άκρα του διαστήματος [a_k , b_k] συναρτήσει του δείκτη βήματος k, για τις ίδιες τιμές επιλογής του τελικού εύρους αναζήτησης λ με πάνω.

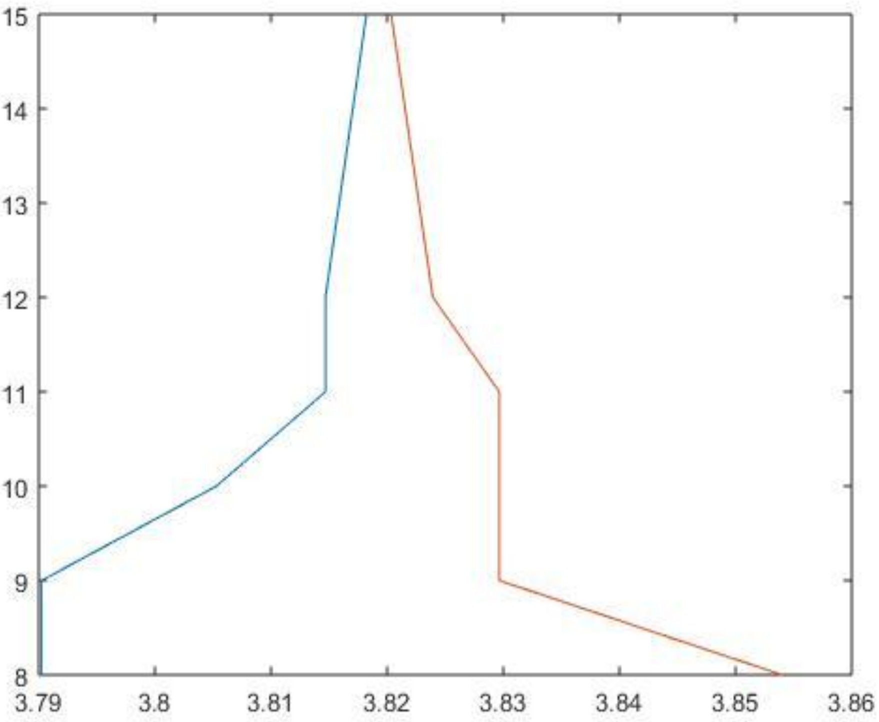
Ακολουθούν τρία διαγράμματα, ένα για κάθε συνάρτηση, που παρουσιάζουν τις γραφικές παραστάσεις (k , a_k) και (k , b_k).



Διάγραμμα 4, f1(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 5, $f_2(x)$: (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 6, $f_3(x)$: (k , a(k)) , (k , b(k)) = (y,x) .

Όπως βλέπουμε στα διαγράμματα 4,5 και 6, και όπως λογικά περιμέναμε, όσο παραπάνω μειώνεται το εύρος του τελικού διαστήματος [a(k),b(k)], χρειάζονται παραπάνω βήματα του αλγορίθμου για να το φτάσουμε.

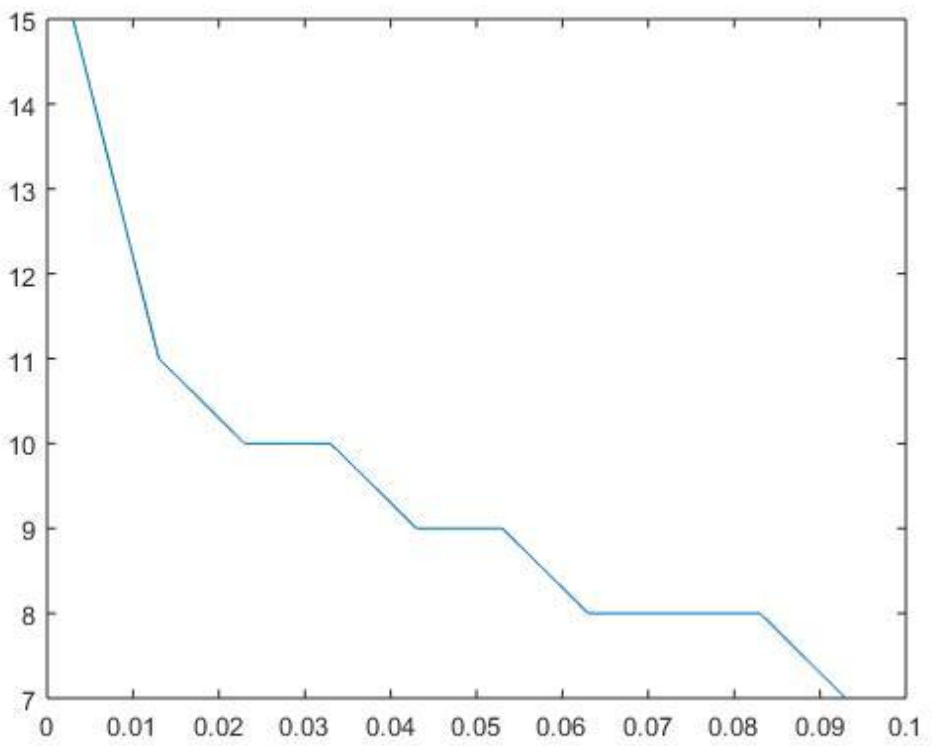
Θέμα 3

<u>Η μέθοδος Fibonacci</u>

Έχει παραδοθεί η υλοποίηση της μεθόδου του Fibonacci στο Matlab. Ο κώδικας έχει επάνω ενσωματωμένα σχόλια που εξηγούν ικανοποιητικά τη λειτουργία του, δεν χρειάζονται επιπλέον.

Μελετάμε το πόσες φορές υπολογίζεται τιμή της συνάρτησης f για διάφορες τιμές του τελικού εύρους διαστήματος l. Το τελικό εύρος διαστήματος l μεταβάλλεται από 0.003 μέχρι 0.1, με step = 0.01.

Ακολουθεί μία γραφική παράσταση από τις τιμές που προκύπτουν (όλες οι συναρτήσεις f(x) έδωσαν την ίδια γραφική παράσταση για τη συγκεκριμένη μελέτη) .

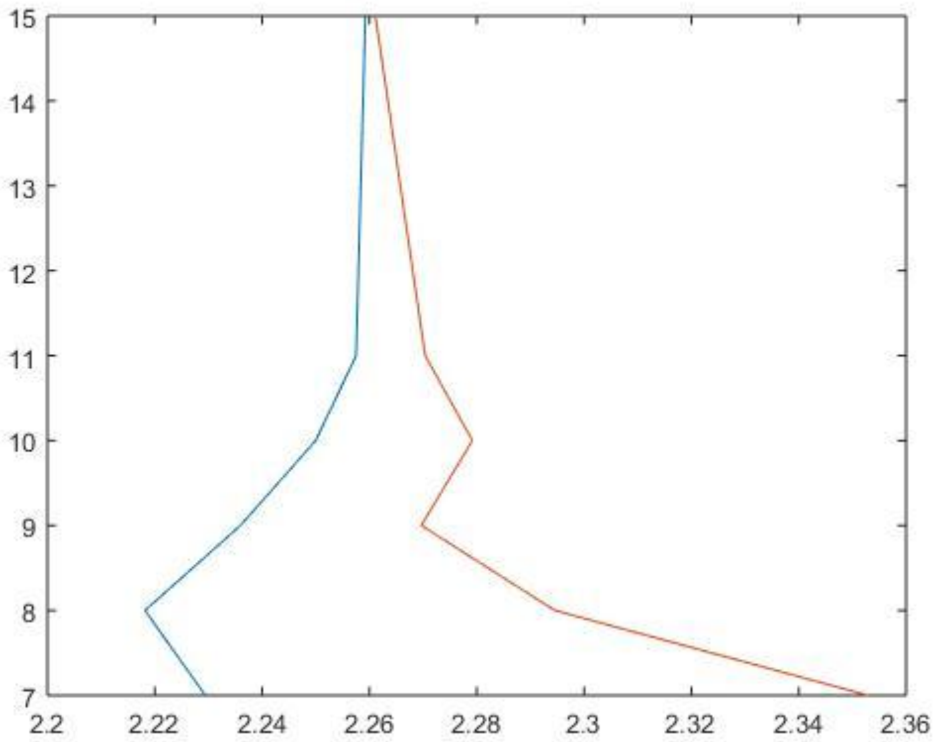


Γραφική παράσταση 4, (κλήσεις f(x) , λ) = (y,x) .

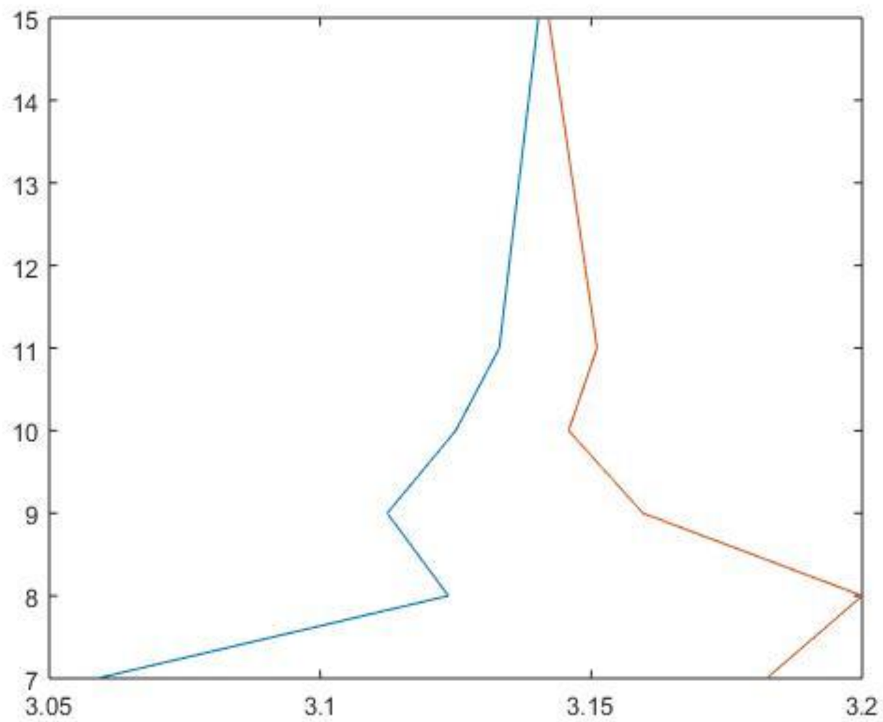
Από την γραφική παράσταση 4, έχουμε πως όσο αυξάνεται το τελικό εύρος διαστήματος λ, τόσο μειώνονται οι κλήσεις της εκάστοτε συνάρτησης f(x). Αυτό συμβαίνει επειδή όσο αυξάνουμε το τελικό εύρος διαστήματος λ, τόσα λιγότερα βήματα – διαμερίσεις θα χρειαστούν να γίνουν στο αρχικό διάστημα αναζήτησης ώστε να φτάσουμε σε λύση επιθυμητής ακρίβειας. Ως εκ τούτου, θα έχουμε και λιγότερες κλήσεις της συνάρτησης.

Τέλος, μελετάμε τα τελικά άκρα του διαστήματος $[a_k , b_k]$ συναρτήσει του δείκτη βήματος k, για τις ίδιες τιμές επιλογής του τελικού εύρους αναζήτησης l με πάνω.

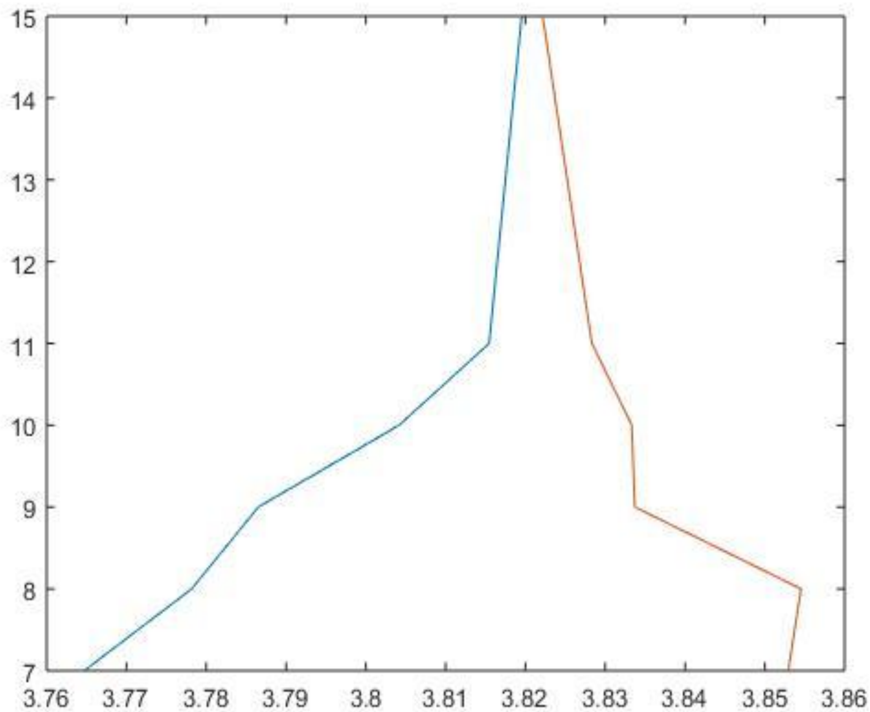
Ακολουθούν τρία διαγράμματα, ένα για κάθε συνάρτηση, που παρουσιάζουν τις γραφικές παραστάσεις (k , a_k) και (k , b_k).



Διάγραμμα 7, f1(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 8, f2(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 9, f3(x): (k , a(k)) , (k , b(k)) = (y,x) .

Όπως βλέπουμε στα διαγράμματα 7,8 και 9, και όπως λογικά περιμέναμε, όσο παραπάνω μειώνεται το εύρος του τελικού διαστήματος [a(k),b(k)], χρειάζονται παραπάνω βήματα του αλγορίθμου για να το φτάσουμε.

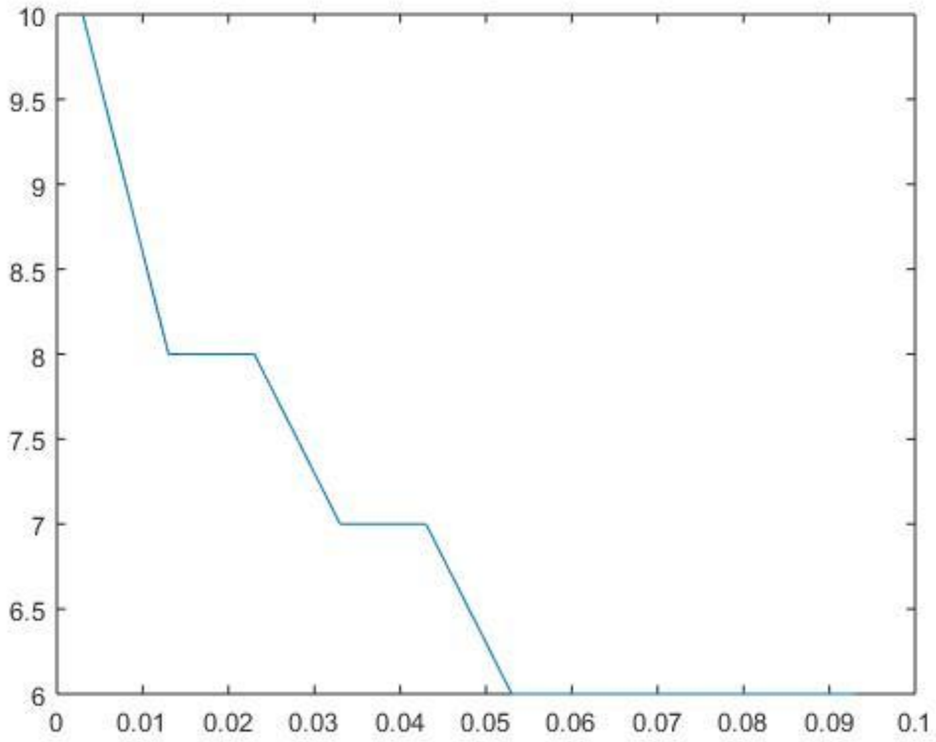
Θέμα 4

Η μέθοδος της Διχοτόμου με χρήση παραγώγου

Έχει παραδοθεί η υλοποίηση της μεθόδου της διχοτόμου με χρήση παραγώγου στο Matlab. Ο κώδικας είναι αρκετά απλός, δεν χρειάζονται επιπλέον σχόλια ως προς την λειτουργία του.

Μελετάμε το πόσες φορές υπολογίζεται τιμή της συνάρτησης f για διάφορες τιμές του τελικού εύρους διαστήματος l. Το τελικό εύρος διαστήματος l μεταβάλλεται από 0.003 μέχρι 0.1, με step = 0.01.

Ακολουθούν τρεις γραφικές παραστάσεις από τις τιμές που προκύπτουν, μία για κάθε συνάρτηση f(x).

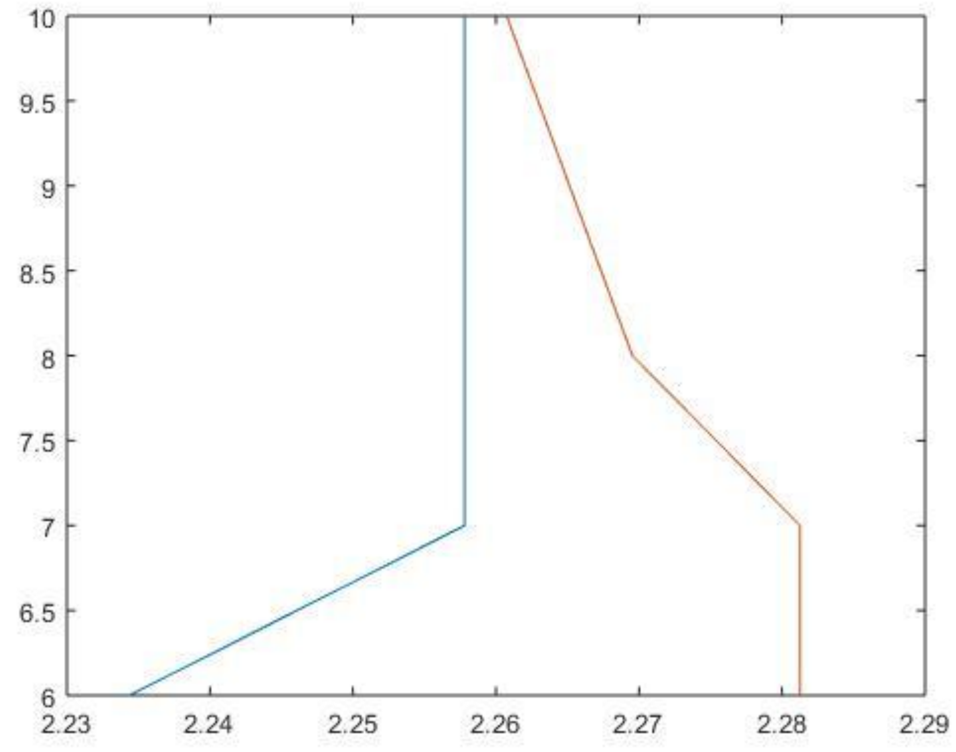


Γραφική παράσταση 5, (κλήσεις f(x) , λ) = (y,x) .

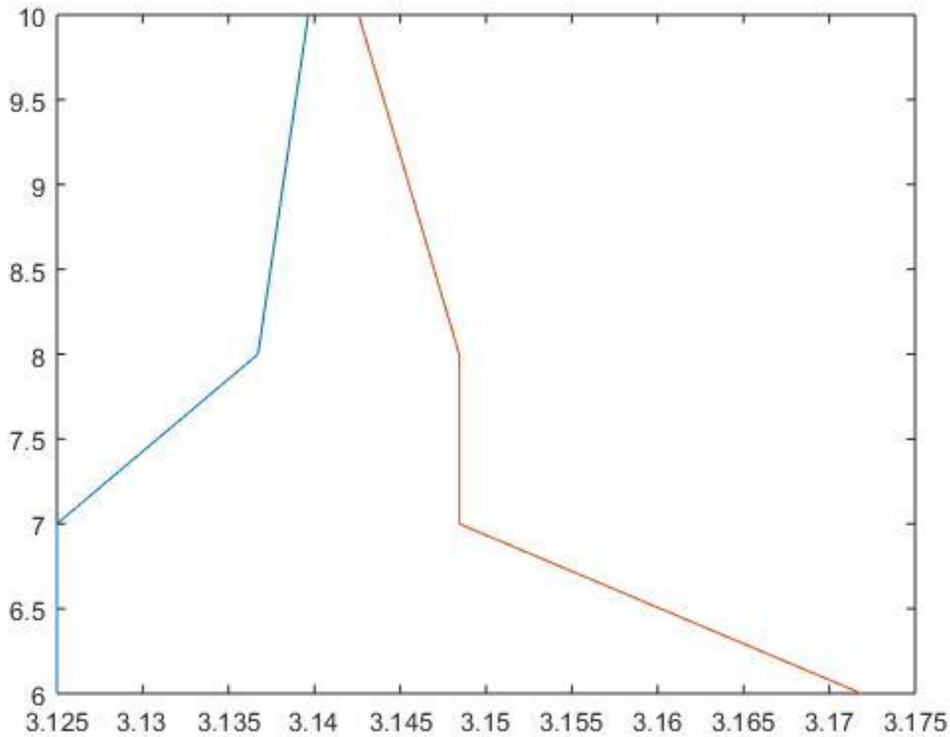
Από την γραφική παράσταση 5, έχουμε πως όσο αυξάνεται το τελικό εύρος διαστήματος λ, τόσο μειώνονται οι κλήσεις της εκάστοτε συνάρτησης f(x). Αυτό συμβαίνει επειδή όσο αυξάνουμε το τελικό εύρος διαστήματος λ, τόσα λιγότερα βήματα – διαμερίσεις θα χρειαστούν να γίνουν στο αρχικό διάστημα αναζήτησης ώστε να φτάσουμε σε λύση επιθυμητής ακρίβειας. Ως εκ τούτου, θα έχουμε και λιγότερες κλήσεις της συνάρτησης.

Τέλος, μελετάμε τα τελικά άκρα του διαστήματος [a_k , b_k] συναρτήσει του δείκτη βήματος k, για τις ίδιες τιμές επιλογής του τελικού εύρους αναζήτησης l με πάνω.

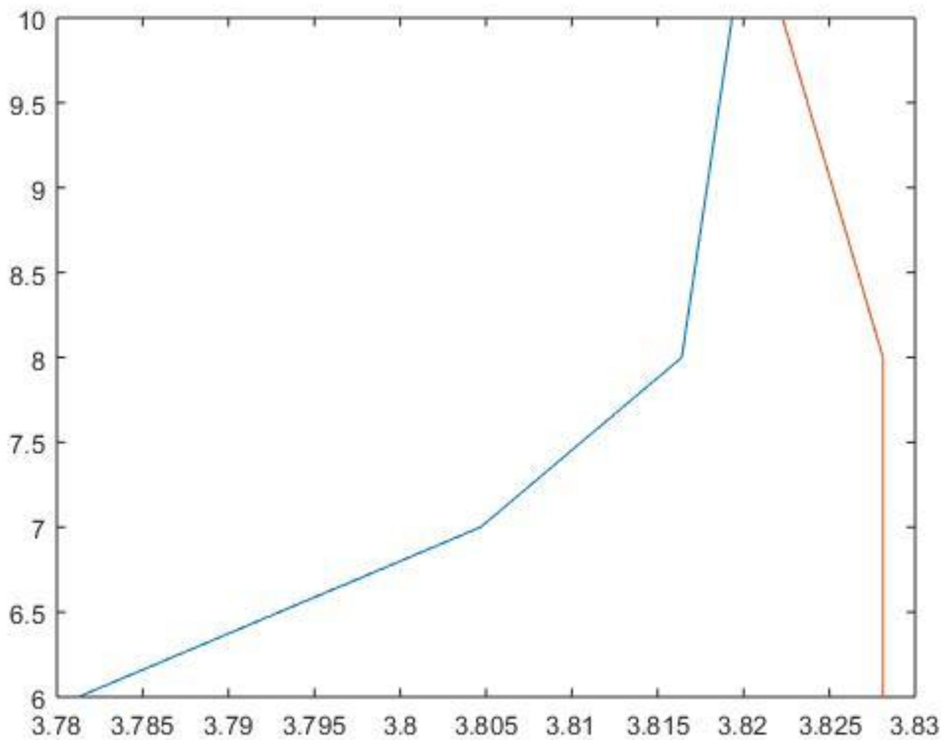
Ακολουθούν τρία διαγράμματα, ένα για κάθε συνάρτηση, που παρουσιάζουν τις γραφικές παραστάσεις (k , a_k) και (k , b_k).



Διάγραμμα 10, f_1(x): (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 11, $f_2(x)$: (k , a(k)) , (k , b(k)) = (y,x) .



Διάγραμμα 12, $f_3(x)$: (k , a(k)) , (k , b(k)) = (y,x) .

Όπως βλέπουμε στα διαγράμματα 10,11 και 12, και όπως λογικά περιμέναμε, όσο παραπάνω μειώνεται το εύρος του τελικού διαστήματος [a(k),b(k)], χρειάζονται παραπάνω βήματα του αλγορίθμου για να το φτάσουμε.

Συγκριτικός σχολιασμός

Αρχικά, αξίζει να τονιστεί το γεγονός πως ο κάθε αλγόριθμος έκανε τον ίδιο αριθμό βημάτων για να διαμερίσει το διάστημα έως ότου το μήκος του είναι μικρότερο ή ίσο του επιθυμητού εύρους διαστήματος λ, ανεξαρτήτως της συνάρτησης f(x).

Έπειτα, όσον αφορά τον αριθμό βημάτων που χρειάστηκε κάθε αλγόριθμος για να επιτυγχάνει τη ζητούμενη ακρίβεια, μπορούμε να συμπεράνουμε με ποια σειρά κατατάσσονται οι αλγόριθμοι από τις γραφικές παραστάσεις 2,3 και 4. Συγκεκριμένα, δεδομένου μία συγκεκριμένης τιμής λ (ας πάρουμε τη μικρότερη), μπορούμε να δούμε πόσες φορές κάλεσε την συνάρτηση f(x) ο κάθε αλγόριθμος. Όσον αφορά το πόσες κλήσεις τις f(x) πραγματοποιεί ο κάθε αλγόριθμος, έχουμε ότι:

- ο αλγόριθμος διχοτόμησης καλεί δύο φορές την συνάρτηση f(x) σε κάθε βήμα,
- οι αλγόριθμοι Χρυσού Τομέα και Fibonacci καλούν την f(x) [2 + (k-1)] φορές (δύο κλήσεις κατά το πρώτο βήμα για να υπολογιστούν τα αρχικά $f(x_1)$ και $f(x_2)$ και μία κλήση κάθε επόμενο βήμα μέχρι το τέλος),
- ενώ τέλος ο αλγόριθμος διχοτόμησης με χρήση παραγώγου καλεί την συνάρτηση f(x) k φορές.

Έτσι, έχουμε ότι για λ = 0.003, ο αλγόριθμος διχοτόμησης με χρήση παραγώγου συγκλίνει πρώτος μετά από 10 βήματα. ο αλγόριθμος διχοτόμησης συγκλίνει δεύτερος μετά από 12 βήματα, ο αλγόριθμος Fibonacci συγκλίνει τρίτος σε 14 βήματα και τέλος, ο αλγόριθμος Χρυσού Τομέα συγκλίνει τελευταίος μετά από 16 βήματα.

Όσον αφορά δε το ποιος αλγόριθμος καλεί λιγότερες φορές την συνάρτηση f(x), ο αλγόριθμος διχοτόμησης με χρήση παραγώγου καλεί τις λιγότερες φορές την f(x), πραγματοποιώντας συνολικά k κλήσεις. Ακολουθούν οι αλγόριθμοι Χρυσού Τομέα και Fibonacci, που πραγματοποιούν k+1 κλήσεις της f(x), ενώ τελευταίος είναι ο αλγόριθμος διχοτόμησης που πραγματοποιεί 2k κλήσεις της αντικειμενικής συνάρτησης f(x).