ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Βελτιστοποίησης:

Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής σε δοσμένο διάστημα

Αφροδίτη Λάτσκου

Νοέμβριος 2024

Σύνοψη εργασίας

Σκοπός μας ήταν η εύρεση του ελαχίστου με 4 μεθόδους και η σύγκριση της αποτελεσματικότητας και της απόδοσης τους στο διάστημα [-1,3]. Για τη σύγκριση χρησιμοποιούμε τις συναρτήσεις:

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

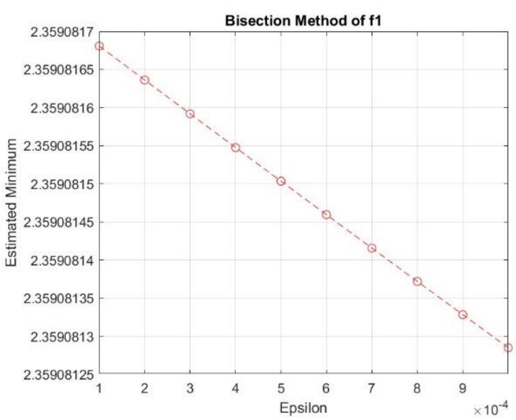
Χρησιμοποιώντας το Wolfram διαπιστώνουμε πως τα ακριβή ελάχιστα των συναρτήσεων είναι $2.359\, για\ \ x_1^* = 1.15$ για την $f_1(x)$ και $0.018\, για\ \ x_2^* = 2.018$ για την $f_2(x)$ αντίστοιχα. Όσον αφορά $f_3(x)$, παρατηρούμε πως δεν υπάρχει ολικό ελάχιστο αλλά υπάρχουν πολλά τοπικά ελάχιστα.

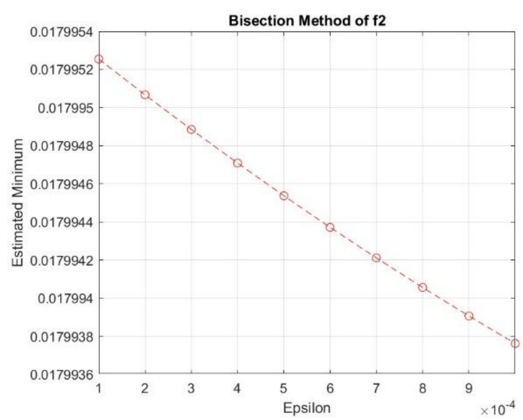
Μέθοδος της Διχοτόμου

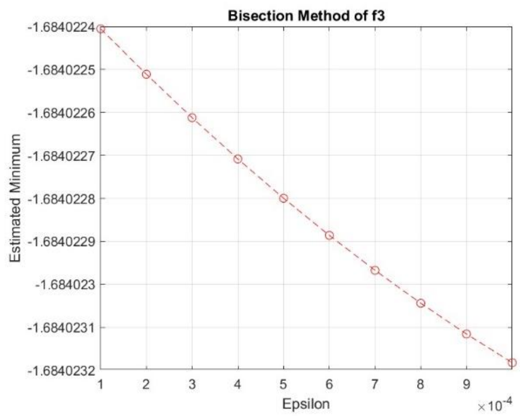
Η μέθοδος της διχοτόμου προσεγγίζει το θέμα της ελαχιστοποίησης παίρνοντας κάθε φορά το σημείο της διχοτόμου των δύο άκρων και μετατοπίζοντας το ένα άκρο εκεί.

Με τη δοκιμή της μεθόδου στις δοθείσες συναρτήσεις παίρνουμε τα εξής γραφήματα:

α. *Τις αλλαγές του αποτελέσματος με βάση το ε και l = 0.01*





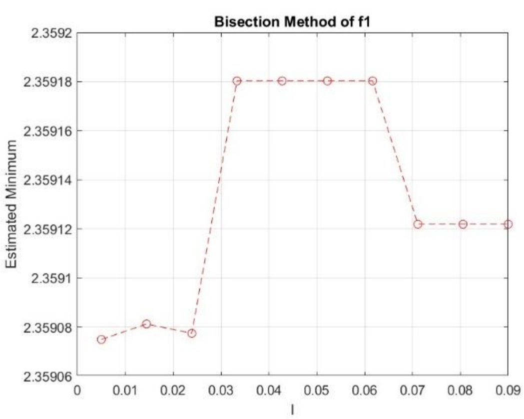


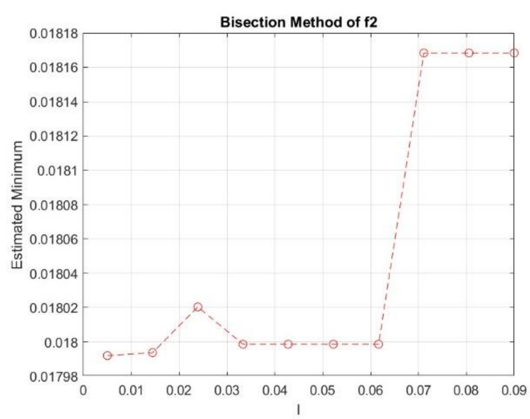
Παρατηρούμε αρχικά ότι οι εκτιμώμενες θέσεις ελαχίστου ταυτίζονται με τα αποτελέσματα που περιμέναμε. Όπως φαίνεται, ειδικά για την $f_3(x)$, ο αλγόριθμος εγκλωβίζεται στο τοπικό ελάχιστο που βρίσκεται στο $x = -1.684$ και τερματίζει εκεί.

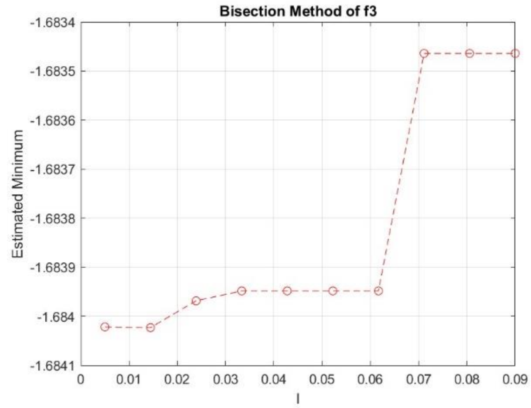
Επίσης βλέπουμε πως η μεταβολή του ελαχίστου είναι γραμμική και για τις τρεις συναρτήσεις. Αυτό εξηγείται εύκολα, καθώς στη συγκεκριμένη μέθοδο το περιθώριο σφάλματος ε δηλώνει απλά το πόσο μακριά θα είναι τα νέα σημεία που επιλέγουμε από την πραγματική διχοτόμο.

Για τα γραφήματα κρατήσαμε το ε σε χαμηλά επίπεδα και επομένως το ελάχιστο δεν επηρεάζεται σε τόσο σημαντικό βαθμό. Ωστόσο παρατηρήθηκε ότι για μεγαλύτερο ε (για παράδειγμα για ε = 0.01) η μέθοδος δεν δίνει αποτέλεσμα καθώς φτάνει σε ένα σημείο όπου τα x1 και x2 αλλάζουν συνεχώς στους εαυτούς τους και επομένως μπαίνει σε infinite loop.

β. Τις αλλαγές του αποτελέσματος με βάση το l και ε = 0.001



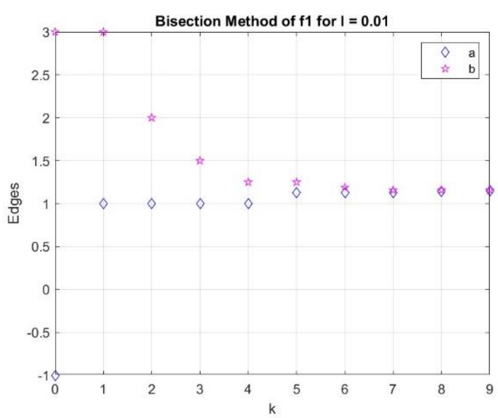


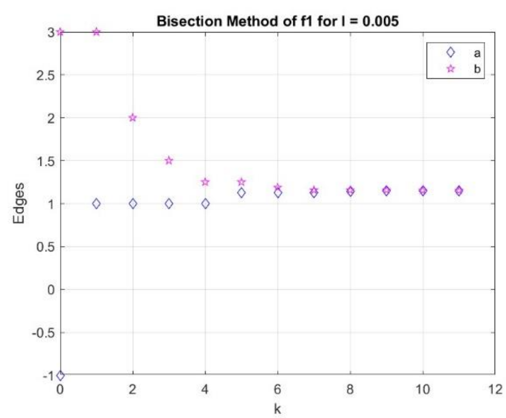


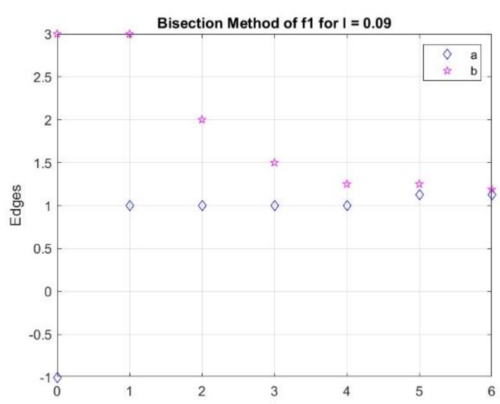
Ομοίως με το ε, παρατηρήθηκε ότι για πολύ μικρές τιμές του l η μέθοδος δε δουλεύει. Ωστόσο η μεταβολή δεν είναι καθόλου γραμμική και μάλιστα για σχετικά μικρή μεταβολή του l μπορεί να έχουμε σχετικά μεγάλη αλλαγή στο ελάχιστο, κάτι που μειώνει την ακρίβεια της μεθόδου σημαντικά.

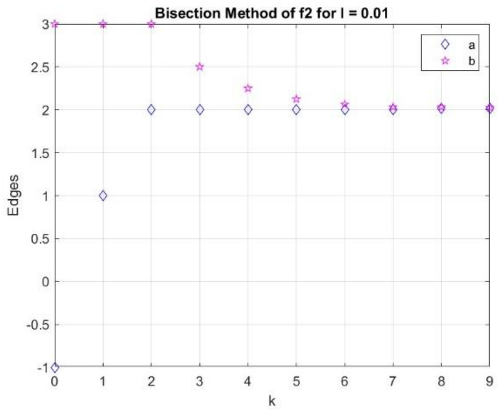
γ. Τις αλλαγές των α και β με βάση κάποια l και ε = 0.001

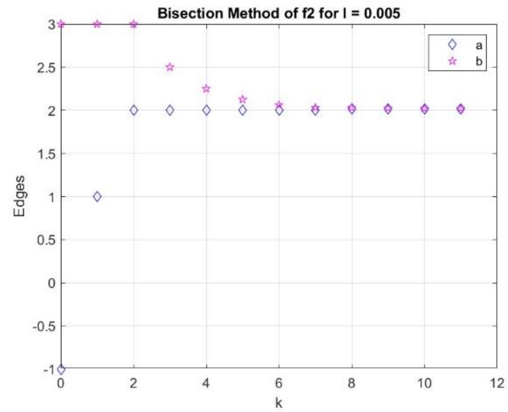
Θέλαμε να δούμε πώς αλλάζουν τα άκρα α και β σε κάθε επανάληψη. Για αυτόν τον σκοπό επιλέχθηκαν οι τιμές του l: 0.01, 0.005, 0.09 καθώς αποτελούν μια τιμή ενδιαφέροντος και τα άκρα του διαστήματος το οποίο μελετάμε.

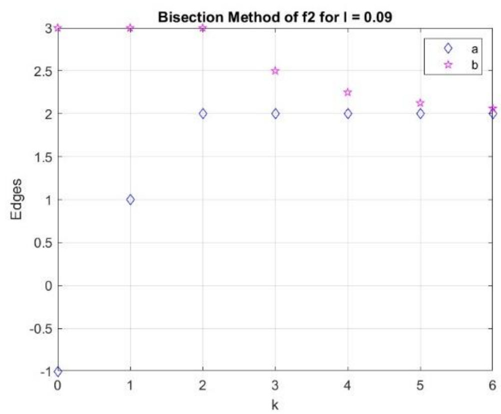


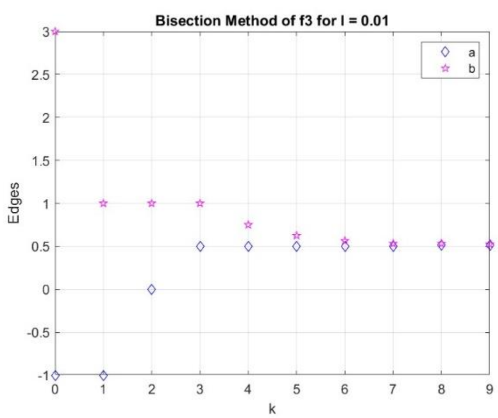


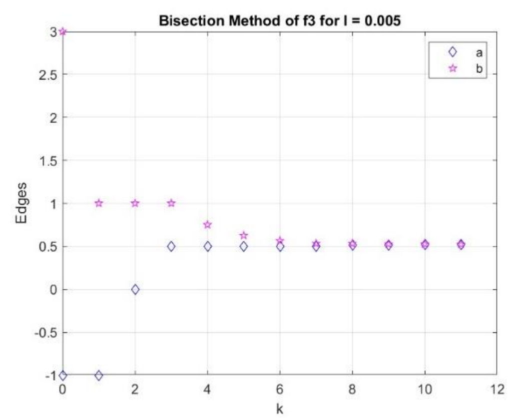


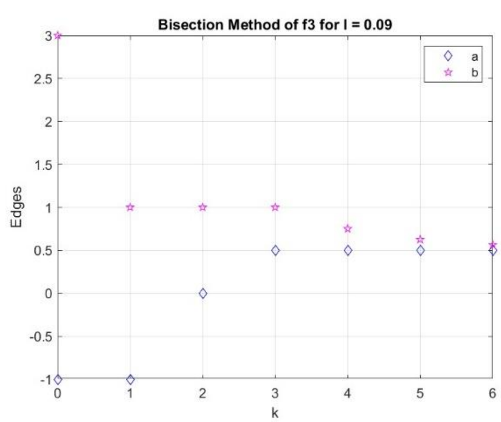

Bisection Method of f2 for l = 0.01
Edges
k
a
b


Bisection Method of f2 for l = 0.005
Edges
k
a
b


Bisection Method of f2 for l = 0.09
Edges
k
a
b


Bisection Method of f3 for l = 0.01
Edges
k
a
b


Bisection Method of f3 for l = 0.005
Edges
k
a
b


Bisection Method of f3 for l = 0.09
Edges
k
a
b

Προφανώς, για μεγαλύτερο l θα έχουμε λιγότερες επαναλήψεις άρα ο αλγόριθμος τερματίζει πιο γρήγορα. Ωστόσο, όπως είπαμε παραπάνω αυτό κοστίζει σε ακρίβεια με ανάλογο τρόπο.

Για λόγους σύγκρισης μετρήσαμε επίσης τους χρόνους που κατέγραψε ο αλγόριθμος με είσοδο την κάθε συνάρτηση για ε=0.001 και l=0.01. Παρατηρούμε ότι παρότι και οι τρεις συναρτήσεις χρειάζονται ακριβώς τον ίδιο αριθμό επαναλήψεων για αυτά τα δεδομένα ο χρόνος εκτέλεσης διαφέρει ελαφρώς για κάθε μια από αυτές, με λίγο μεγαλύτερο αυτόν της f3. Ενδεικτικά κάποιοι από αυτούς:

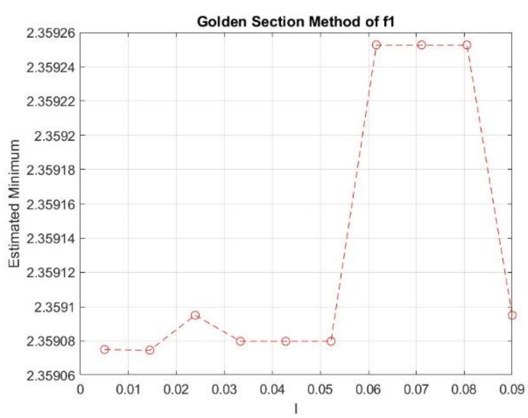
f1	f2	f3
1.3159	1.5886	1.7751
0.8831	0.9458	1.9881
0.8015	1.2317	1.9788

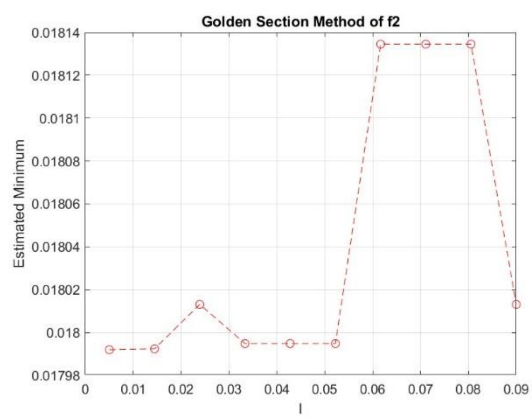
Συνολικά, η μέθοδος της διχοτόμου είχε την ευκολότερη υλοποίηση και προσφέρει έναν πολύ απλό τρόπο εύρεσης του ελάχιστου.

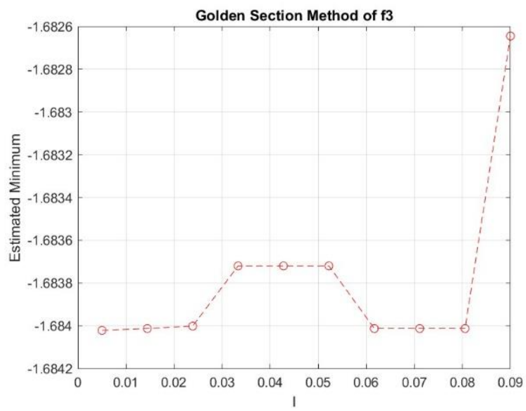
Μέθοδος του Χρυσού Τομέα

Η μέθοδος του χρυσού τομέα εκμεταλλεύεται ιδιότητες της χρυσής τομής ώστε να βρει το ελάχιστο μιας συνάρτησης. Σε αντίθεση με την προηγούμενη, δεν χρησιμοποιεί κάποια απόσταση ε. Με τη δοκιμή της στις δοθείσες συναρτήσεις παίρνουμε τα εξής γραφήματα:

α. Τις αλλαγές του αποτελέσματος με βάση το l



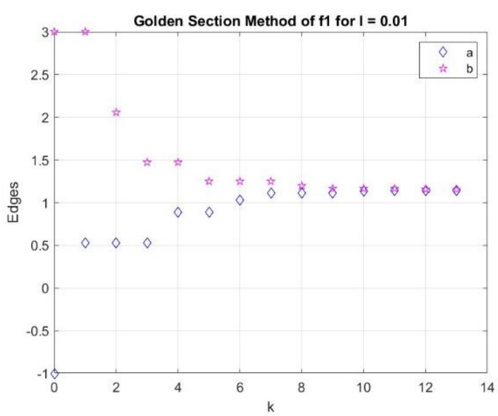


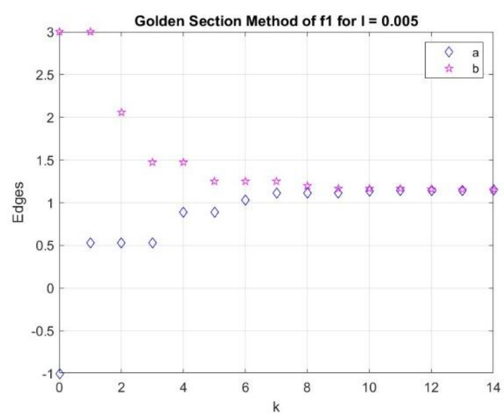


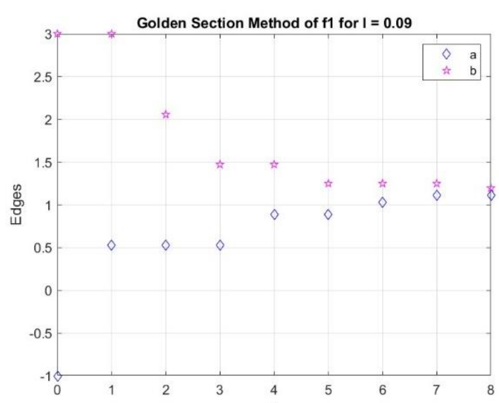
Για αυτή την μέθοδο οι συμπεριφορές των συναρτήσεων φαίνεται να έχουν αρκετά κοινά στοιχεία με της προηγούμενης, συγκεκριμένα βλέπουμε πάλι πως η ακρίβεια δεν μειώνεται απαραίτητα γραμμικά με την αύξηση του διαστήματος, αλλά έχει βέλτιστες περιοχές στις οποίες λειτουργεί καλύτερα.

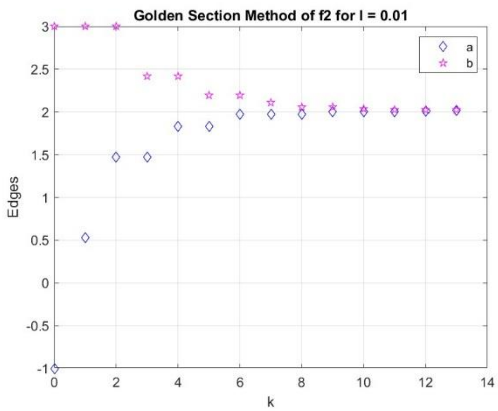
β. Τις αλλαγές των α και β με βάση κάποια l

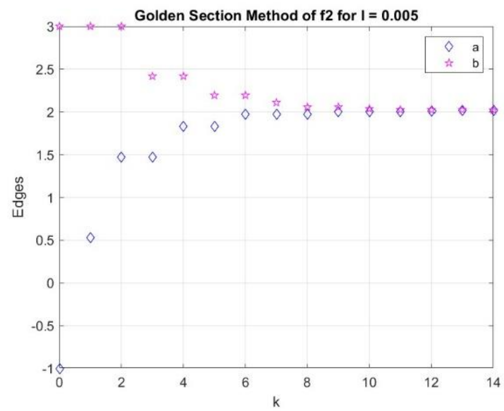
Συνεχίζουμε με τα l που επιλέξαμε και στην προηγούμενη μέθοδο και βρίσκουμε:

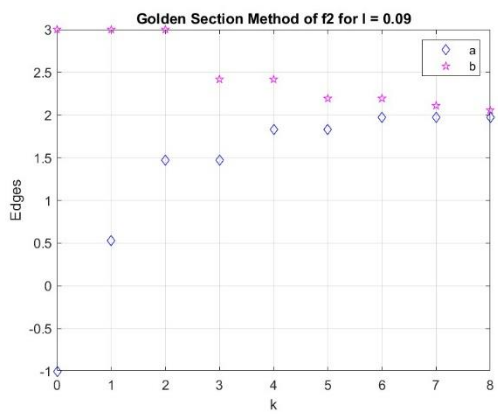


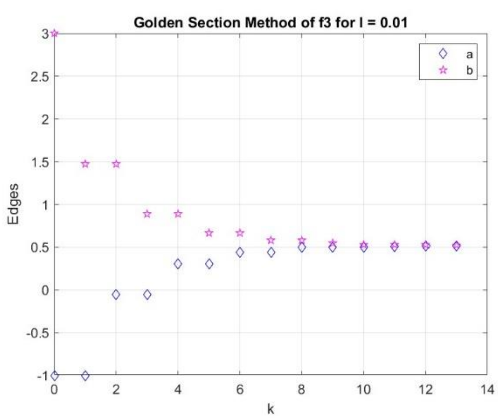


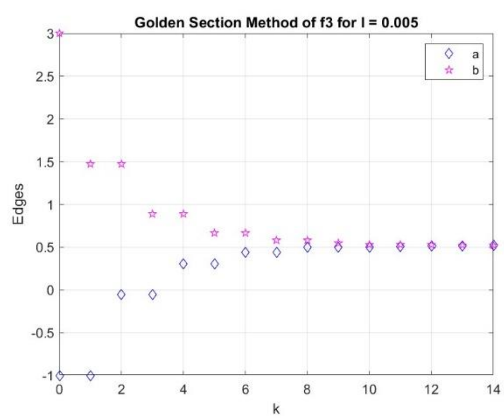


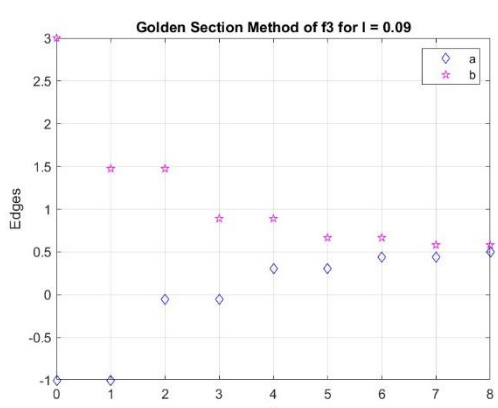

Golden Section Method of f2 for l = 0.01
a
b
Edges
k


Golden Section Method of f2 for l = 0.005
a
b
Edges
k


Golden Section Method of f2 for l = 0.09
a
b
Edges
k


Golden Section Method of f3 for l = 0.01
a
b
Edges
k


Golden Section Method of f3 for l = 0.005
a
b
Edges
k


Golden Section Method of f3 for l = 0.09
a
b
Edges
k

Σε σύγκριση με τη μέθοδο της διχοτόμου βλέπουμε ότι η συγκεκριμένη μέθοδος χρειάζεται περισσότερες επαναλήψεις για να φτάσει στο ελάχιστο, ωστόσο η διαφορά στα αποτελέσματα διαφέρει ελάχιστα, σε βαθμό που φαινομενικά τουλάχιστον δεν είναι αρκετός για να πούμε ότι κάποια είναι καλύτερη από την άλλη μόνο από αυτό. Ωστόσο, αν κοιτάξουμε τους νέους χρόνους:

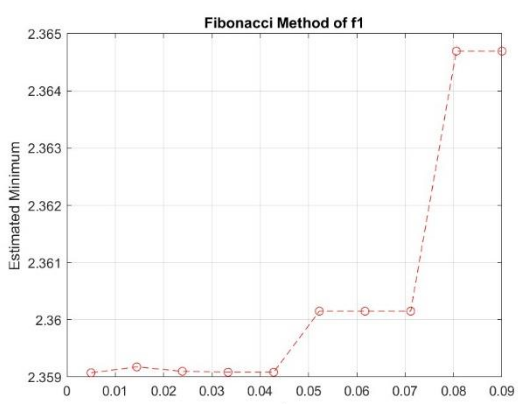
f1	f2	f3
0.9023	1.0023	0.8923
0.8897	0.8484	0.9133
0.9060	0.9286	0.8972

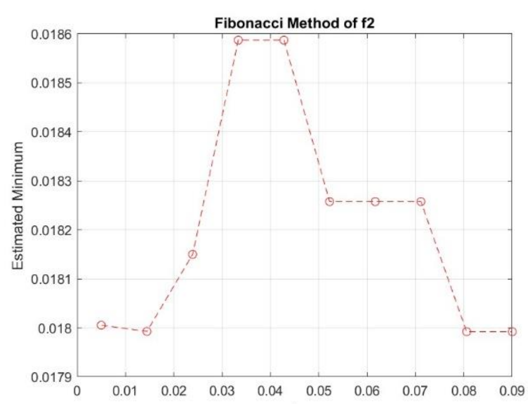
Παρατηρούμε μια βελτίωση και κυρίως παρατηρούμε πως οι χρόνοι μεταξύ των συναρτήσεων είναι πολύ πιο κοντά. Αυτό συμβαίνει καθώς σε κάθε επανάληψη χρειάζεται μόνο ένας επιπλέον υπολογισμός, καθώς και επειδή αυτή η μέθοδος αφαιρεί από το διάστημα κάθε φορά μια μεγαλύτερη περιοχή στην οποία δεν υπάρχει το ελάχιστο.

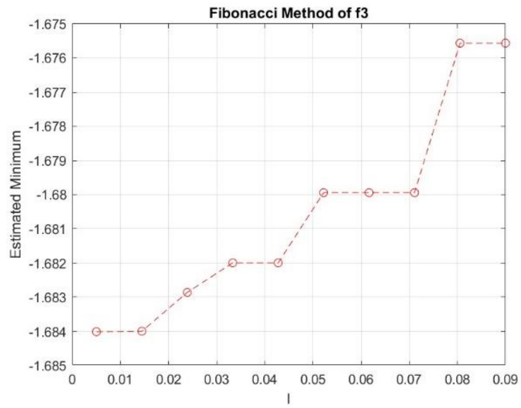
Μέθοδος Fibonacci

Η μέθοδος Fibonacci επιλέγει το νέο διάστημα σε κάθε επανάληψη χρησιμοποιώντας τους ήδη υπολογισμένους αριθμούς Fibonacci μέχρι κάποιο n που ικανοποιεί το κριτήριο $F_n > (b - a)/l$. Με τη δοκιμή της στις δοθείσες συναρτήσεις παίρνουμε τα εξής γραφήματα:

α. Τις αλλαγές του αποτελέσματος με βάση το l



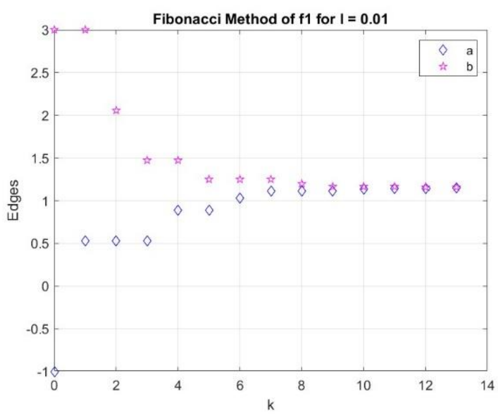


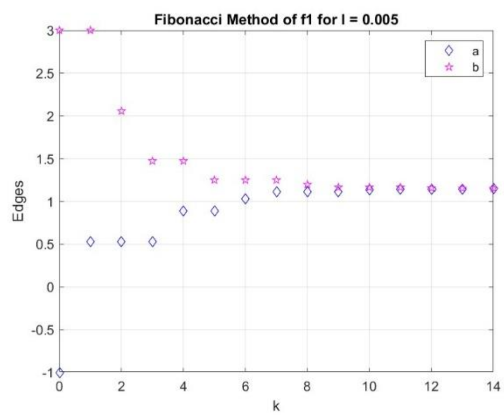


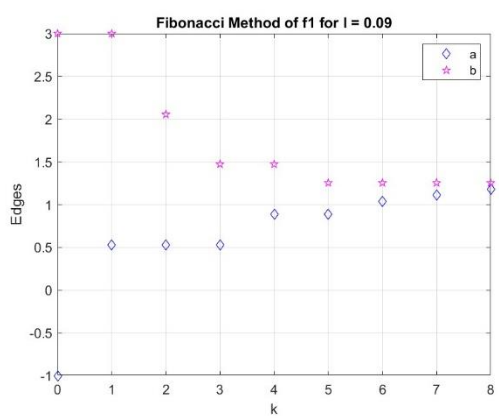
Εδώ παρατηρούμε πως τα διαγράμματα αλλάζουν αρκετά ανάλογα με την συνάρτηση. Ωστόσο υπάρχει πάλι μια τάση να μειώνεται η ακρίβεια όσο προχωράμε σε μεγαλύτερα διαστήματα.

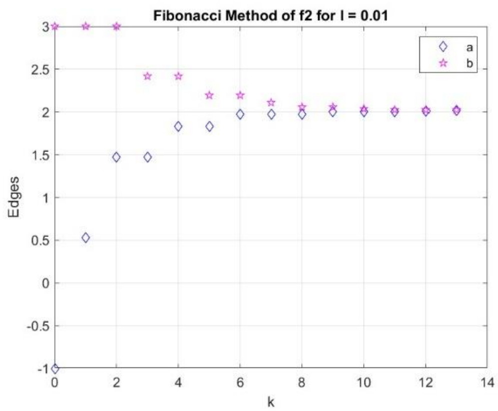
β. Τις αλλαγές των α και β με βάση κάποια l

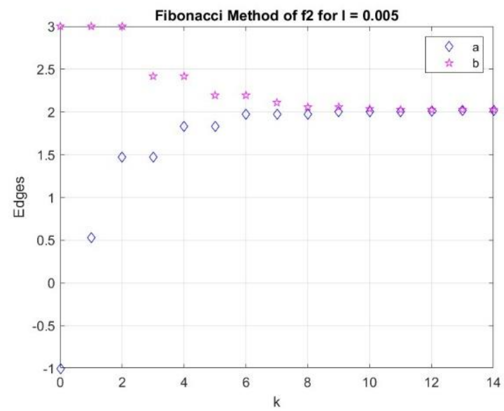
Συνεχίζουμε με τα l που επιλέξαμε και στην προηγούμενη μέθοδο και βρίσκουμε:

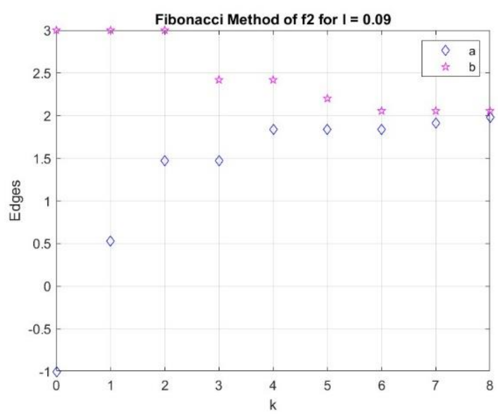


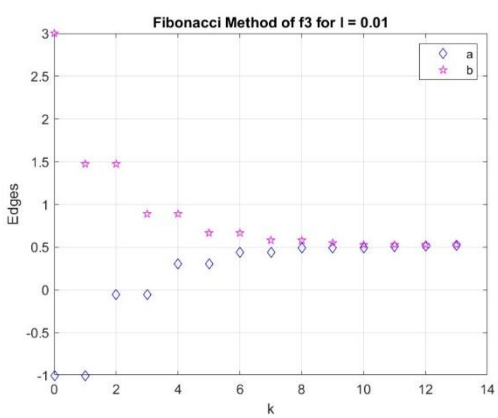


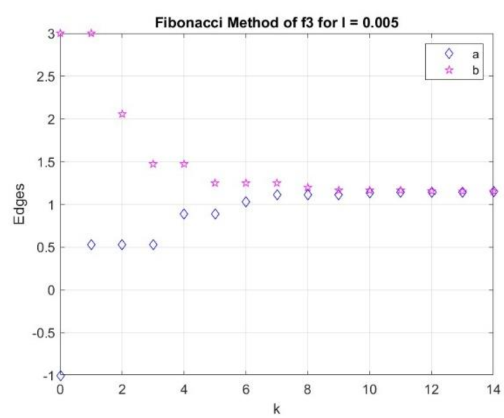


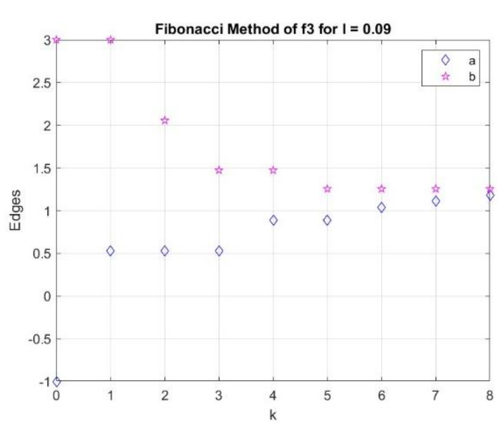

Fibonacci Method of f2 for l = 0.01
Edges
k
a
b


Fibonacci Method of f2 for l = 0.005
Edges
k
a
b


Fibonacci Method of f2 for l = 0.09
Edges
k
a
b


Fibonacci Method of f3 for l = 0.01
Edges
k
a
b


Fibonacci Method of f3 for l = 0.005
Edges
k
a
b


Fibonacci Method of f3 for l = 0.09
Edges
k
a
b

Βλέπουμε πως παίρνουμε πολύ παρόμοια αποτελέσματα με την μέθοδο της χρυσής τομής και πάλι ο αλγόριθμος χρειάζεται γενικά περισσότερα βήματα από την διχοτόμο, με παρόμοια ακρίβεια. Όσον αφορά τους χρόνους:

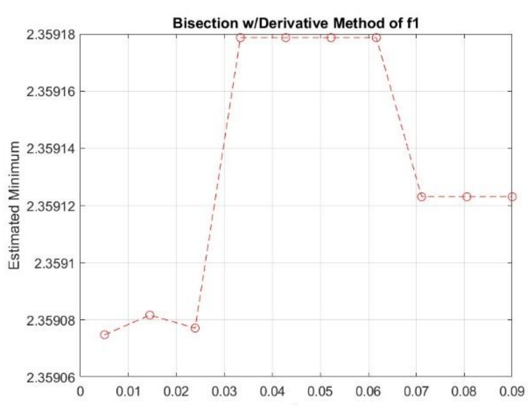
f1	f2	f3
0.8823	0.9823	0.8874
0.8724	0.9564	0.8939
0.9120	1.0234	0.8785

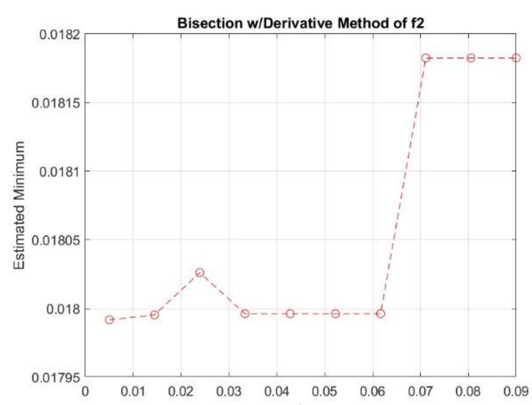
Μπορούμε δηλαδή να συμπεράνουμε πως η ταχύτητα του αλγορίθμου Fibonacci και της χρυσής τομής είναι συγκρίσιμη σε σημείο να πούμε πως οι δύο αυτοί αλγόριθμοι όσον αφορά την ακρίβεια, την ταχύτητα και την πολυπλοκότητα υλοποίησης βρίσκονται στην ίδια θέση.

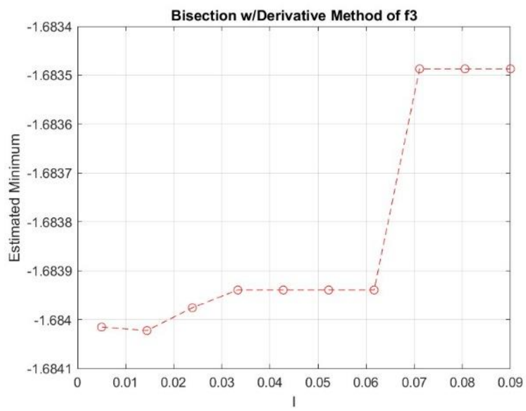
Μέθοδος Διχοτόμου με Χρήση Παραγώγου

Αυτή η μέθοδος είναι παρόμοια με τη μέθοδο διχοτόμου αλλά χρησιμοποιεί την παράγωγο στην απόφαση του ποιο άκρο να αλλάξει. Με τη δοκιμή της στις δοθείσες συναρτήσεις παίρνουμε τα εξής γραφήματα:

α. Τις αλλαγές του αποτελέσματος με βάση το l



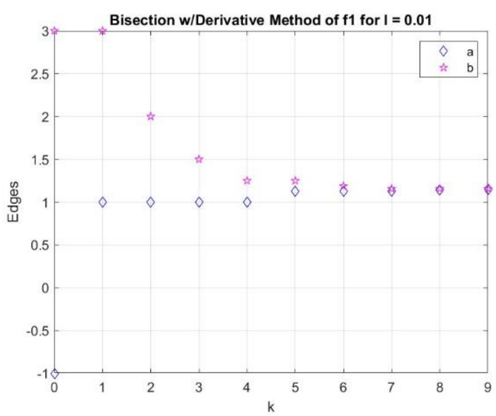


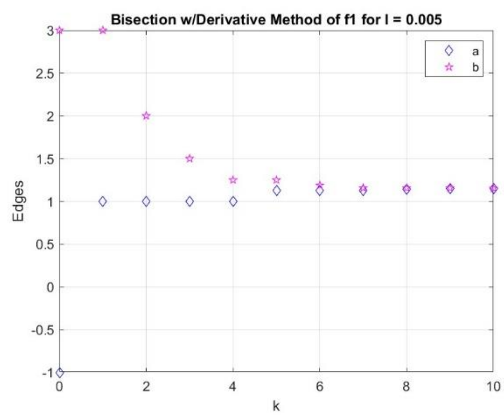


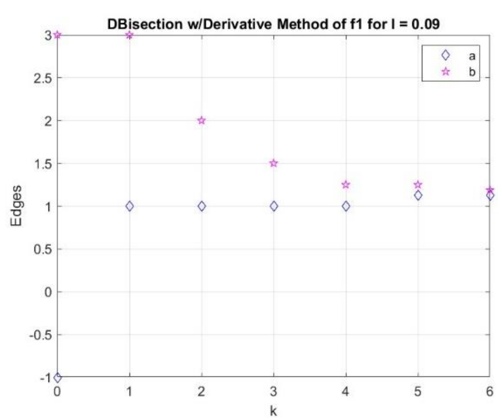
Παρατηρούμε πως η συμπεριφορά των συναρτήσεων είναι ολόιδια, και με την αύξηση του l έχουμε και μια απόκλιση από την εκτιμώμενη τιμή που όμως είναι πάντα αυξανόμενη.

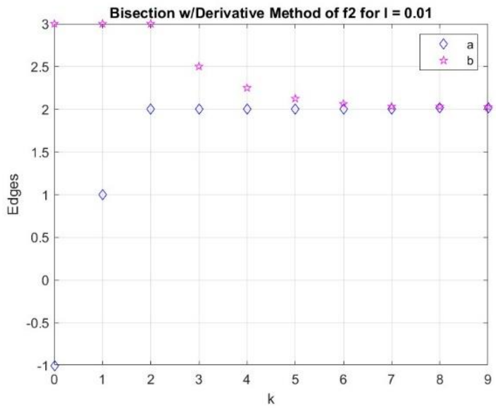
β. Τις αλλαγές των α και β με βάση κάποια l

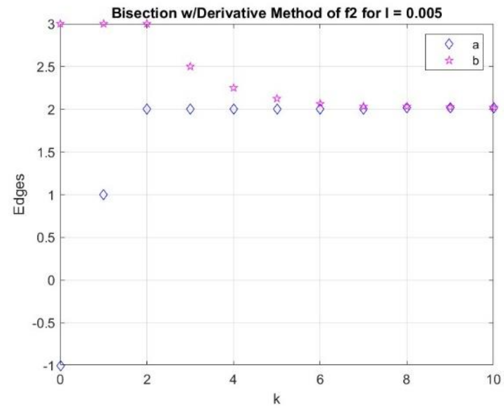
Συνεχίζουμε με τα l που επιλέξαμε και στην προηγούμενη μέθοδο και βρίσκουμε:

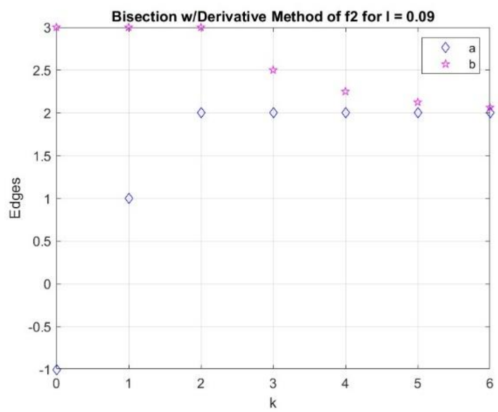


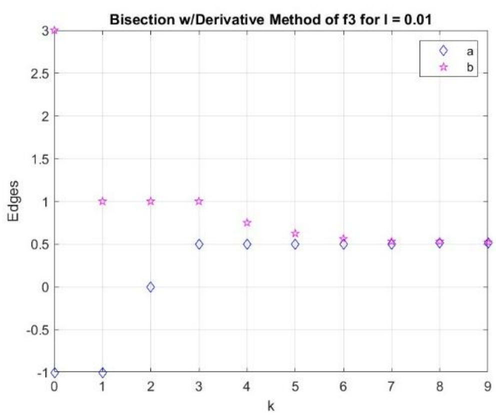


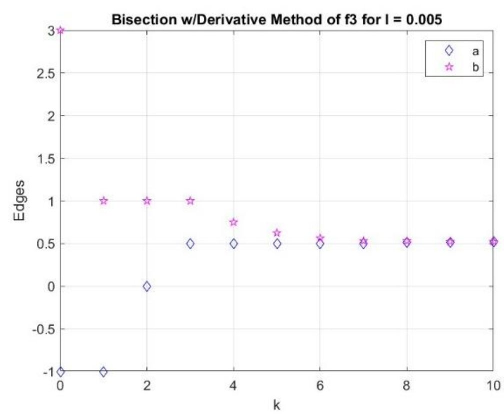


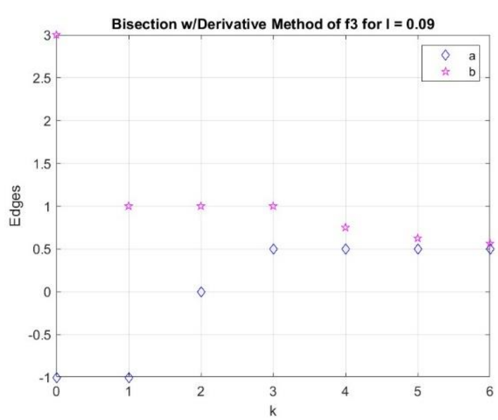

Bisection w/Derivative Method of f2 for l = 0.01
Edges
k
a
b


Bisection w/Derivative Method of f2 for l = 0.005
Edges
k
a
b


Bisection w/Derivative Method of f2 for l = 0.09
Edges
k
a
b


Bisection w/Derivative Method of f3 for l = 0.01
Edges
k
a
b


Bisection w/Derivative Method of f3 for l = 0.005
Edges
k
a
b


Bisection w/Derivative Method of f3 for l = 0.09
Edges
k
a
b

Είναι προφανές πως για κοινές παραμέτρους φτάνουμε στα επιθυμητά αποτελέσματα με αρκετά λιγότερες επαναλήψεις από τις προηγούμενες μεθόδους χωρίς να μειωθεί η ακρίβεια. Όσον αφορά τον χρόνο:

f1	f2	f3
0.7934	0.8134	0.6123
0.7983	0.8244	0.7239
0.8123	0.8783	0.7867

Ο χρόνος εκτέλεσης είναι σχετικά καλύτερος πράγμα που κάνει τη μέθοδο της διχοτόμου με παράγωγο την θεωρητικά γρηγορότερη από όσες είδαμε μέχρι στιγμής, κάτι που επιβεβαιώνεται από τη χρήση λιγότερων βημάτων. Ωστόσο, οι χρόνοι αυτοί είναι ενδεικτικοί και δεν επαρκούν για να γενικεύσουμε τα συμπεράσματα μας όσον αφορά την ταχύτητα, εφόσον εμπεριέχουν και την κατάσταση του υπολογιστή την δεδομένη στιγμή και όχι καθαρά την ταχύτητα του αλγορίθμου καθαυτού. Ένα σημαντικό μειονέκτημα αυτής της μεθόδου είναι ότι για τη χρήση της είναι απαραίτητος ο υπολογισμός της παραγώγου εκ των προτέρων. Αυτό προσθέτει μια μικρή δυσκολία στην υλοποίηση της καθώς ειδικά για το matlab χρειάζεται συγκεκριμένη βιβλιοθήκη και χρήση νέων συναρτήσεων.