1η Εργαστηριακή Άσκηση:

Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής σε δοσμένο διάστημα

Σκοπός μας ήταν η εύρεση του ελαχίστου με 4 μεθόδους και η σύγκριση της αποτελεσματικότητας και της απόδοσης τους. Για τη σύγκριση χρησιμοποιούμε τις συναρτήσεις

$$f1(x) = (x - 1)^3 + (x - 4)^2 cos(x)$$

$$f2(x) = e^{-2x} + (x - 2)^2$$

$$f(x) = x^2 \cdot ln(0.5x) + sin(0.2x)^2$$

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.

Τα αποτελέσματα που πήραμε είναι:

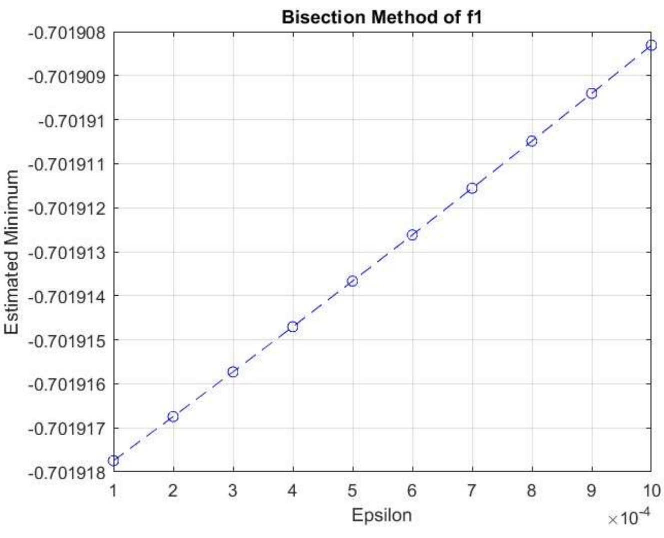
$f1(x) = -0.7019$ για $x = 1.9245$

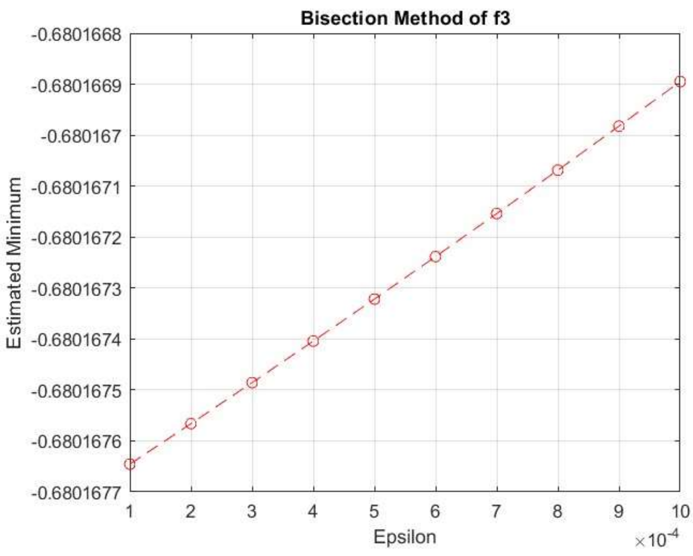
$f2(x) = 0.0180$ για $x = 2.0182$

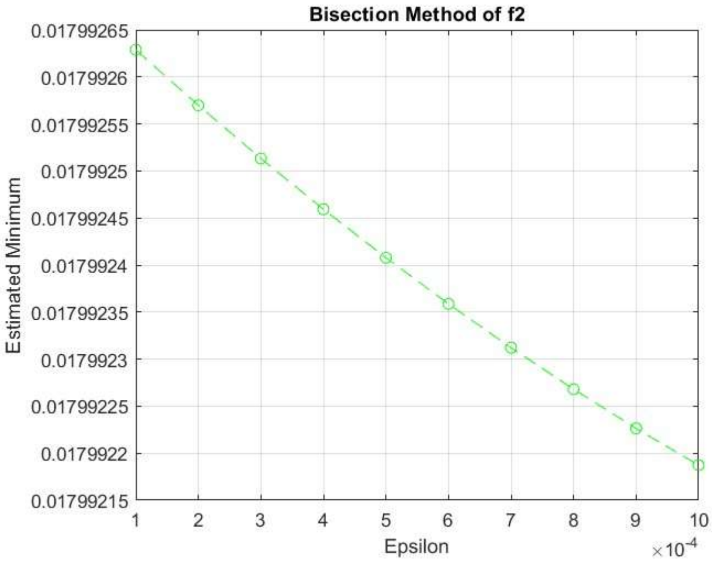
$f3(x) = -0.6792$ για $x = 1.1633$

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

α. <u>Τις αλλαγές του αποτελέσματος με βάση το ε</u>



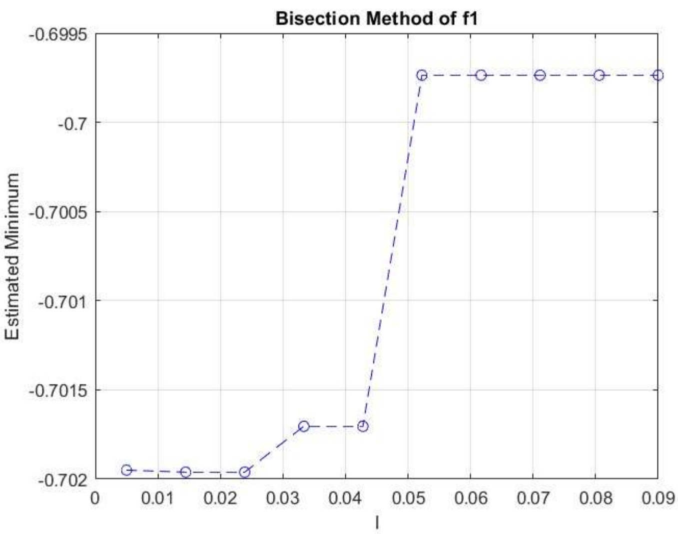


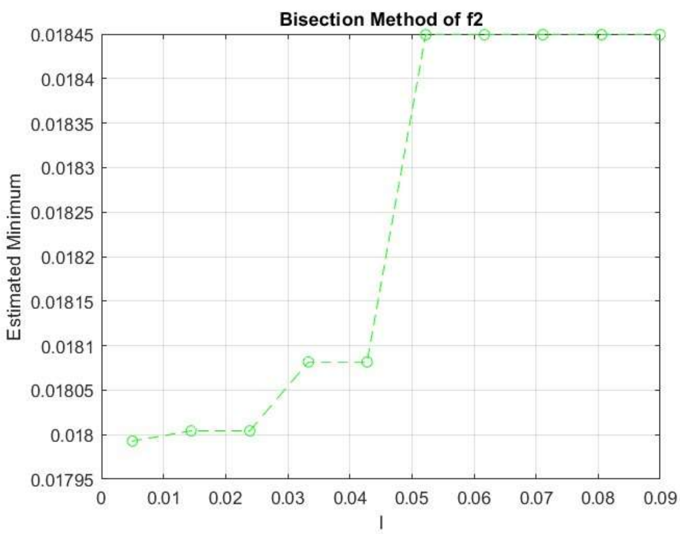


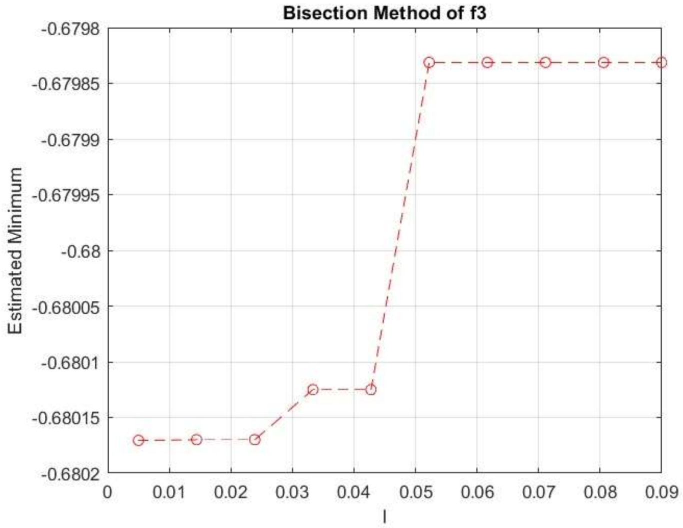
Παρατηρούμε ότι η μεταβολή του ελαχίστου είναι γραμμική και για τις τρεις συναρτήσεις. Αυτό εξηγείται εύκολα, καθώς στη συγκεκριμένη μέθοδο το περιθώριο σφάλματος ε δηλώνει απλά το πόσο μακριά θα είναι τα νέα σημεία που επιλέγουμε από την πραγματική διχοτόμο.

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. <u>Τις αλλαγές του αποτελέσματος με βάση το l</u>



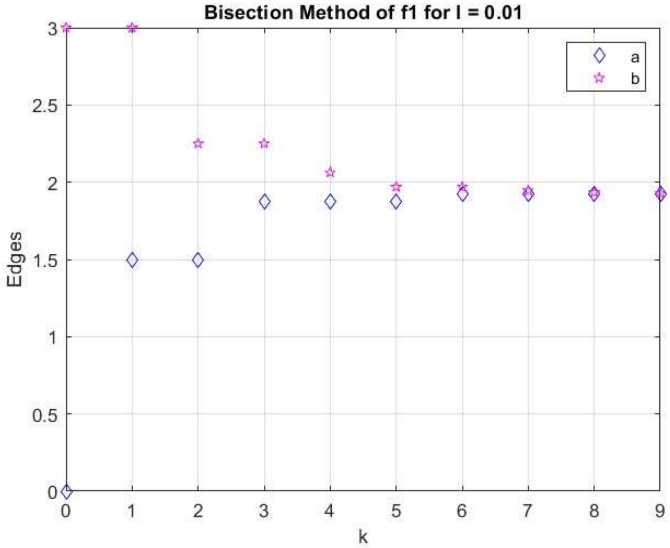


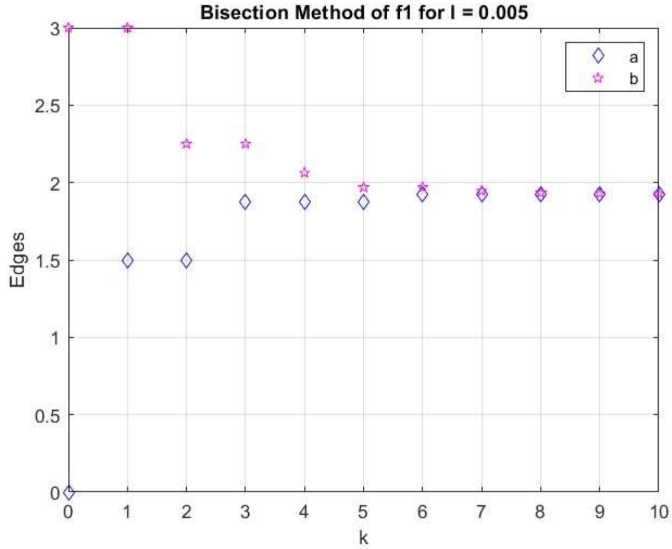


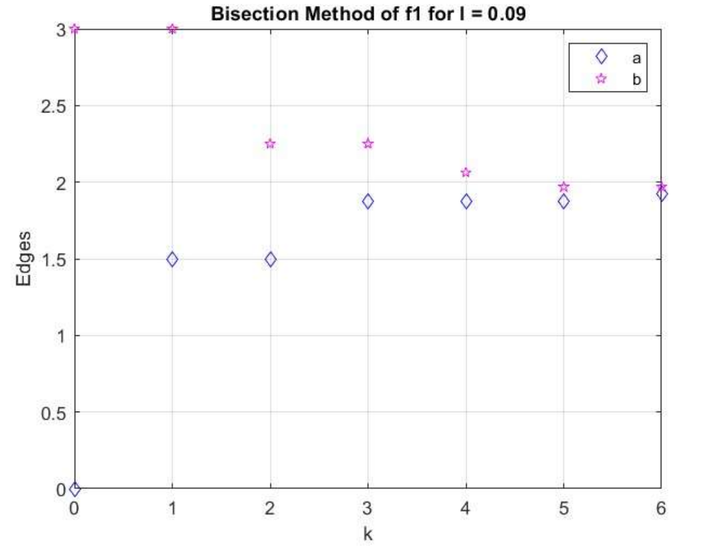
Ομοίως με το ε, παρατηρήθηκε ότι για πολύ μικρές τιμές του l η μέθοδος δε δουλεύει. Ωστόσο η μεταβολή δεν είναι καθόλου γραμμική και μάλιστα για σχετικά μικρή μεταβολή του l μπορεί να έχουμε σχετικά μεγάλη αλλαγή στο ελάχιστο, κάτι που μειώνει την ακρίβεια της μεθόδου σημαντικά.

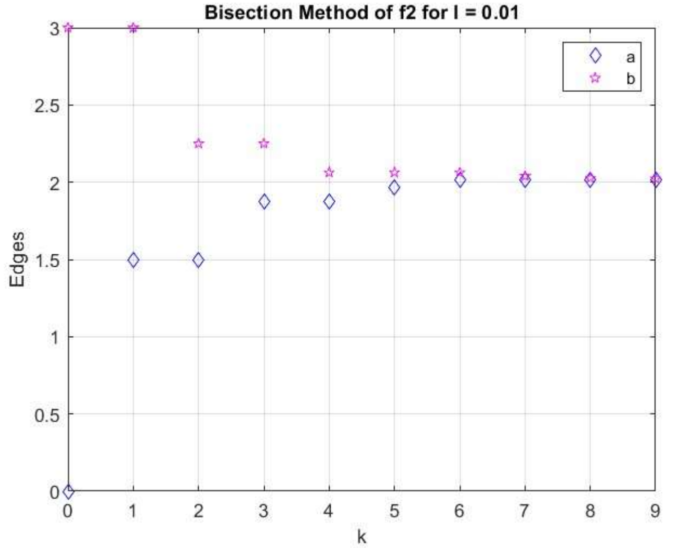
γ. <u>Τις αλλαγές των α και β με βάση κάποια l</u>

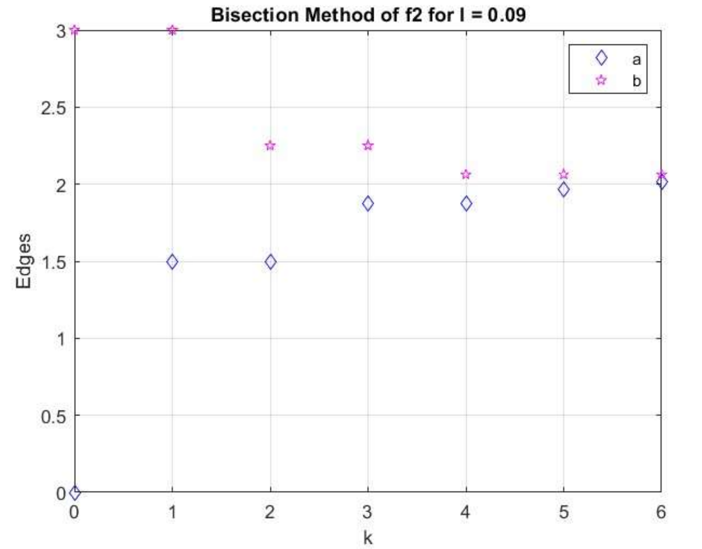
Θέλαμε να δούμε πώς αλλάζουν τα άκρα α και β σε κάθε επανάληψη. Για αυτόν τον σκοπό επιλέχθηκαν οι τιμές του l 0.005, 0.01, 0.09 καθώς οι πρώτη και τελευταία είναι τα άκρα του διαστήματος της τιμής το οποίο μελετάμε και το κεντρικό καθώς είναι μια τιμή που μας ενδιαφέρει σε προηγούμενο παράδειγμα.

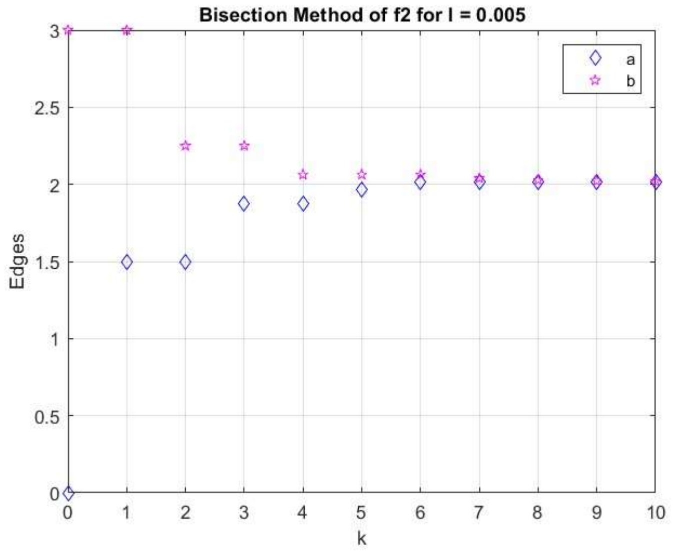


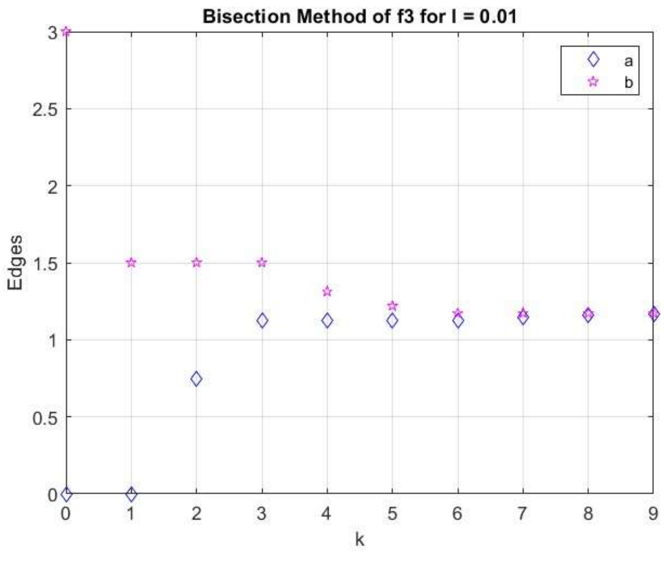


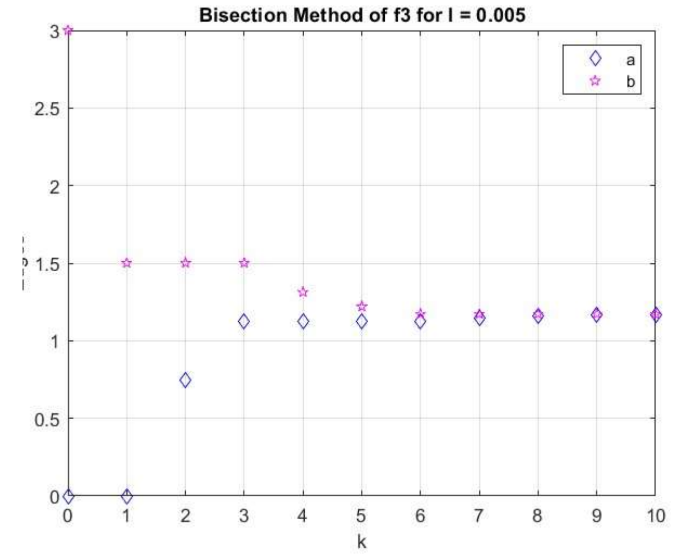

Bisection Method of f1 for l = 0.09
a
b
Edges
k

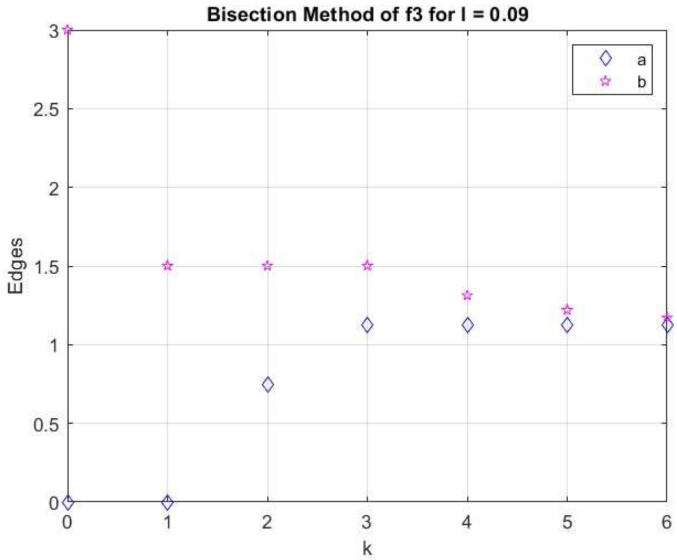

Bisection Method of f2 for l = 0.01
a
b
Edges
k


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


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


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


Bisection Method of f3 for l = 0.005
a
b
k



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

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 για κάθε μια από αυτές, με μεγαλύτερο αυτόν της f2. Ενδεικτικά κάποιοι από αυτούς:

f1	f2	f3
0.2235	0.4751	0.1790
0.3072	0.6315	0.2257
0.2134	0.5816	0.2094

Περιέργως, αυτό δεν οφείλεται ούτε στην πράξη της εύρεσης της τιμής της f(x) μέσω της συνάρτησης subs() αλλά και ούτε στην ίδια την πράξη της σύγκρισης καθώς όταν αυτές δοκιμάστηκαν μόνες τους ο χρόνος της f2 ήταν μικρότερος αν όχι και από τις δύο τουλάχιστον απο τη μια άλλη συνάρτηση.

Συνολικά, η μέθοδος της διχοτόμου είχε την ευκολότερη υπολοποίηση 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.

Τα αποτελέσματα που πήραμε για είναι:

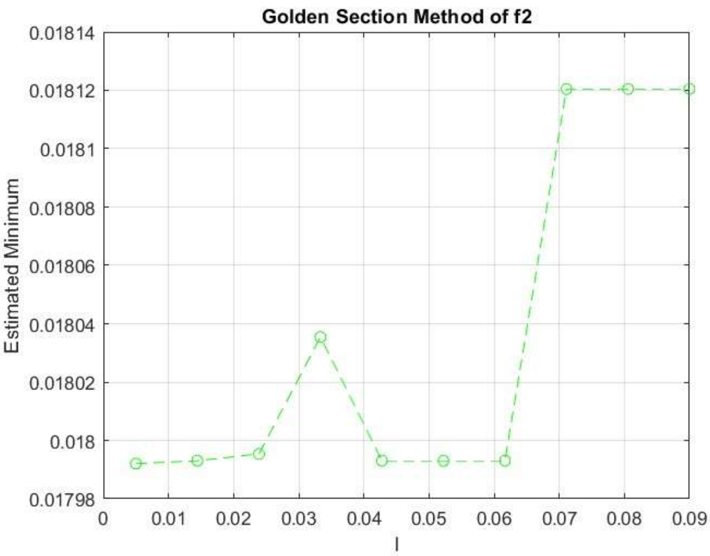
$f1(x) \ = \ -0.7020$ για $x \ = \ 1.9284$

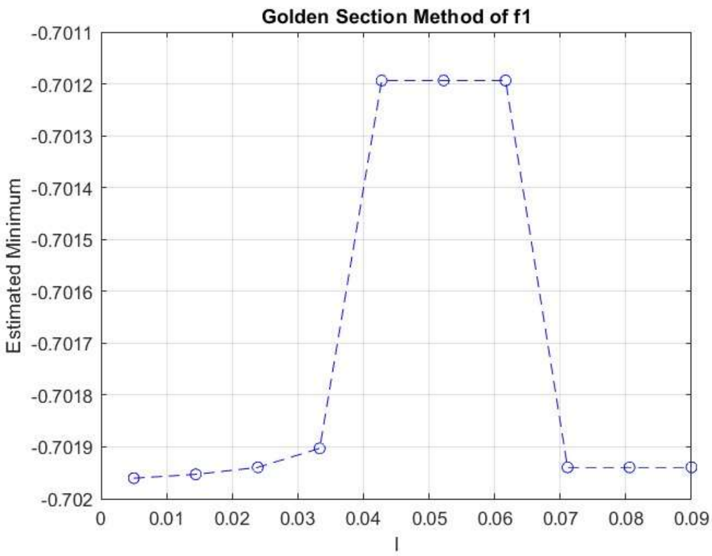
$f2(x) \ = \ 0.0180$ για $x \ = \ 2.0166$

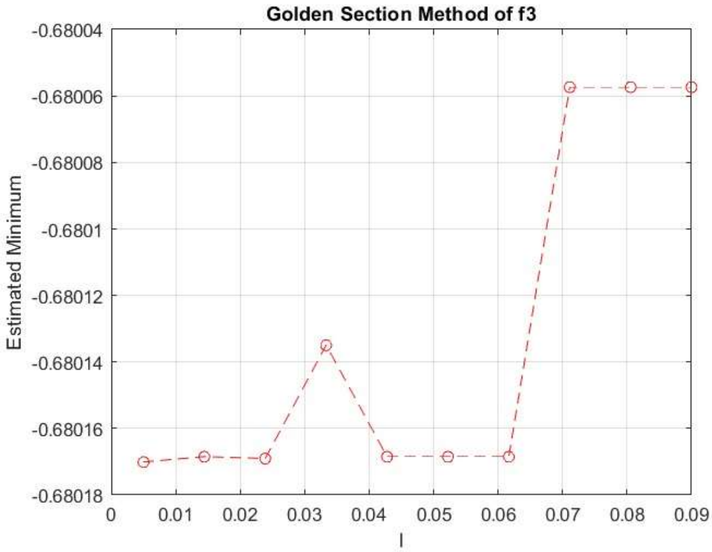
$f3(x) \ = \ -0.6792$ για $x \ = \ 1.1656$

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

α. <u>Τις αλλαγές του αποτελέσματος με βάση το l</u>



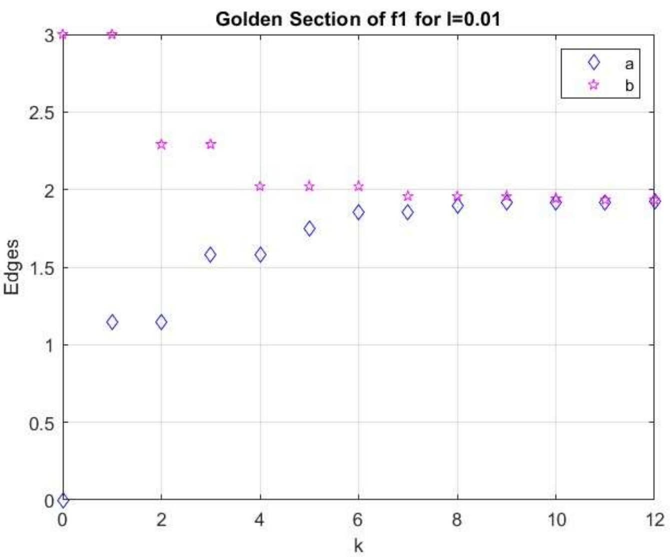


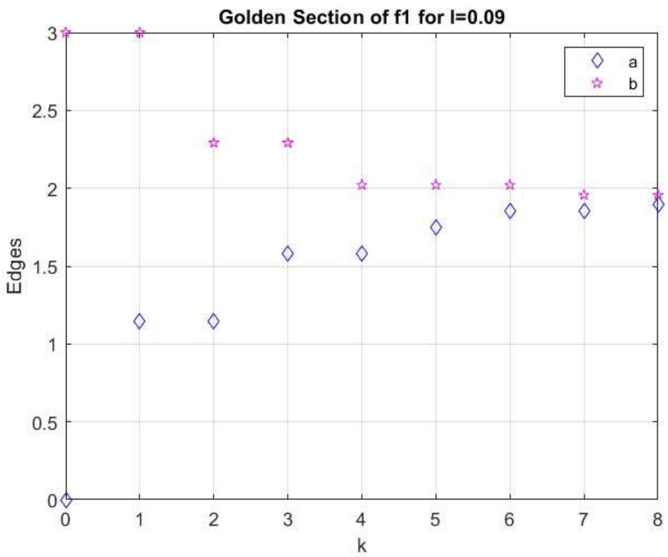


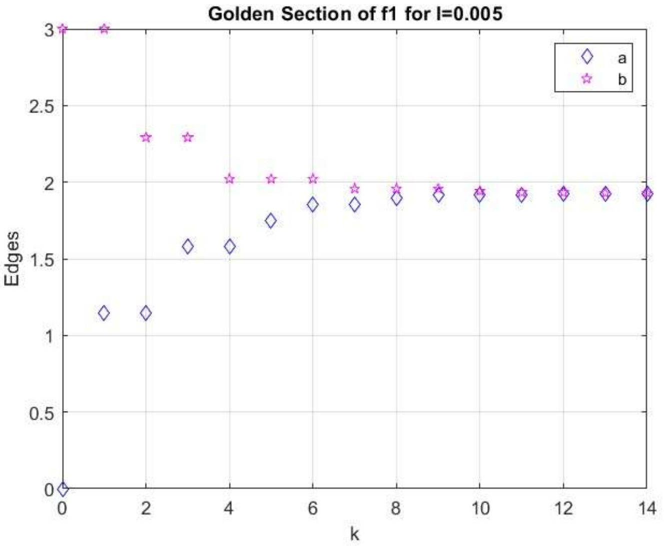
Για αυτή την μέθοδο οι συμπεριφορές των συναρτήσεων φαίνεται να είναι λίγο πιο περίπλοκες. Παρατηρούμε πώς η αύξηση του l δεν φέρνει πάντα λιγότερη ακρίβεια ενώ είναι πιθανό μια μικρότερη τιμη του l να δίνει χειρότερο αποτέλεσμα από μια μεγαλύτερη (όπως συμβαίνει στην f1)

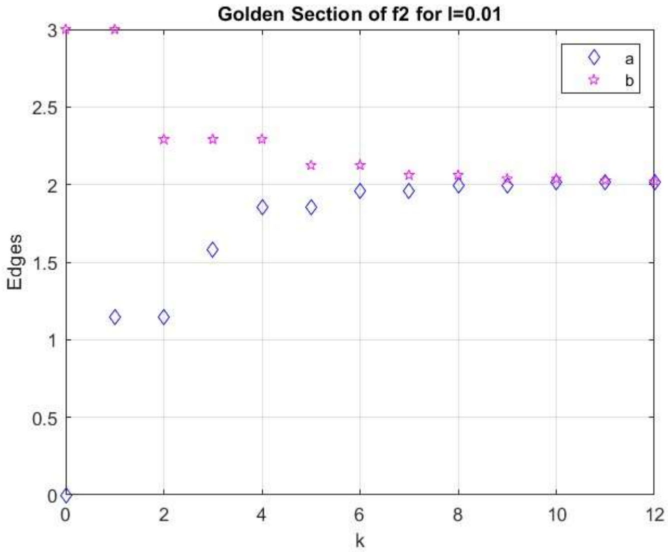
β. <u>Τις αλλαγές των α και β με βάση κάποια l</u>

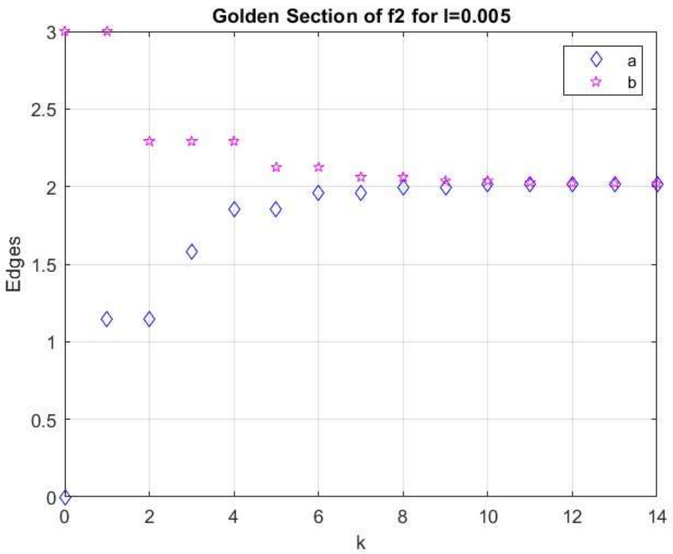
Συνεχίζουμε με τα l που επιλέξαμε και στην προηγούμενη μέθοδο και βρίσκουμε:

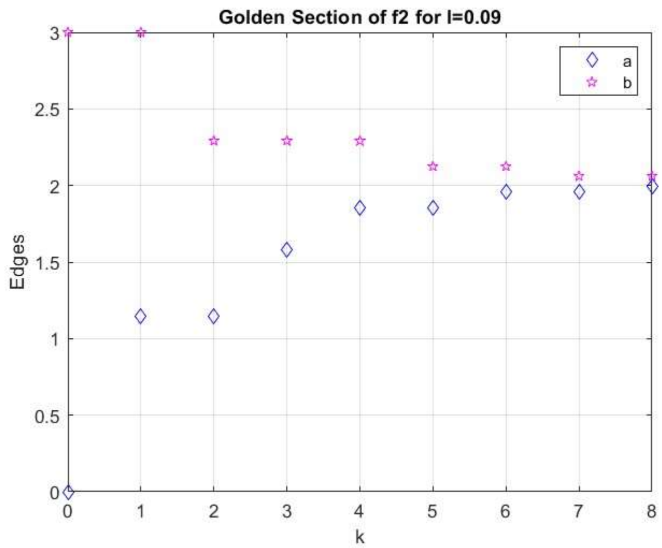


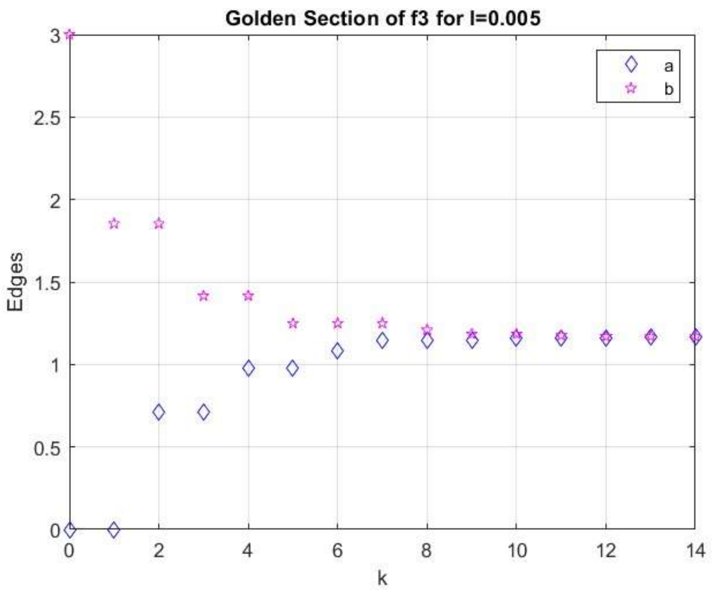


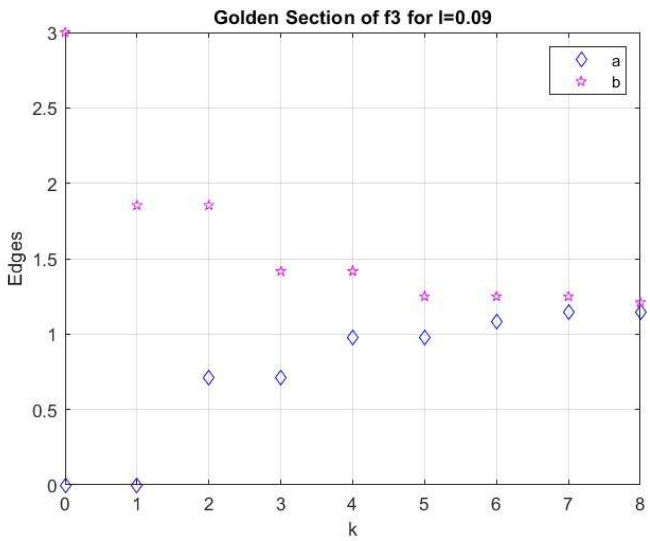

Golden Section of f1 for l=0.005
Edges
k
a
b

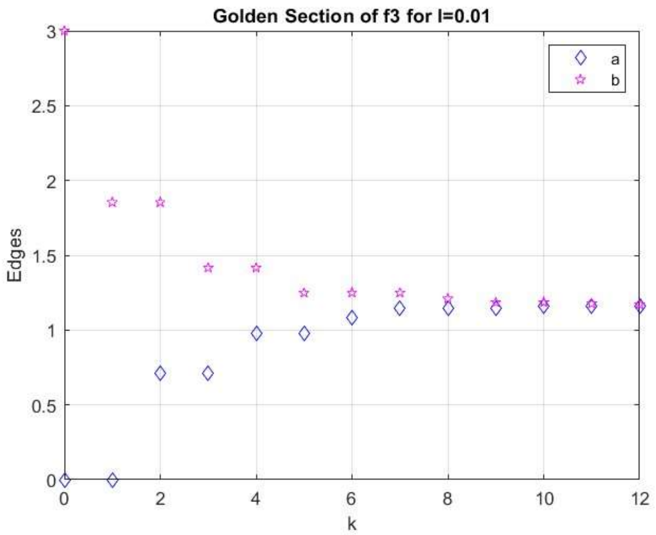

Golden Section of f2 for l=0.01
Edges
k
a
b


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


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


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


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



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
0.239394	0.123648	0.119556
0.248738	0.117495	0.143647
0.240024	0.116716	0.123573

Είναι ξεκάθαρη η βελτίωση, ιδιαίτερα για την f2 όπου ο χρόνος έπεσε σχεδόν κατά 0.45 ή 77%. 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 $Fn > (b - a)/l$. Αυτό σημαίνει πως, όπως θα φανεί παρακάτω, ο αριθμός των επαναλήψεων και για τις τρεις συναρτήσεις θα είναι ίδιος για συγκεκριμένο l.

Δυστυχώς τελευταία στιγμή παρατήρησα ένα λάθος που είχα κάνει στον κώδικα και δεν κατάφερα να το κάνω να λειτουργήσει σωστά μέχρι το τέλος της διορίας. Στο project είναι και αυτός ο κώδικας.

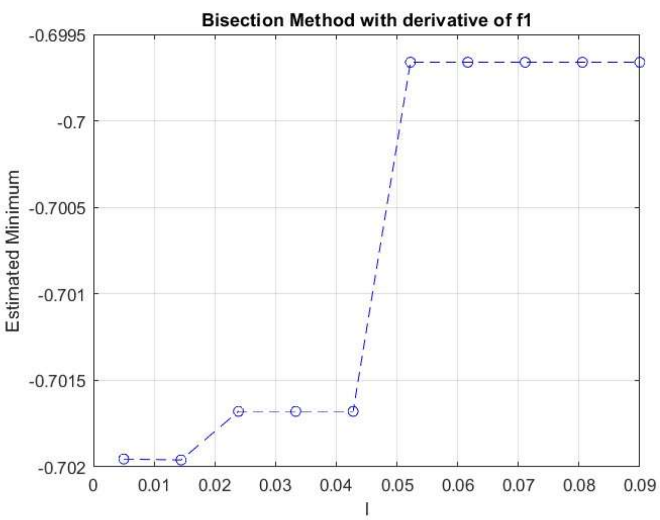
Μέθοδος Διχοτόμου με χρήση παραγώγου

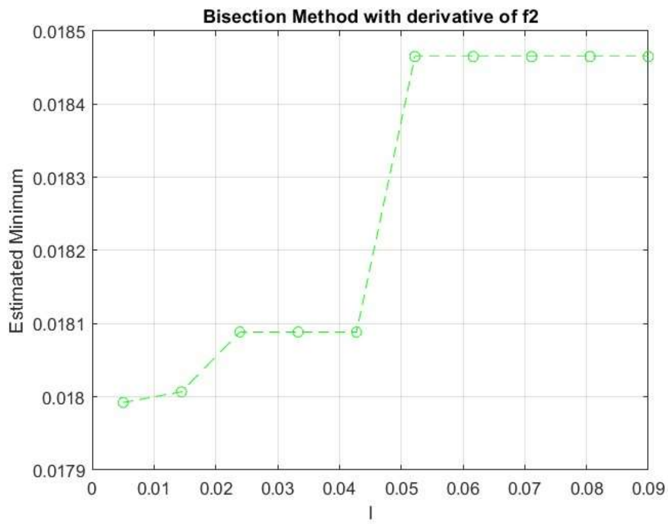
Αυτή η μέθοδος είναι παρόμοια με τη μέθοδο διχοτόμου αλλά χρησιμοποιεί την παράγωγο στην απόφαση του ποιο άκρο να αλλάξει.

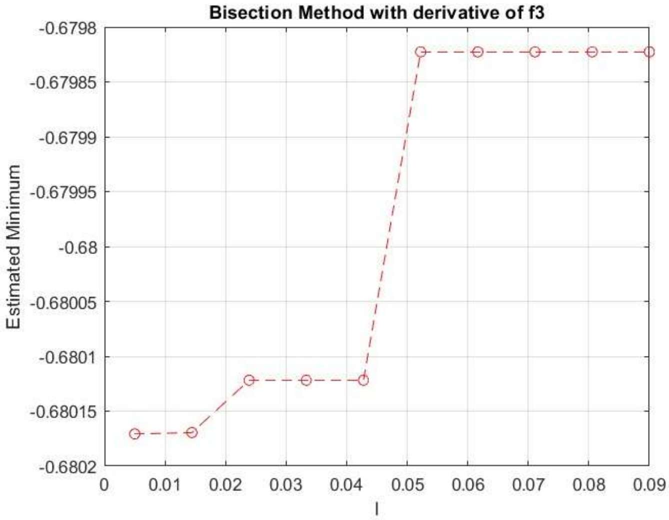
Τα αποτελέσματα που πήραμε είναι:
$f1(x) = -0.7019$ για $x = 1.9248$
$f2(x) = 0.0180$ για $x = 2.0186$
$f3(x) = -0.6792$ για $x = 1.1631$

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

α. <u>Τις αλλαγές του αποτελέσματος με βάση το l</u>

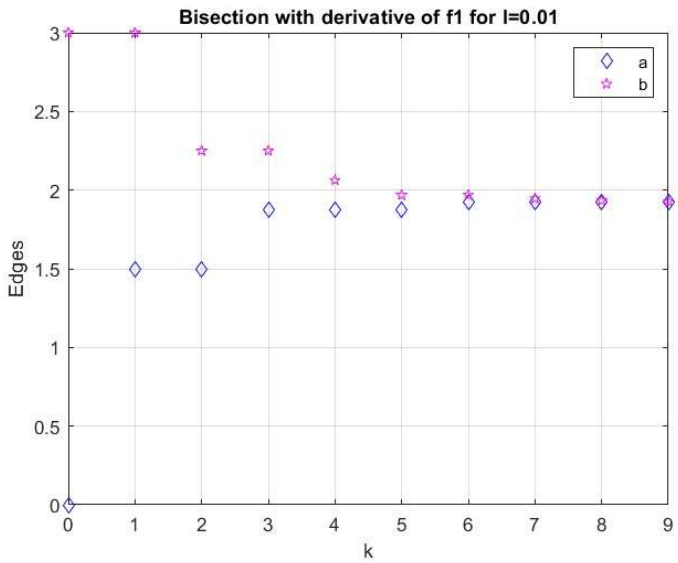


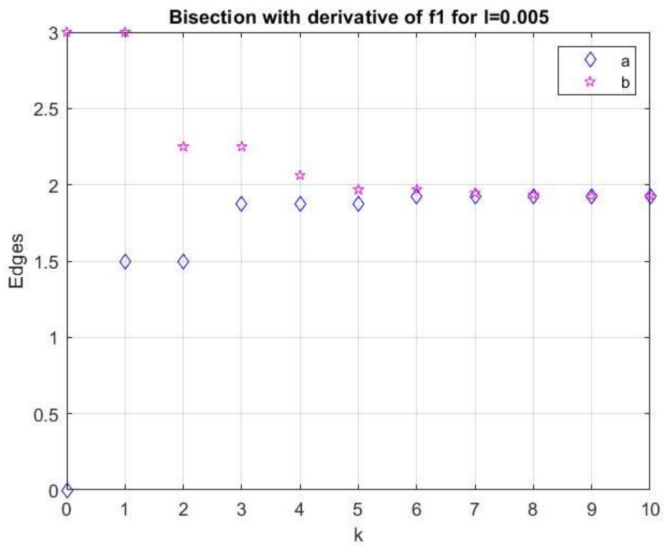


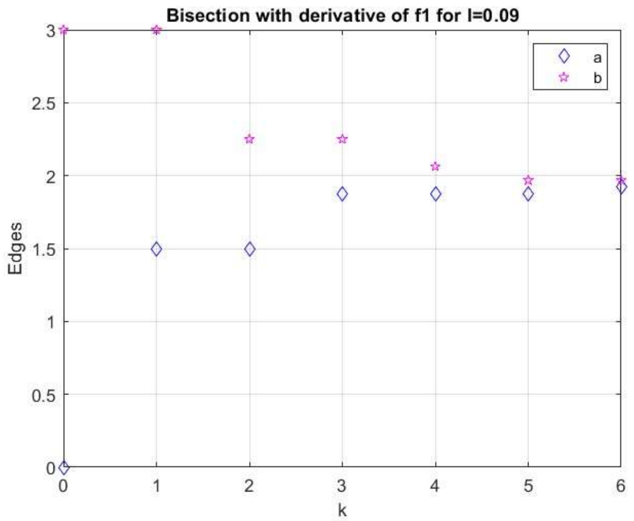


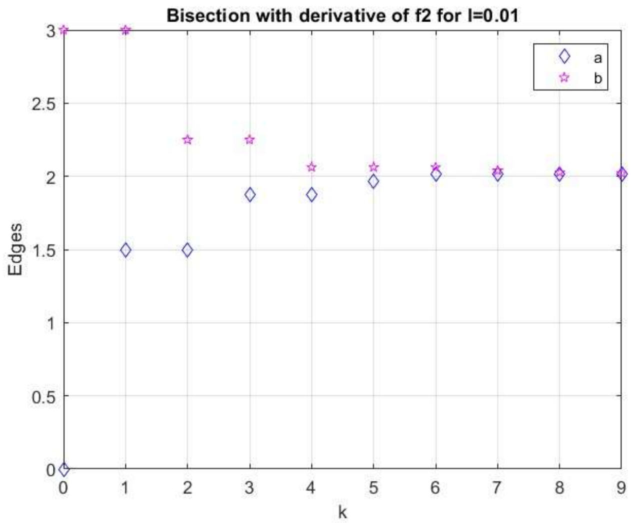
Παρατηρούμε πως η συμπεριφορά των συναρτήσεων ολόιδια, και με την αύξηση του l έχουμε και μια απόκλιση από την εκτιμώμενη τιμή που όμως είναι πάντα αυξανούμενη.

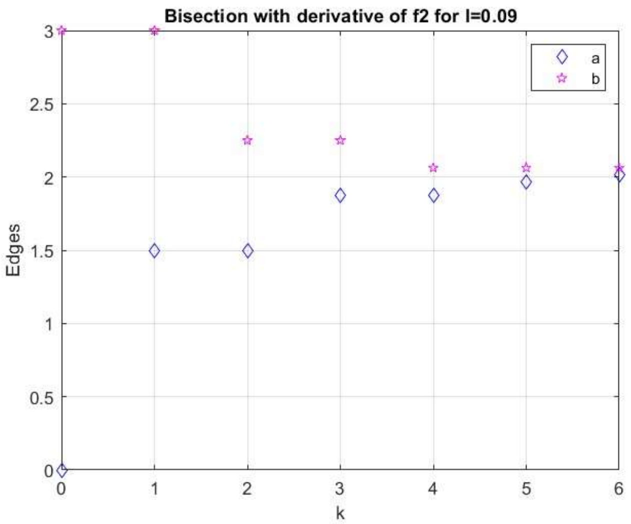
β. <u>Τις αλλαγές των α και β με βάση κάποια l</u>
Συνεχίζουμε με τα l που επιλέξαμε και στην προηγούμενη μέθοδο και βρίσκουμε:

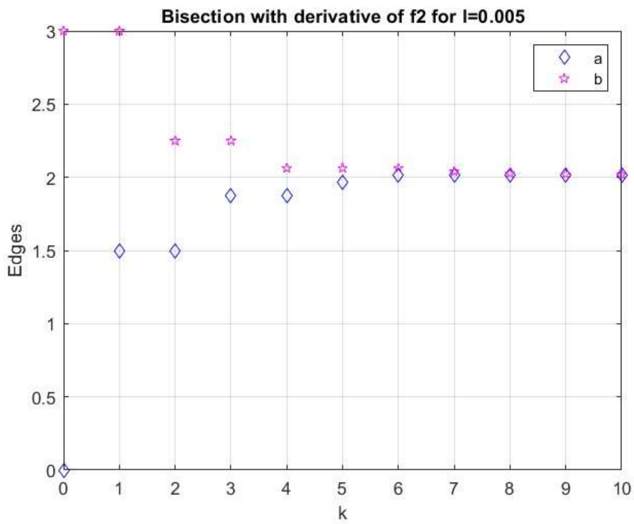


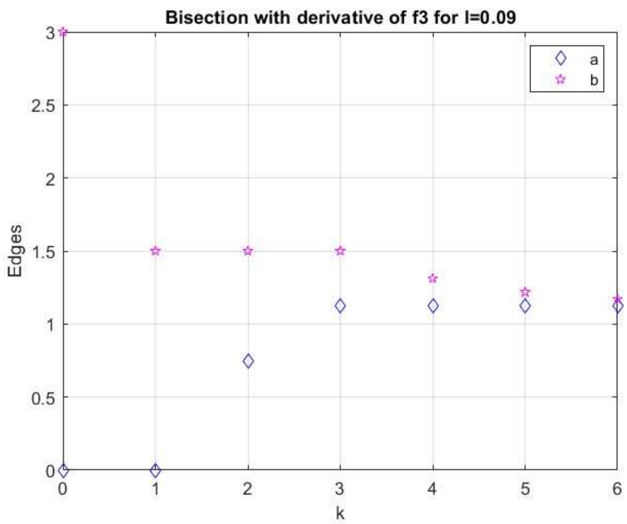


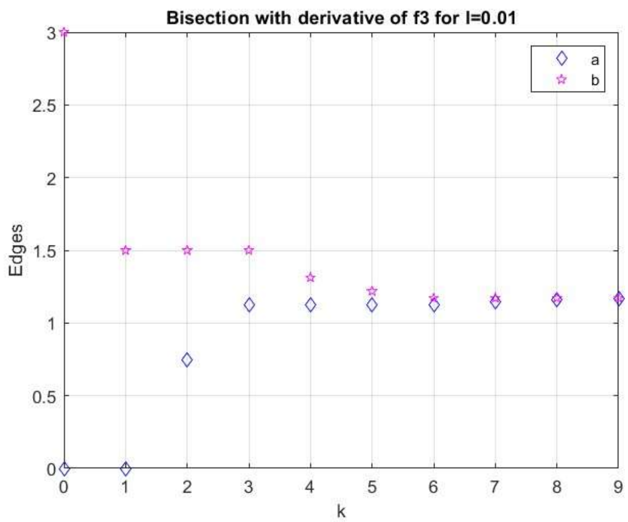

Bisection with derivative of f1 for l=0.09
Edges
k
a
b

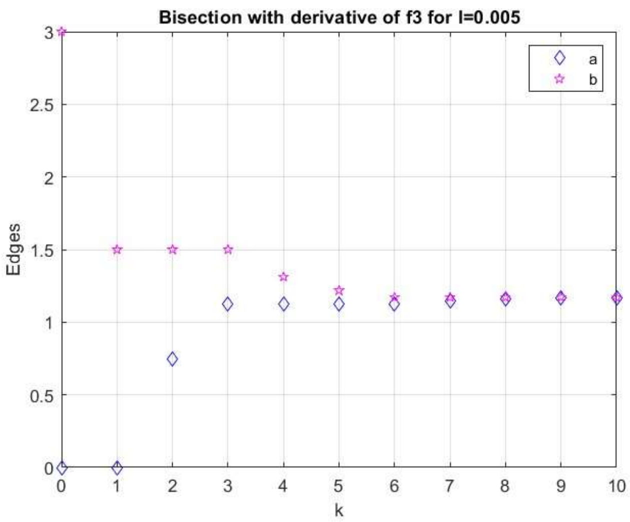

Bisection with derivative of f2 for l=0.01
Edges
k
a
b


Bisection with derivative of f2 for l=0.09
Edges
k
a
b


Bisection with derivative of f2 for l=0.005
Edges
k
a
b


Bisection with derivative of f3 for l=0.09
Edges
k
a
b


Bisection with derivative of f3 for l=0.01
Edges
k
a
b



Βλέπουμε πως φτάνει στο αποτέλεσμα μας με σχετικά λίγες επαναλήψεις. Όσον αφορά τον χρόνο:

f1	f2	f3
0.205710	0.081878	0.083561
0.210710	0.083003	0.089224
0.201925	0.081580	0.082676

Ο χρόνος εκτέλεσης είναι εμφανέστατα καλύτερος πράγμα που κάνει τη μέθοδο της διχοτόμου με παράγωγο την γρηγορότερη από όσες είδαμε μέχρι στιγμής. Το μόνο σημαντικό μειωνέκτημα 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.