ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Βελτιστοποίησης

1η Εργαστηριακή Άσκηση

Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής σε δοσμένο διάστημα

- Ονοματεπώνυμο: Κυπριανίδης Άρης-Ευτύχιος
- ΑΕΜ: 10086
- Email: akyprian@ece.auth.gr

ΘΕΣΣΑΛΟΝΙΚΗ 2022

Συναρτήσεις

1) $f(x) = (x-2)^2 + x\ln(x+3)$
2) $f(x) = 5^x + (2 - cos(x))^2$
3) $f(x) = e^x(x^3 - 1) + (x-1)\,sin(x)$

Για να χρησιμοποιήσουμε τις αλγορίθμους, πρέπει οι συναρτήσεις να είναι κυρτές στο [-1,3], δηλαδή η δεύτερη παράγωγος να είναι θετική.

Οι δεύτεροι παράγωγοι είναι:

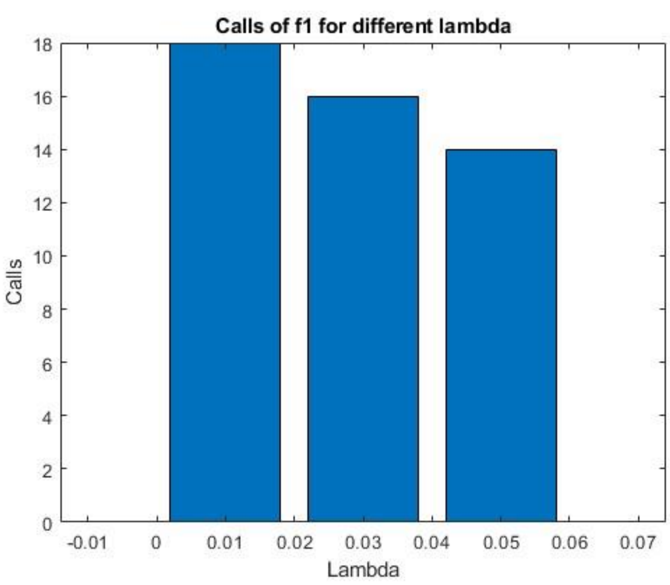
1) $\frac{2x^2+13x+24}{(x+3)^2}$
2) $\ln^2(5)5^x + 2(\sin^2(x) + \cos(x)(2-\cos(x)))$
3) $e^x x^3 + 6e^x x^2 + 6e^x x - xsin(x) - e^x + 2\cos(x) + \sin(x)$

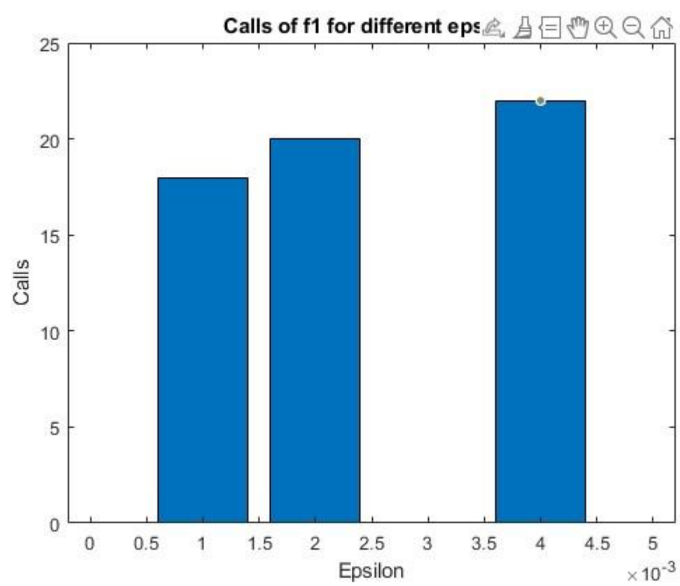
Όλοι οι παράγωγοι είναι θετικοί στο διάστημα [-1,3], οπότε οι συναρτήσεις είναι κυρτές.

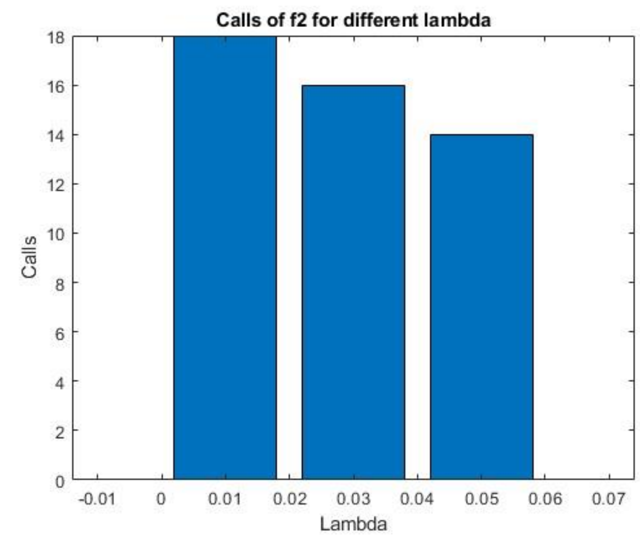
Θέμα 1: Μέθοδος διχοτόμου

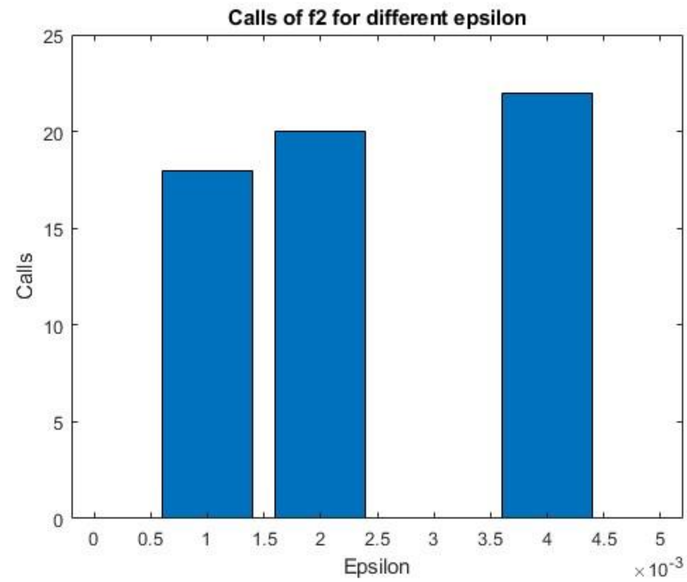
Αρχεία: dixotomou.m, thema_1.m

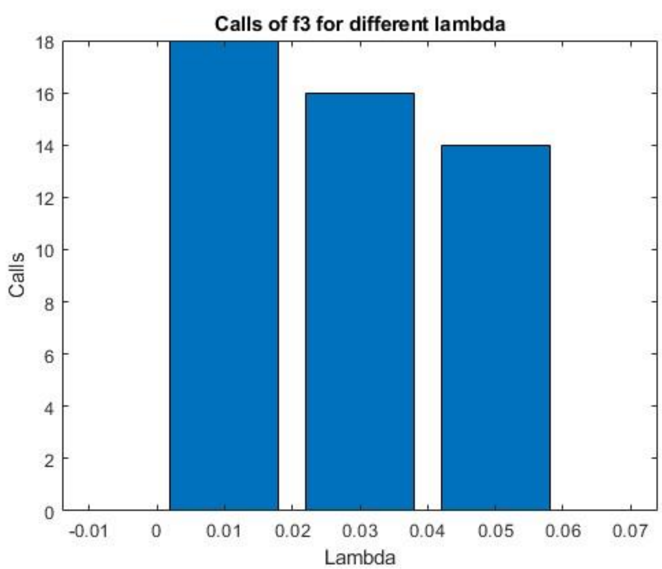
Οι υπολογισμοί της αντικειμενικής συνάρτησης για διάφορα λάμδα και έψιλον:

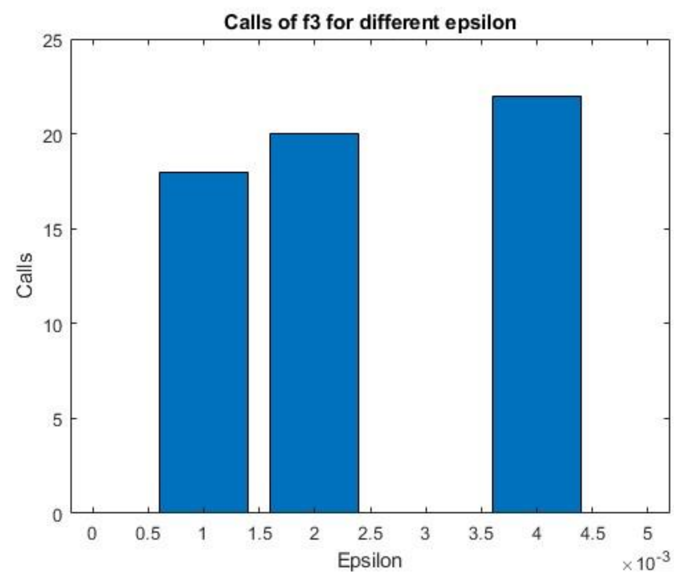






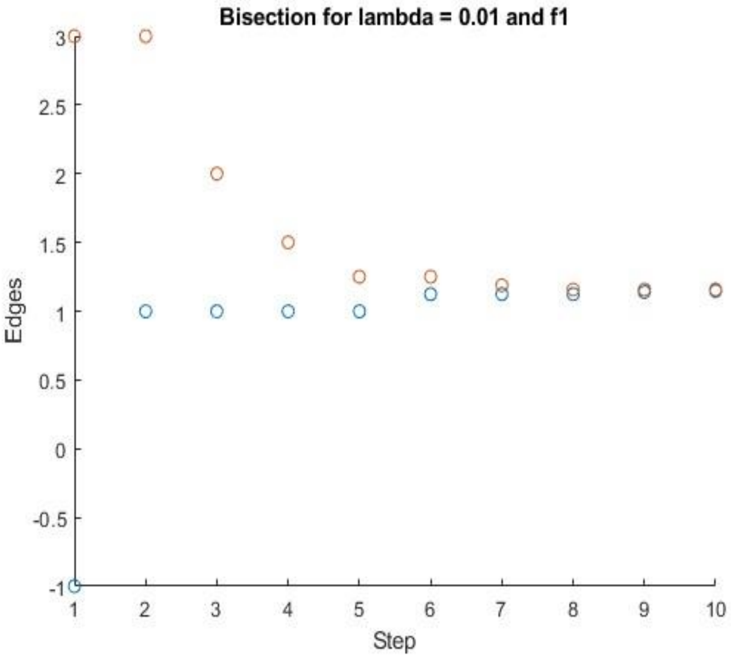


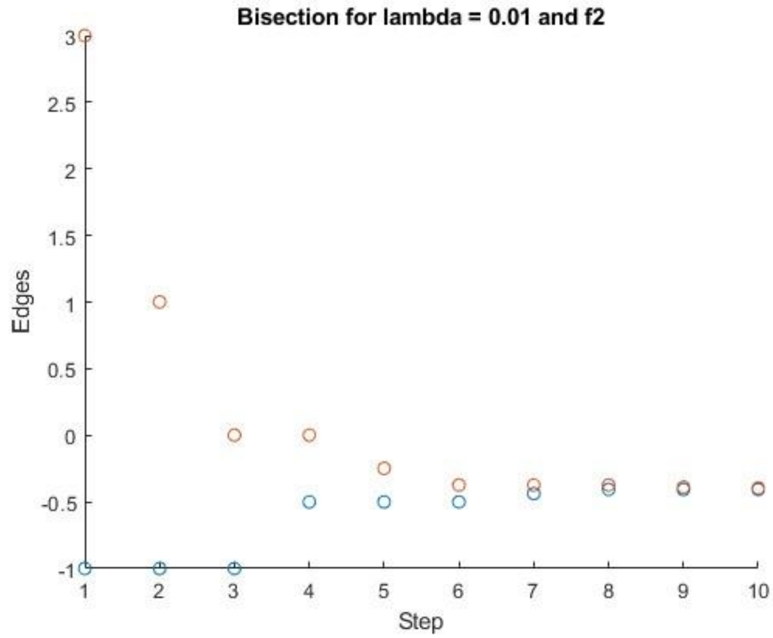


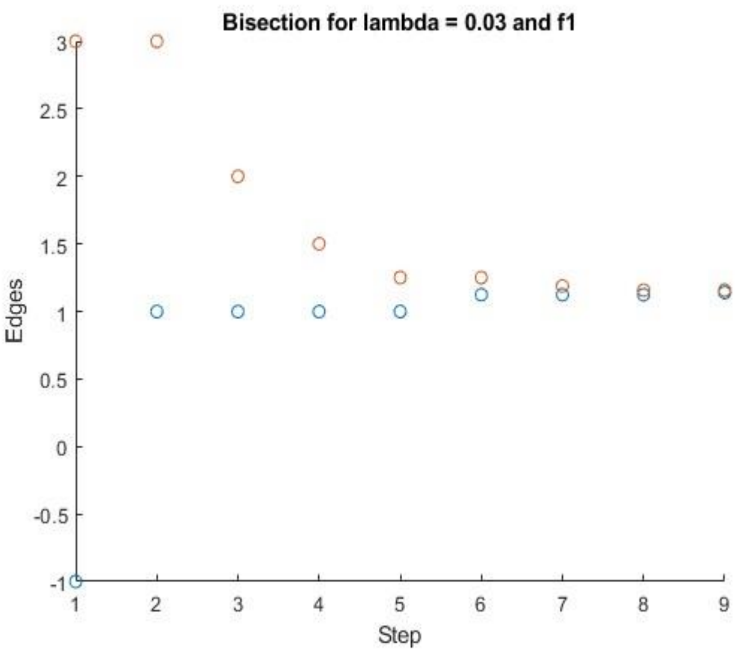


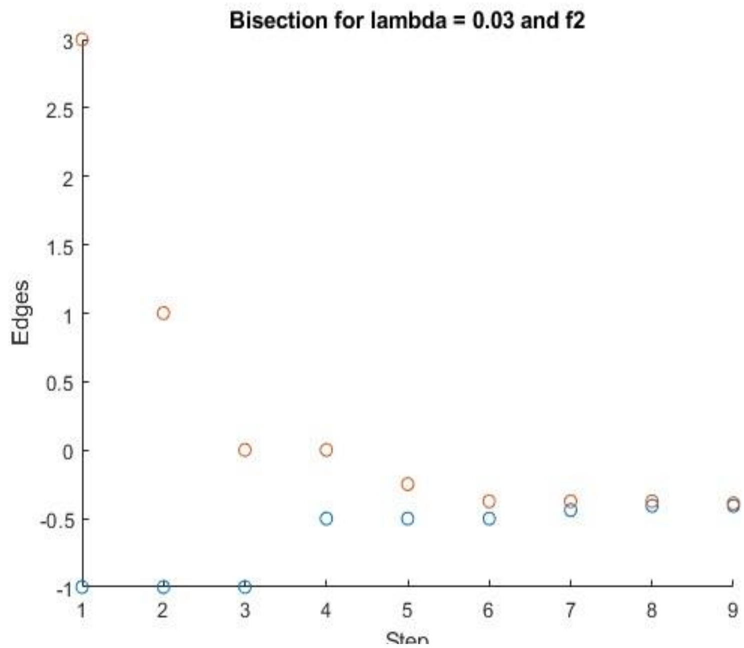
Βλέπουμε πως για μικρότερες τιμές λάμδα, έχουμε περισσότερες κλήσεις της αντικειμενικής, και το αντίθετο συμβαίνει για τις τιμές του έψιλον. Τα λάμδα που χρησιμοποιήθηκαν ήταν τα [0.01 0.03 0.05] και τα έψιλον [0.001 0.002 0.004].

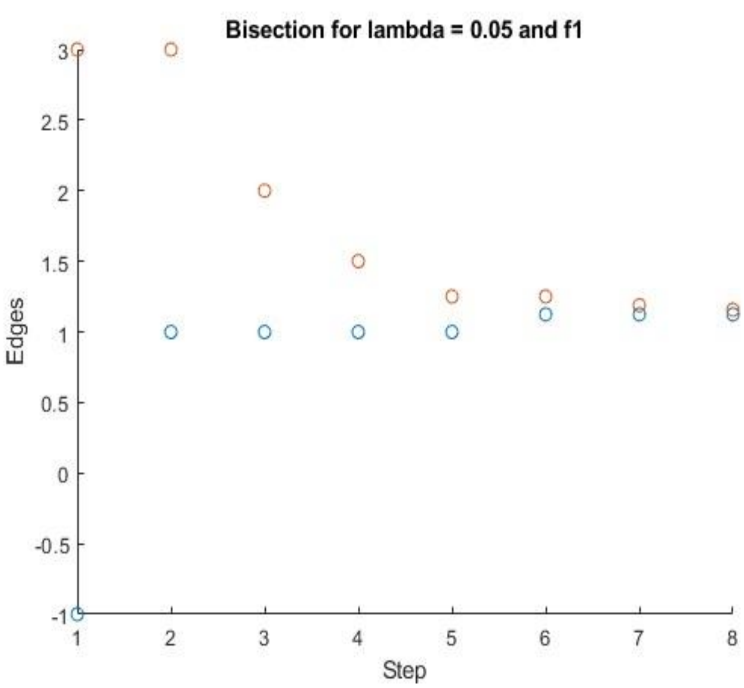
Συνεχίζοντας στις τιμές των άκρων συναρτήσει του βήματος k :

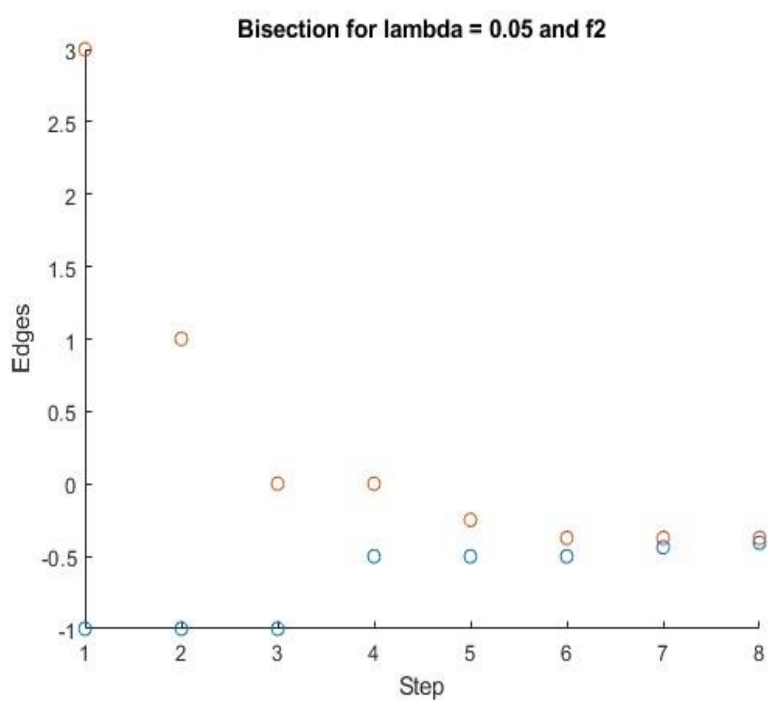

Bisection for lambda = 0.01 and f1
Edges
Step

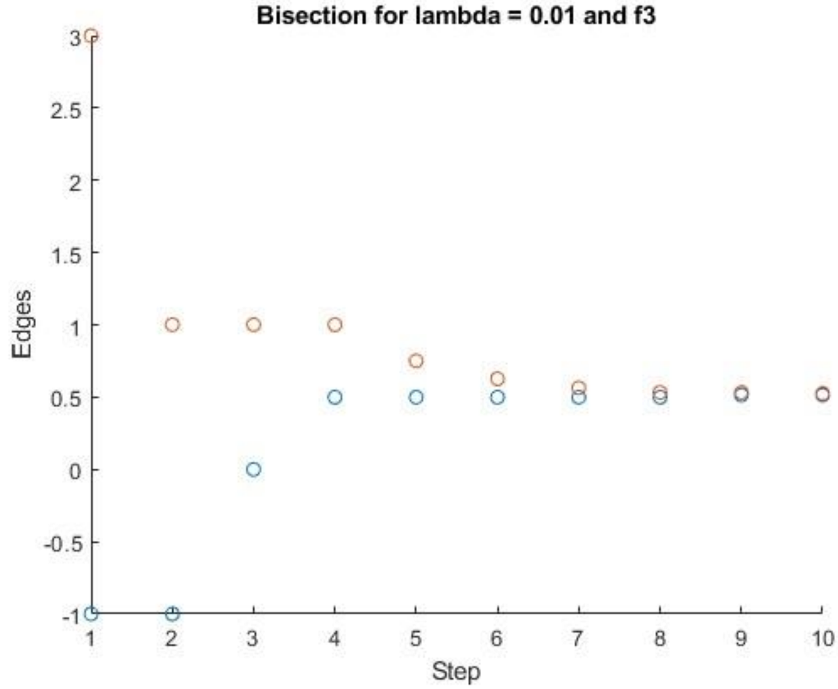

Bisection for lambda = 0.01 and f2
Edges
Step


Bisection for lambda = 0.03 and f1
Edges
Step

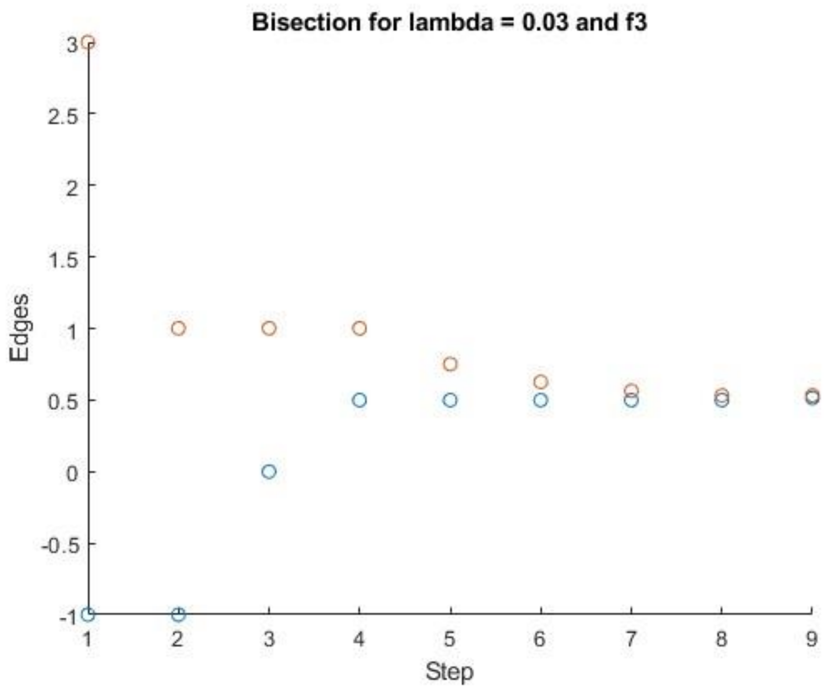

Bisection for lambda = 0.03 and f2
Edges


Bisection for lambda = 0.05 and f1
Edges
Step

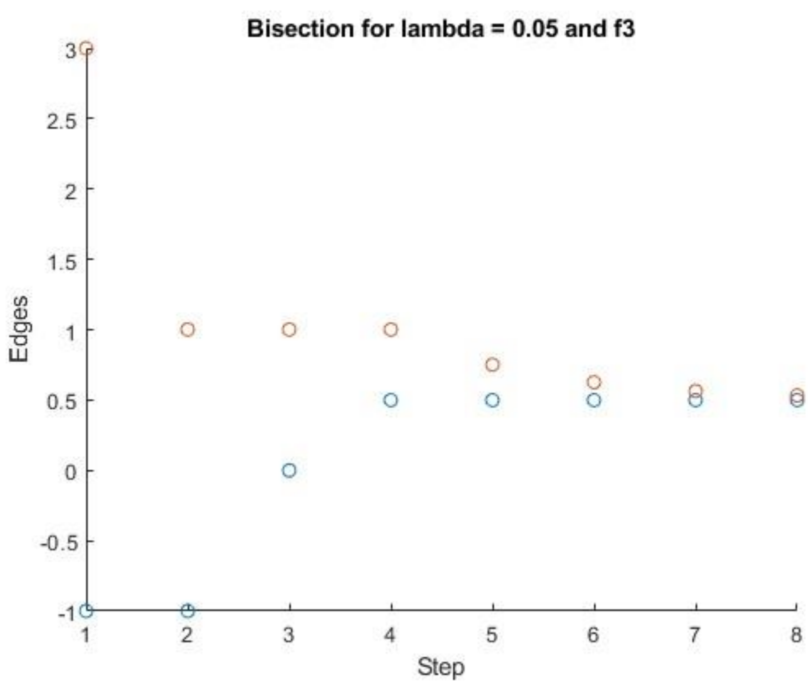

Bisection for lambda = 0.05 and f2
Edges
Step



Bisection for lambda = 0.01 and f3
Edges
Step



Bisection for lambda = 0.03 and f3
Edges
Step



Bisection for lambda = 0.05 and f3
Edges
Step

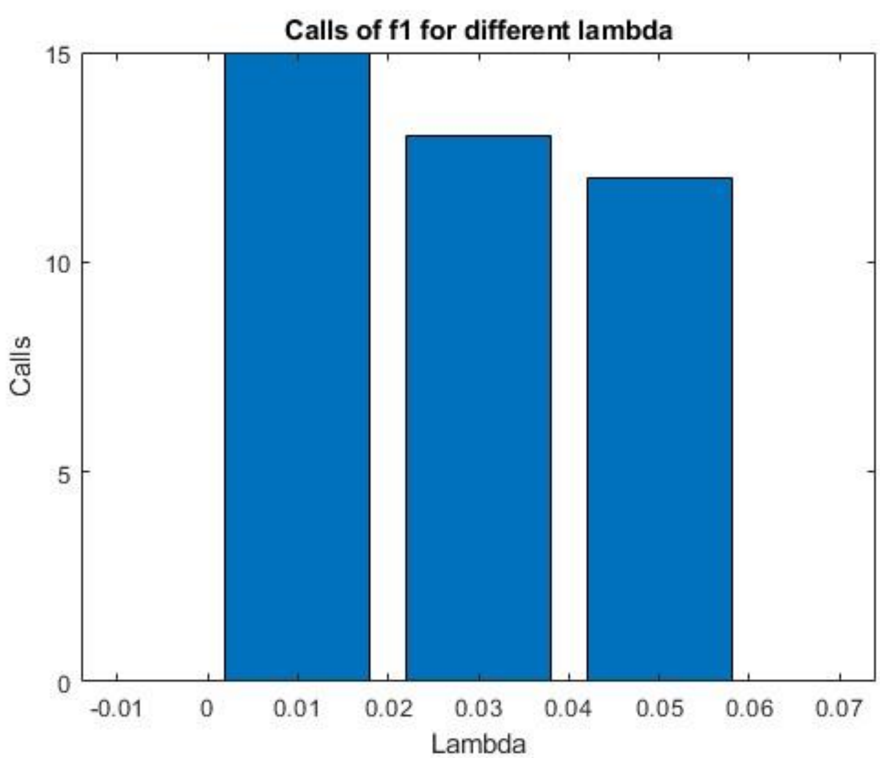
Είναι εμφανές πως για μεγαλύτερα λάμδα, το διάστημα που βρίσκεται το ελάχιστο είναι μεγαλύτερο, αλλά κερδίζουμε μερικές επαναλήψεις στον αλγόριθμο. Οπότε, χάνουμε σε ακρίβεια για να τρέξει λίγο γρηγορότερα ο αλγόριθμος. Τα διαγράμματα έγιναν σαν ($α_κ$, $β_κ$, k) για να φαίνεται ότι συγκλίνουν στην ίδια τιμή.

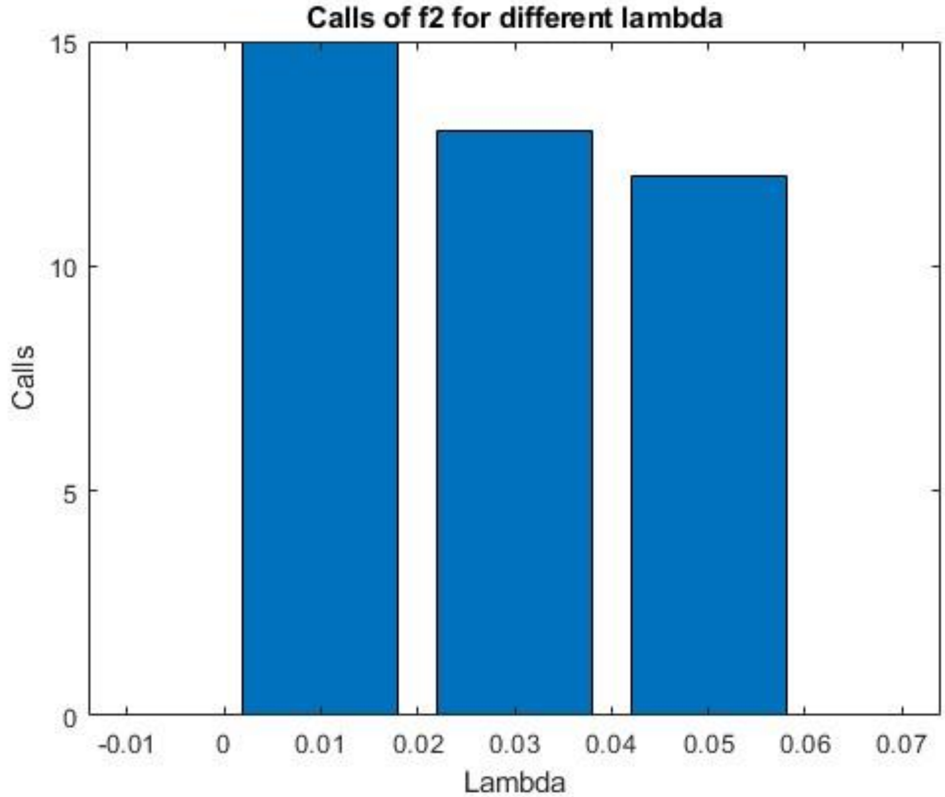
Σχόλια: σε χρόνο τρέχει σχετικά γρήγορα, αλλά φαίνεται πως καλεί πολλές φορές την αντικειμενική συνάρτηση, σε σχέση με τους άλλους αλγορίθμους.

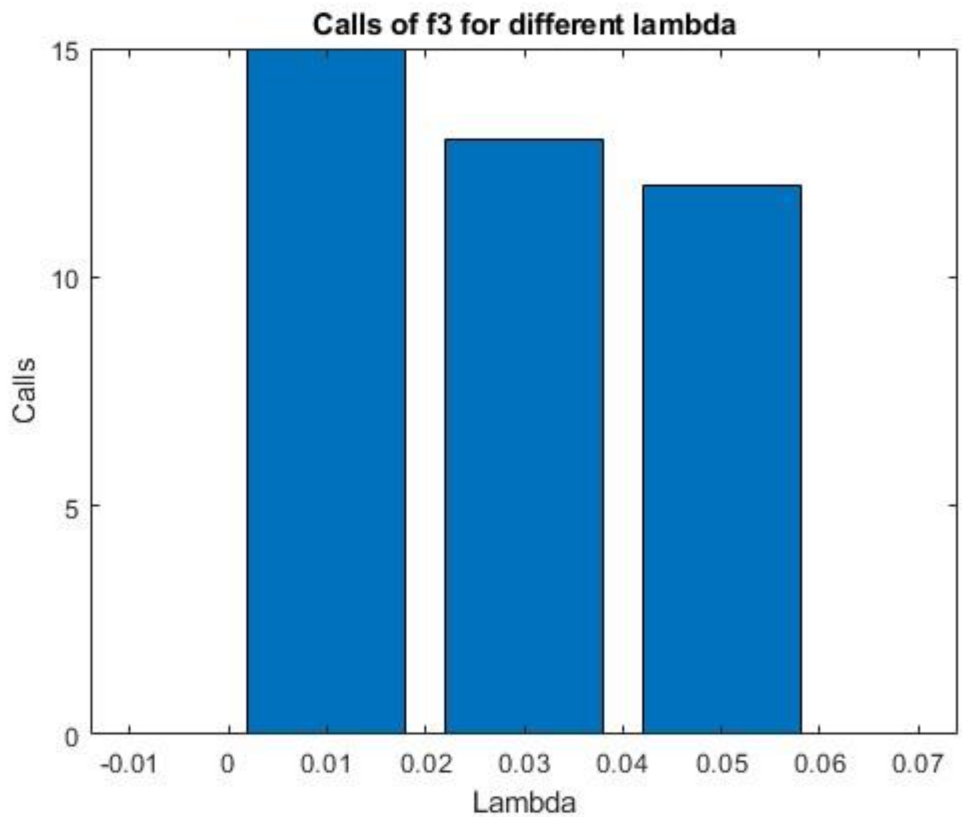
Θέμα 2: Μέθοδος χρυσής τομής

Αρχεία: golden_section.m, thema_2.m

Οι υπολογισμοί της αντικειμενικής συνάρτησης για διάφορα λάμδα:

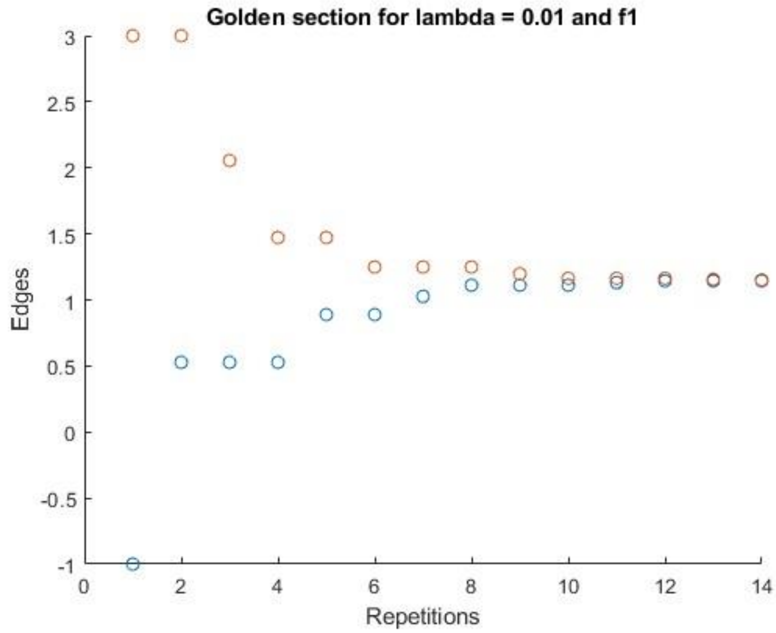


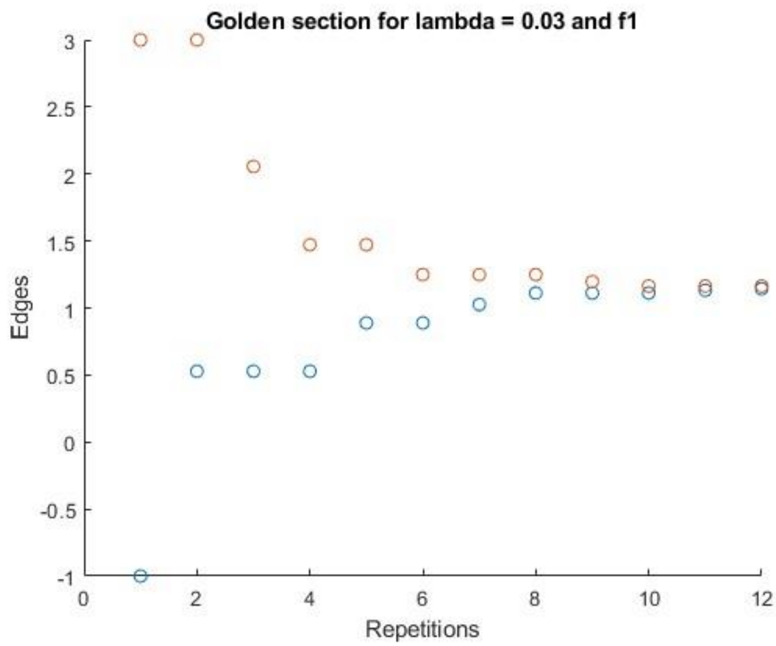


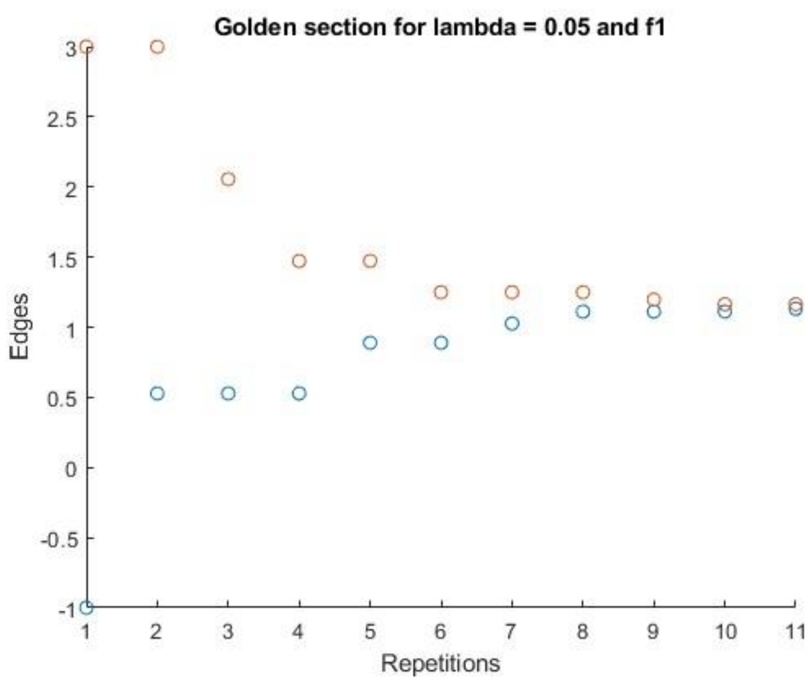


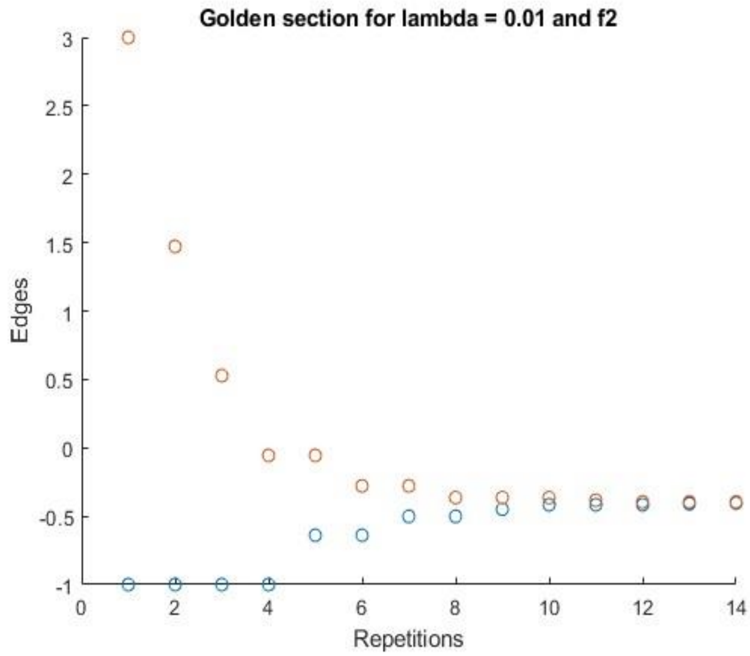
Όπως και στο προηγούμενο, για αυξανόμενες τιμές του λάμδα, η αντικειμενική καλείται λιγότερες φορές.

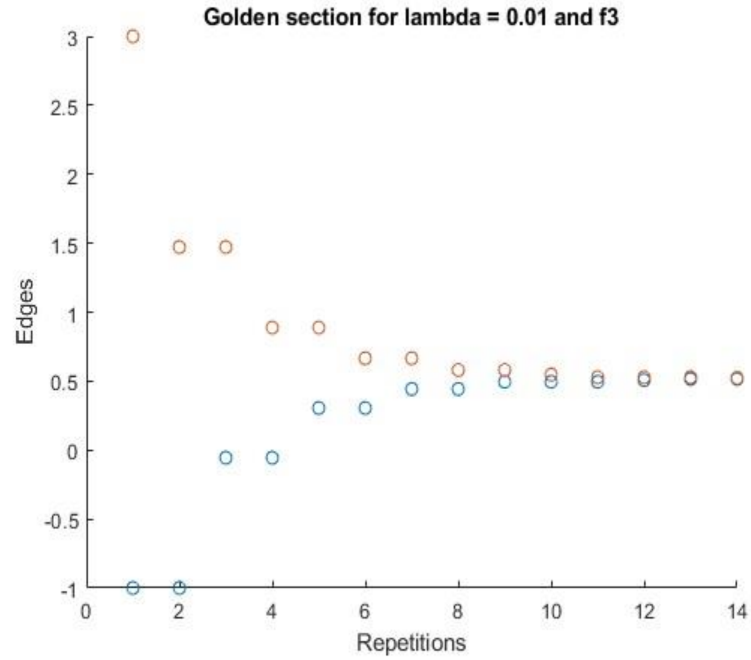
Συνεχίζοντας για τα άκρα του διαστήματος:

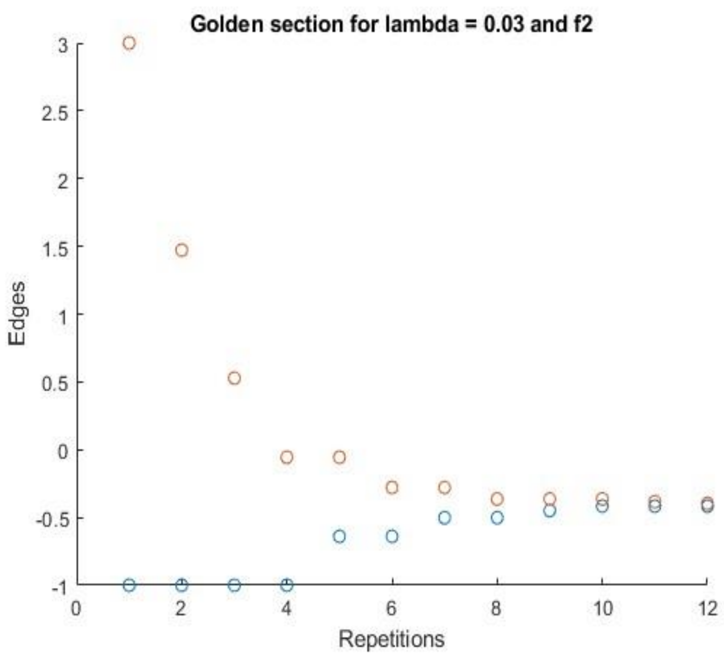

Golden section for lambda = 0.01 and f1
Edges
Repetitions

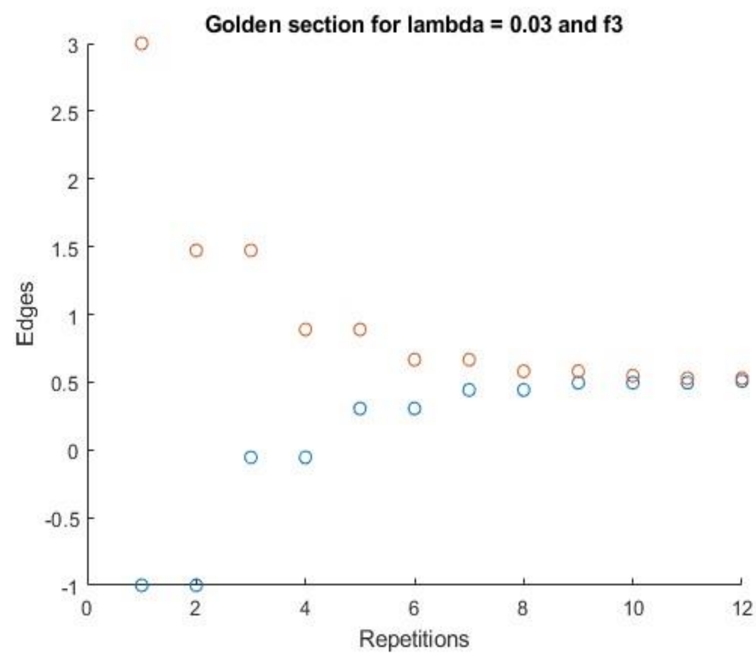

Golden section for lambda = 0.03 and f1
Edges
Repetitions

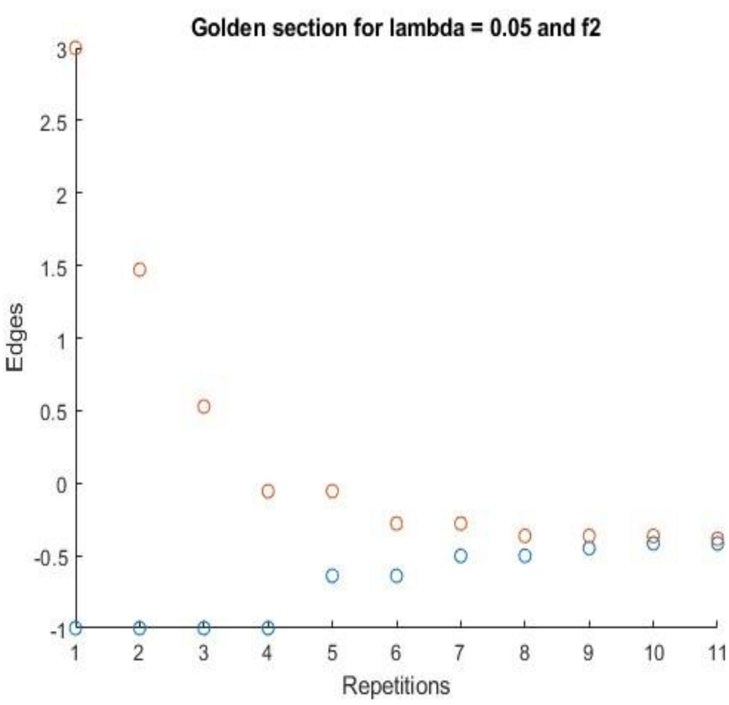

Golden section for lambda = 0.05 and f1
Edges
Repetitions

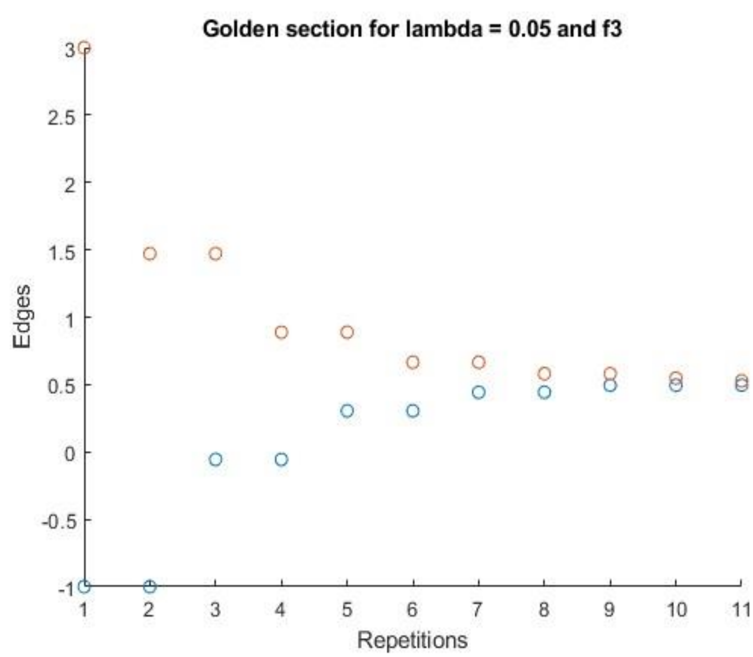

Golden section for lambda = 0.01 and f2
Edges
Repetitions


Golden section for lambda = 0.01 and f3
Edges
Repetitions


Golden section for lambda = 0.03 and f2
Edges
Repetitions


Golden section for lambda = 0.03 and f3
Edges
Repetitions


Golden section for lambda = 0.05 and f2
Edges
Repetitions


Golden section for lambda = 0.05 and f3
Edges
Repetitions

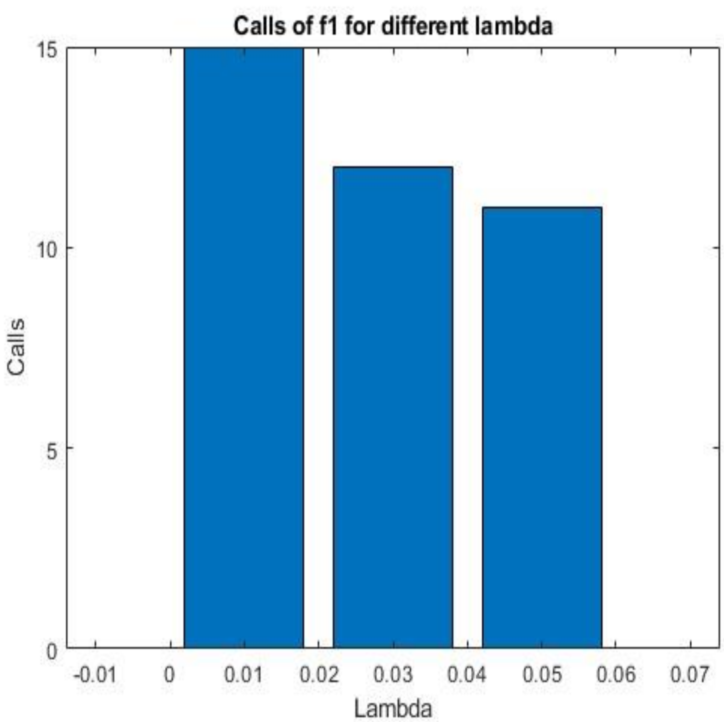
Όπως και πριν, για μεγαλύτερα λάμδα, το διάστημα που βρίσκεται το ελάχιστο είναι μεγαλύτερο, αλλά κερδίζουμε μερικές επαναλήψεις στον αλγόριθμο. Οπότε, χάνουμε σε ακρίβεια για να τρέξει λίγο γρηγορότερα ο αλγόριθμος.

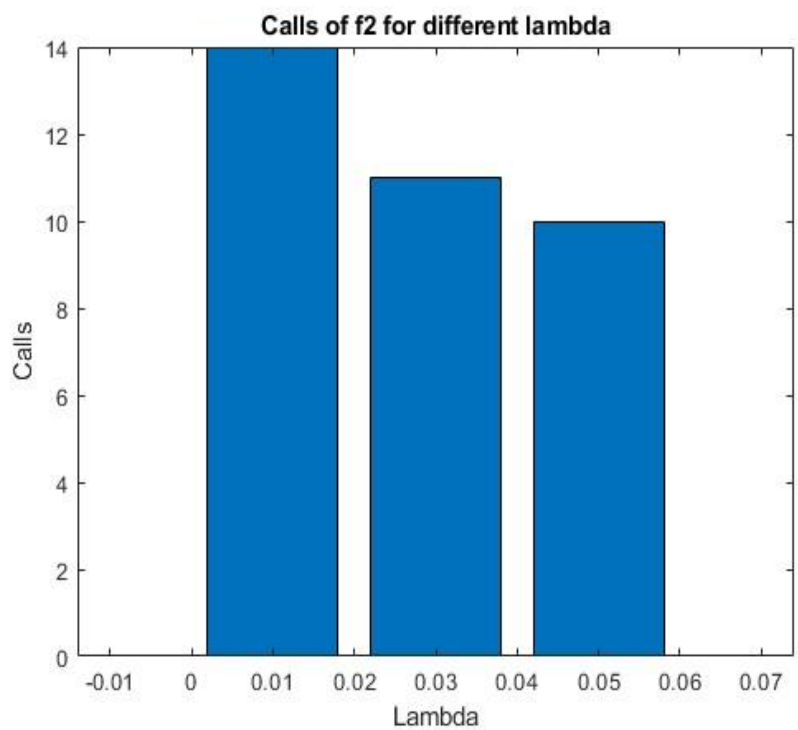
Σχόλια: γρήγορος αλγόριθμος με σχετικά λίγες κλήσεις αλλά αρκετές επαναλήψεις.

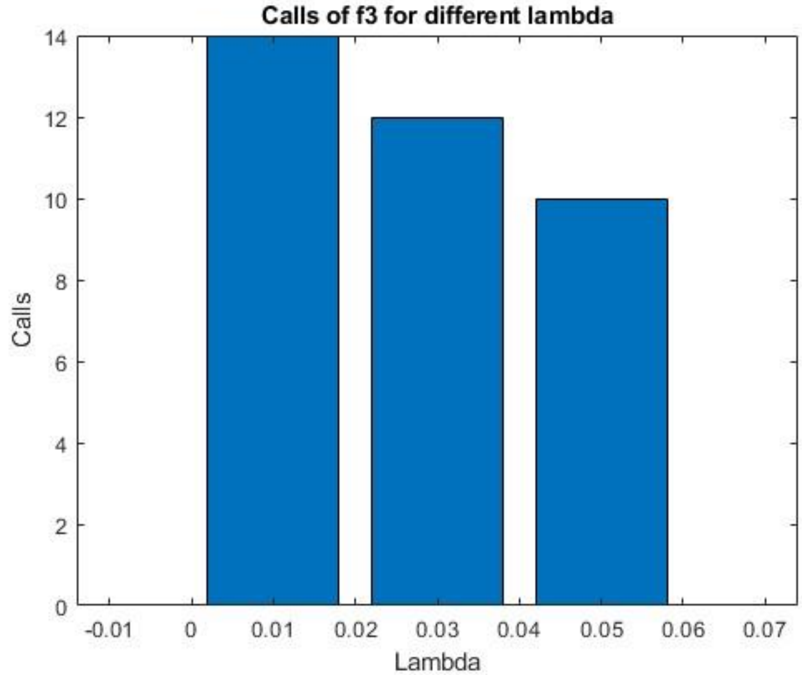
Θέμα 3: Μέθοδος Fibonacci

Αρχεία: FibonacciMethod.m, thema_3.m

Οι υπολογισμοί της αντικειμενικής συνάρτησης για διάφορα λάμδα:

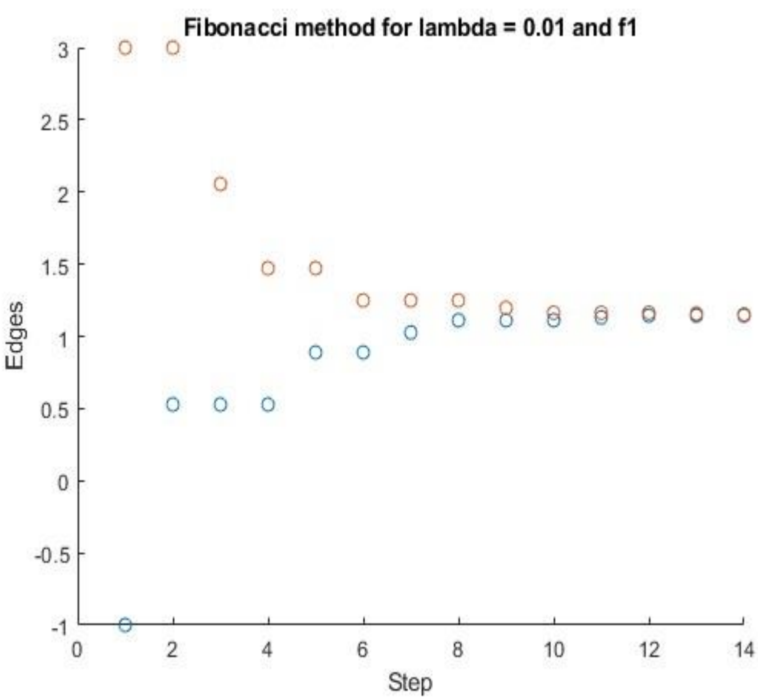


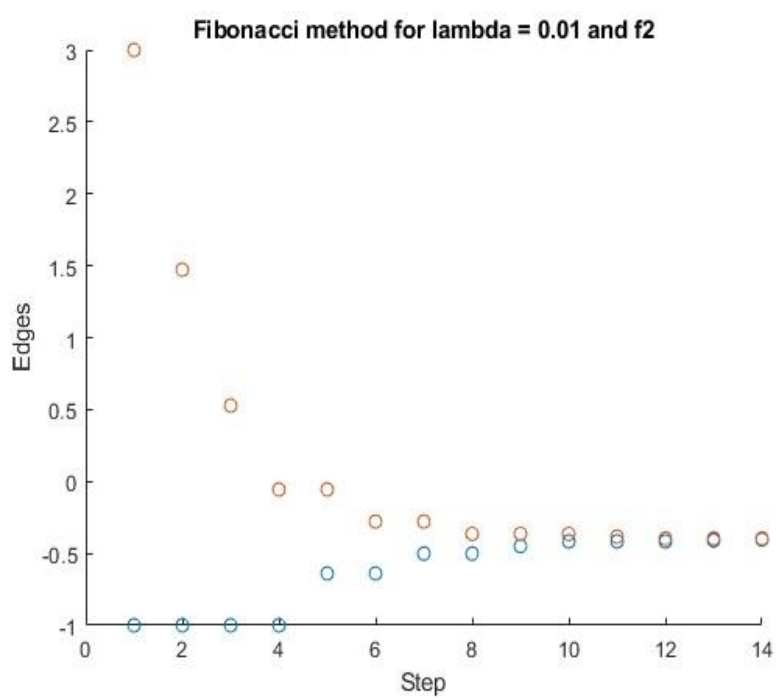


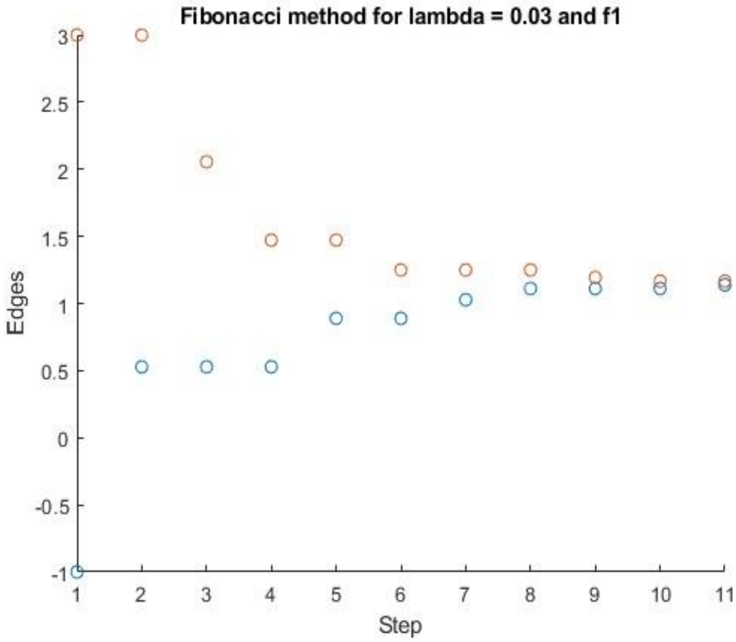


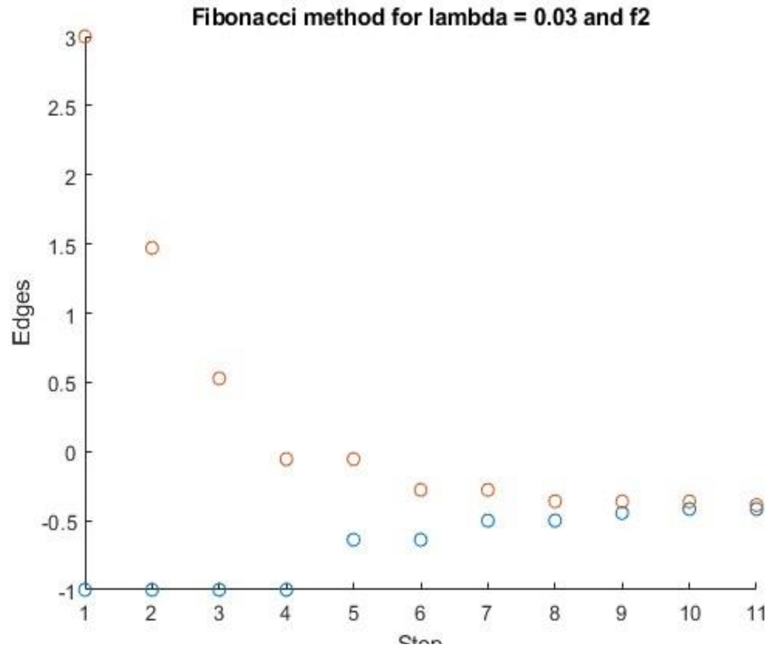
Ξανά, για αυξανόμενες τιμές του λάμδα, η αντικειμενική καλείται λιγότερες φορές.

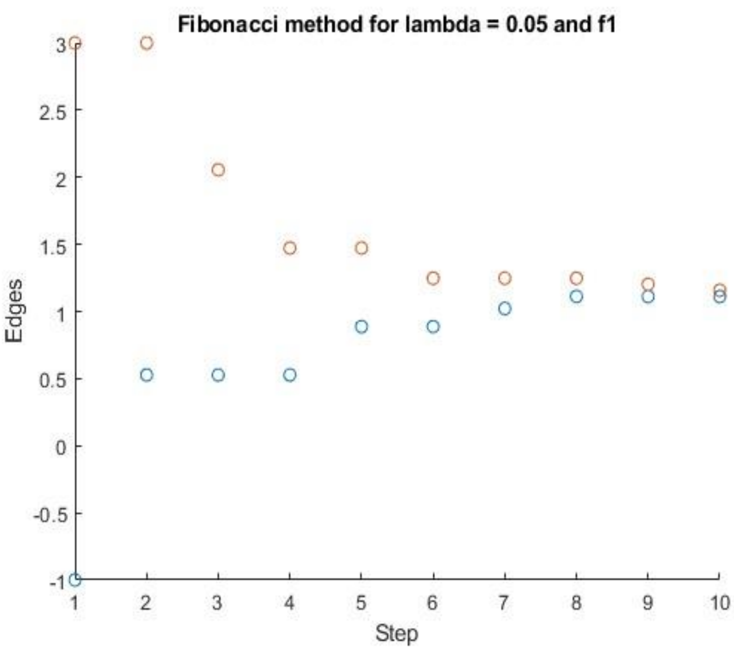
Συνεχίζοντας για τα άκρα του διαστήματος:

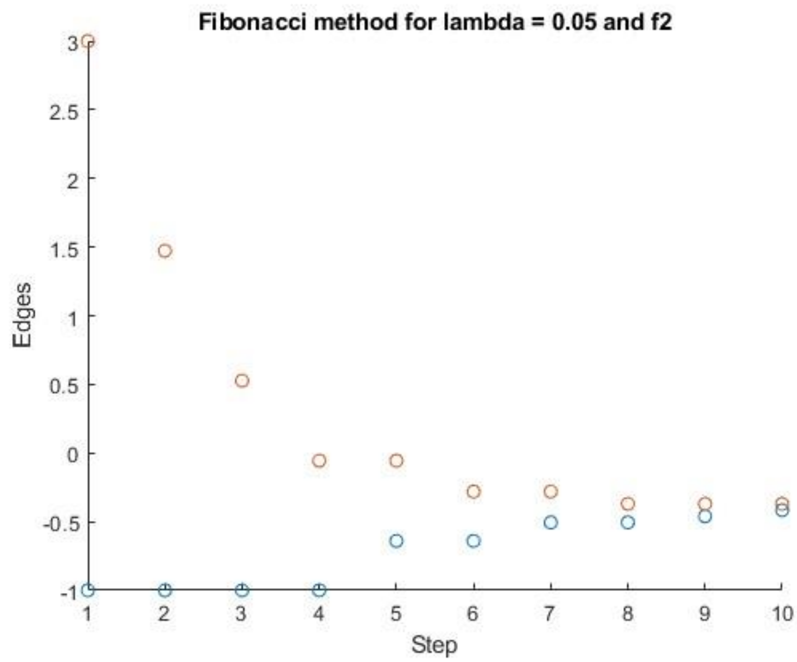


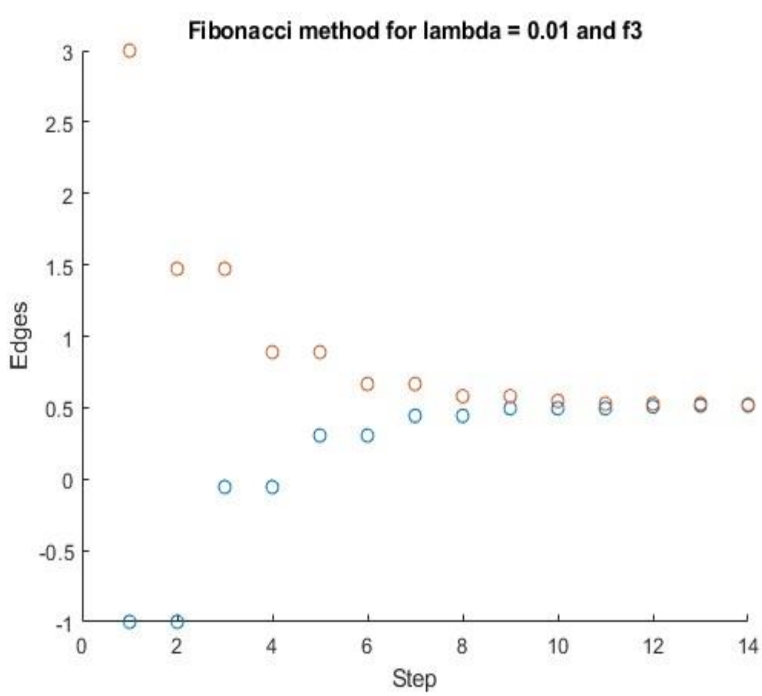


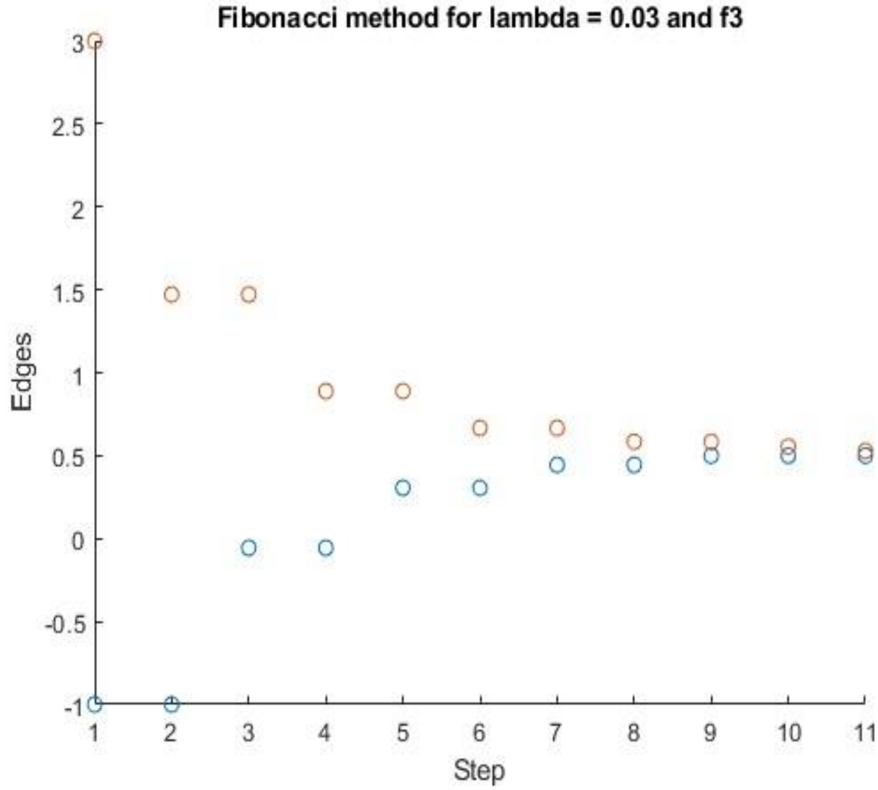

Fibonacci method for lambda = 0.03 and f1
Edges
Step

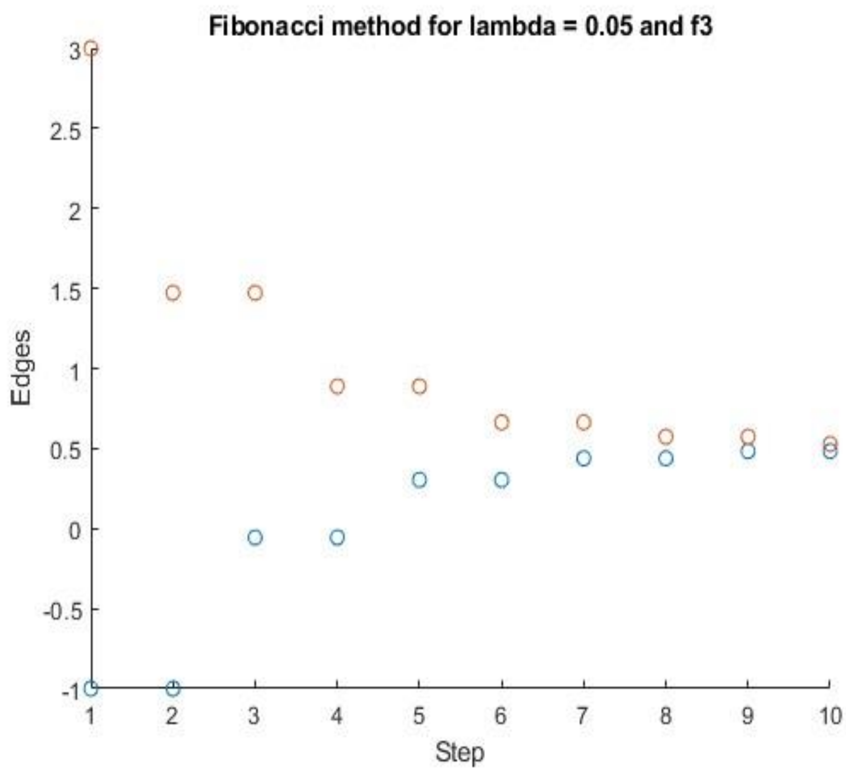

Fibonacci method for lambda = 0.03 and f2
Edges


Fibonacci method for lambda = 0.05 and f1
Edges
Step


Fibonacci method for lambda = 0.05 and f2
Edges
Step


Fibonacci method for lambda = 0.01 and f3
Edges
Step





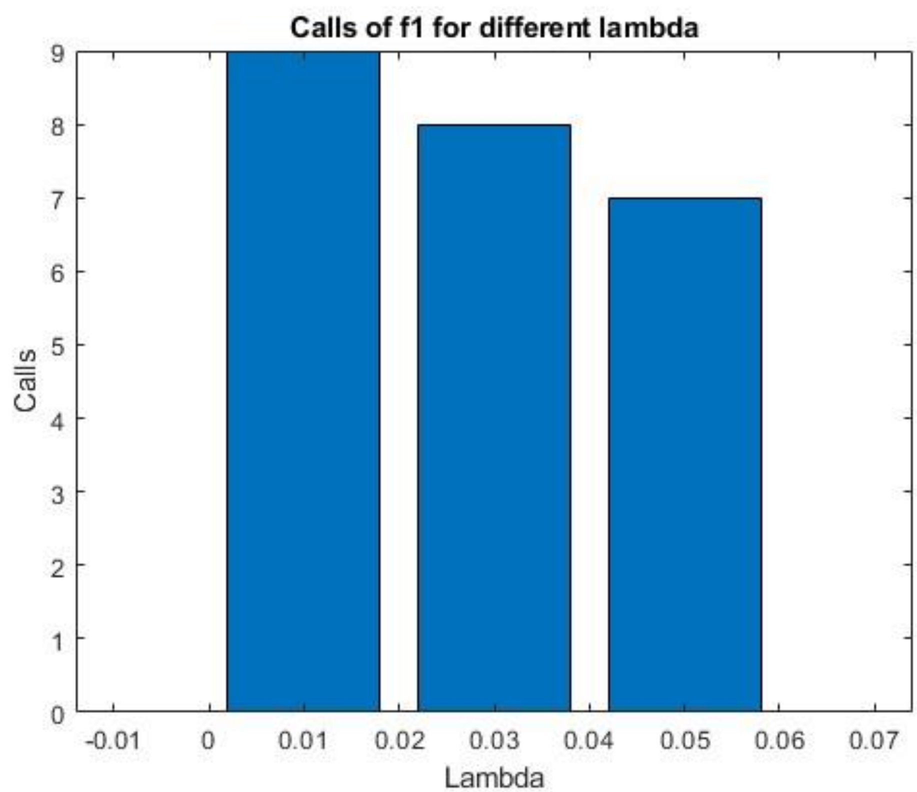
Όπως και πριν, για μεγαλύτερα λάμδα, το διάστημα που βρίσκεται το ελάχιστο είναι μεγαλύτερο, αλλά κερδίζουμε μερικές επαναλήψεις στον αλγόριθμο. Οπότε, χάνουμε σε ακρίβεια για να τρέξει λίγο γρηγορότερα ο αλγόριθμος.

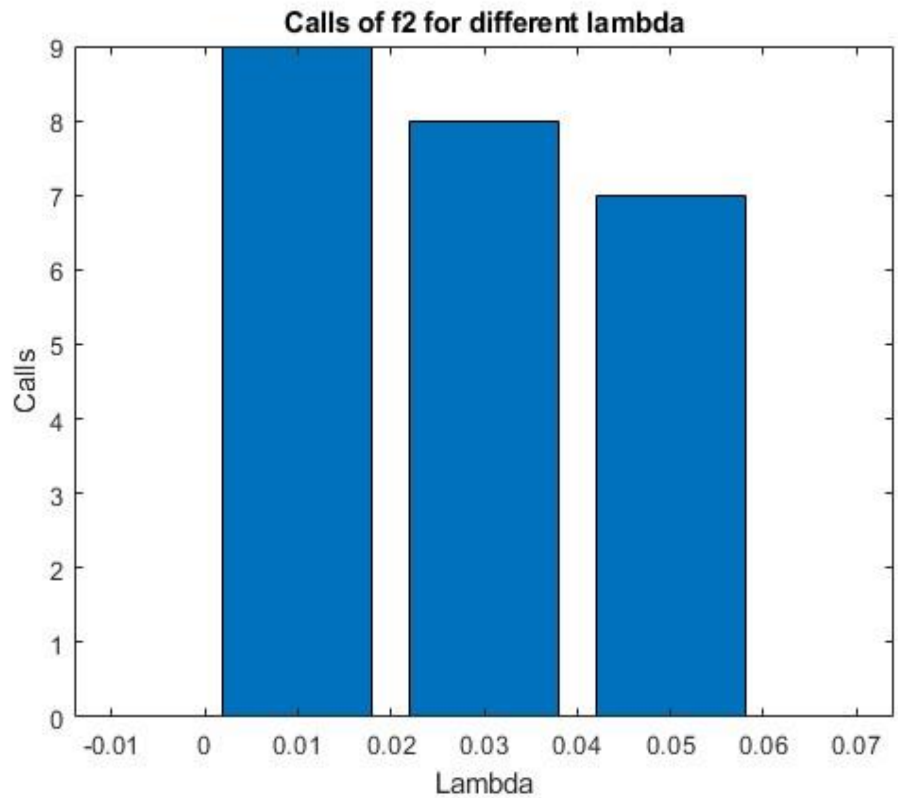
Σχόλια: αρκετά αργός αλγόριθμος, αλλά με λίγες κλήσεις και επαναλήψεις. Επίσης, επειδή η Fibonacci στην Matlab ξεκινάει από το 1 και όχι από το 0, αφού βρω το n το αφαιρώ κατά 1 και όπου καλείται μέσα στον αλγόριθμο η Fibonacci προσθέτω +1 για να λειτουργούν όλα όπως πρέπει. Ιδιαίτερο είναι ότι για διαφορετική συνάρτηση με κοινό λάμδα μπορεί να έχει λιγότερες ή περισσότερες κλήσεις, πράγμα που δεν φαινόταν στους άλλους αλγορίθμους.

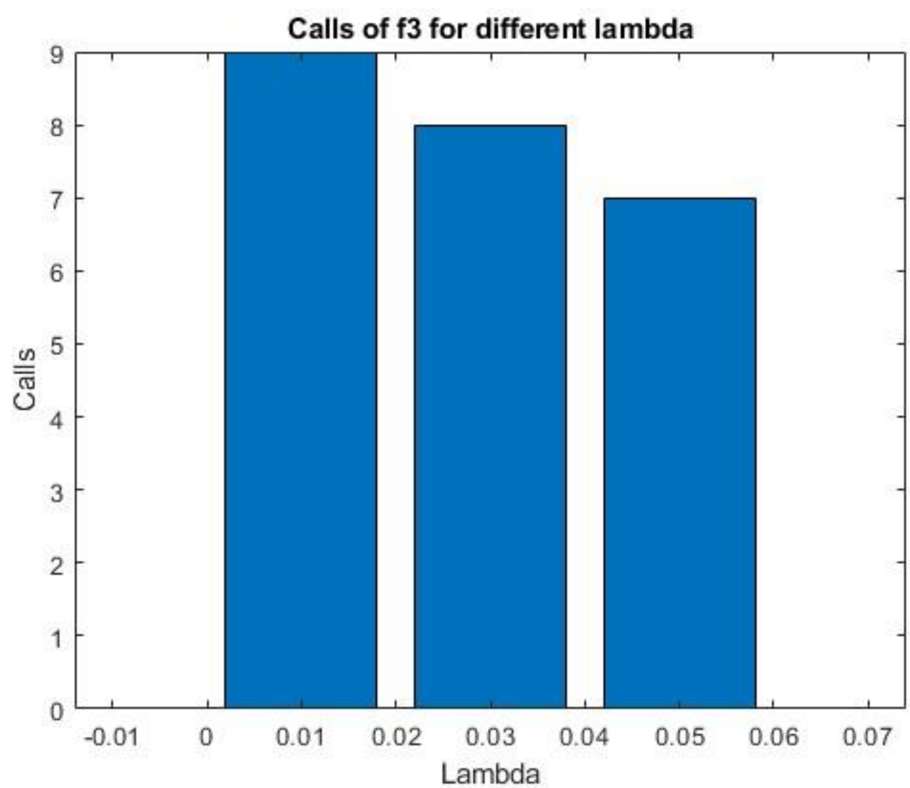
Θέμα 4: Μέθοδος Διχοτόμου με χρήση παραγώγου

Αρχεία: dixotomouDerivative.m, thema_4.m

Οι υπολογισμοί της αντικειμενικής συνάρτησης για διάφορα λάμδα:

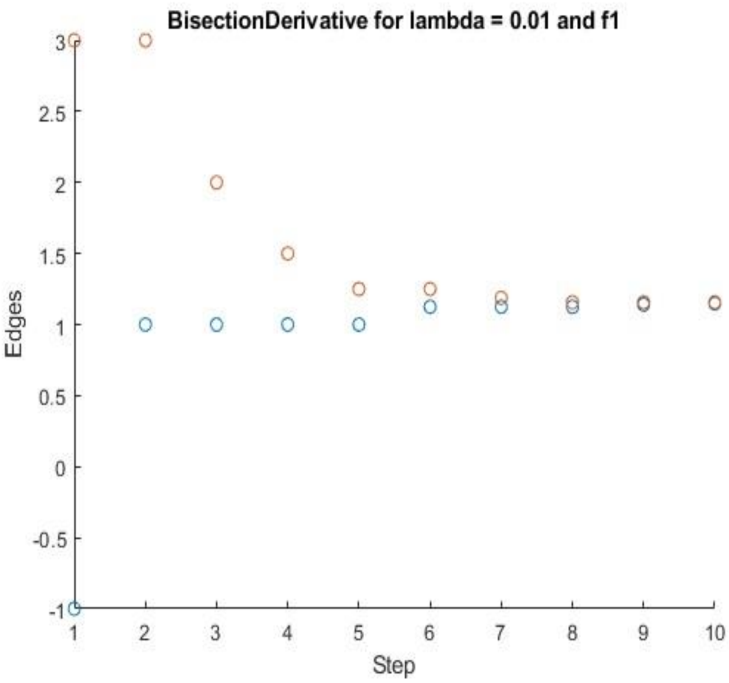


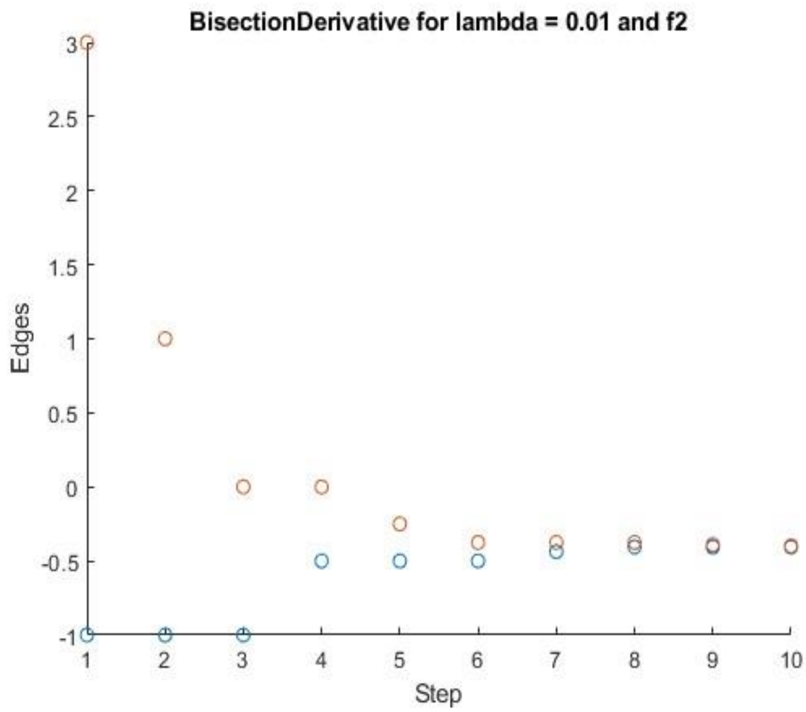

Calls of f2 for different lambda
Calls
Lambda
0
1
2
3
4
5
6
7
8
9
-0.01
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07

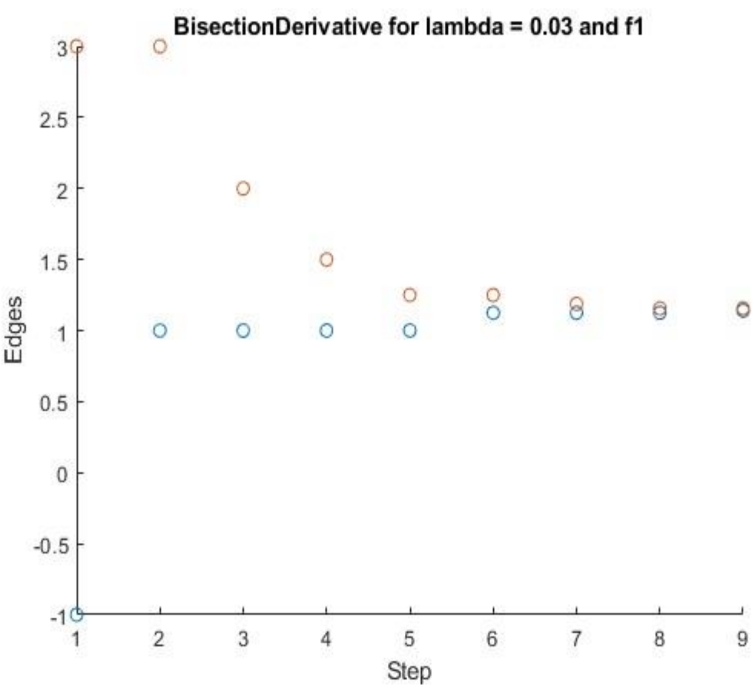

Calls of f3 for different lambda
Calls
Lambda
0
1
2
3
4
5
6
7
8
9
-0.01
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07

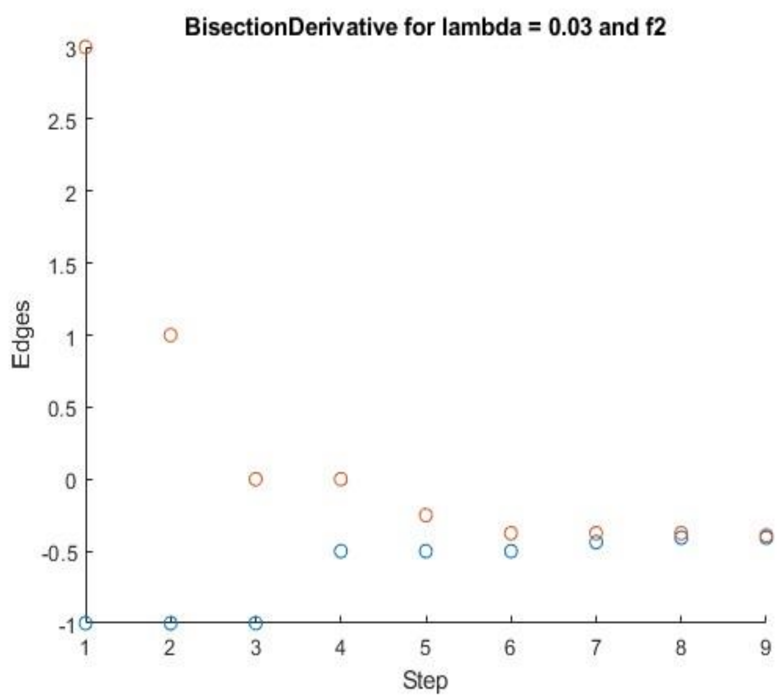
Ξανά, για αυξανόμενες τιμές του λάμδα, η αντικειμενική καλείται λιγότερες φορές.

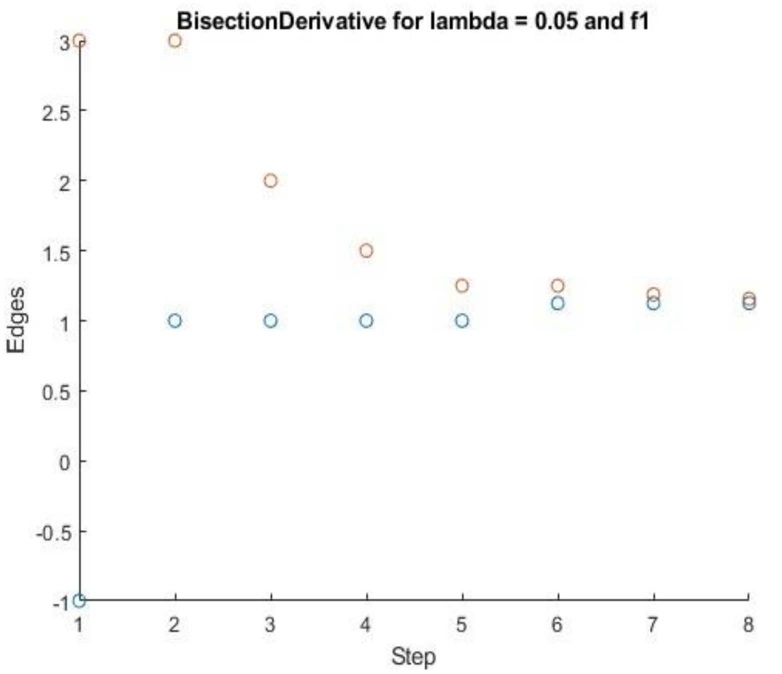
Συνεχίζοντας για τα άκρα του διαστήματος:

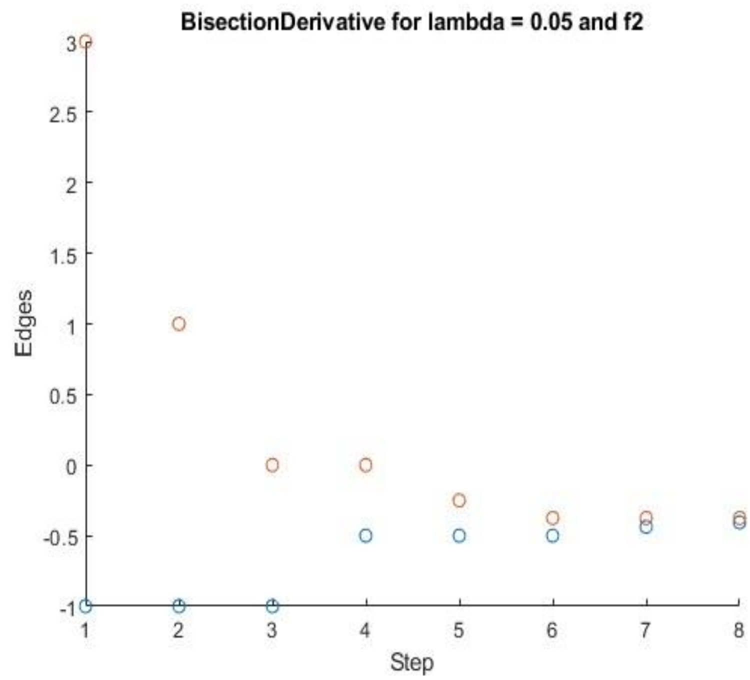


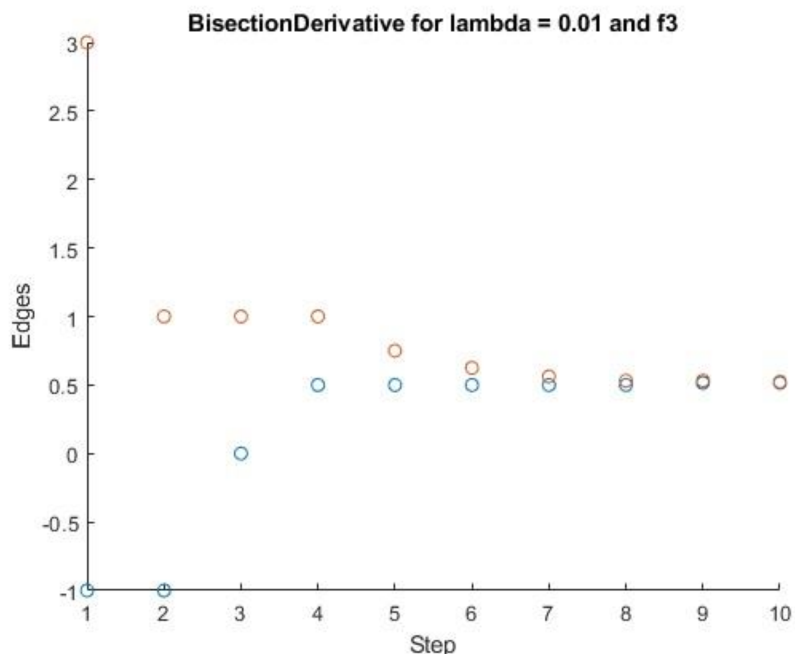


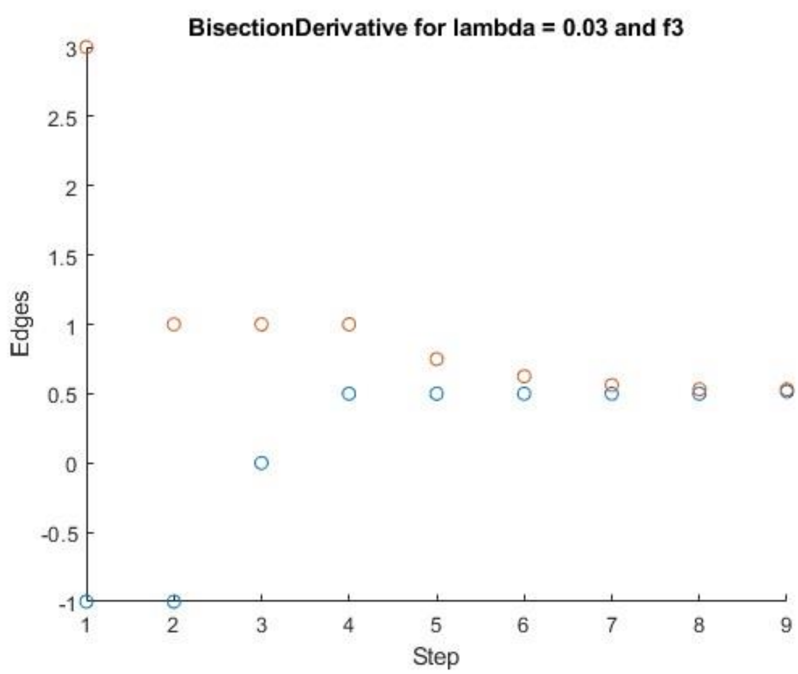


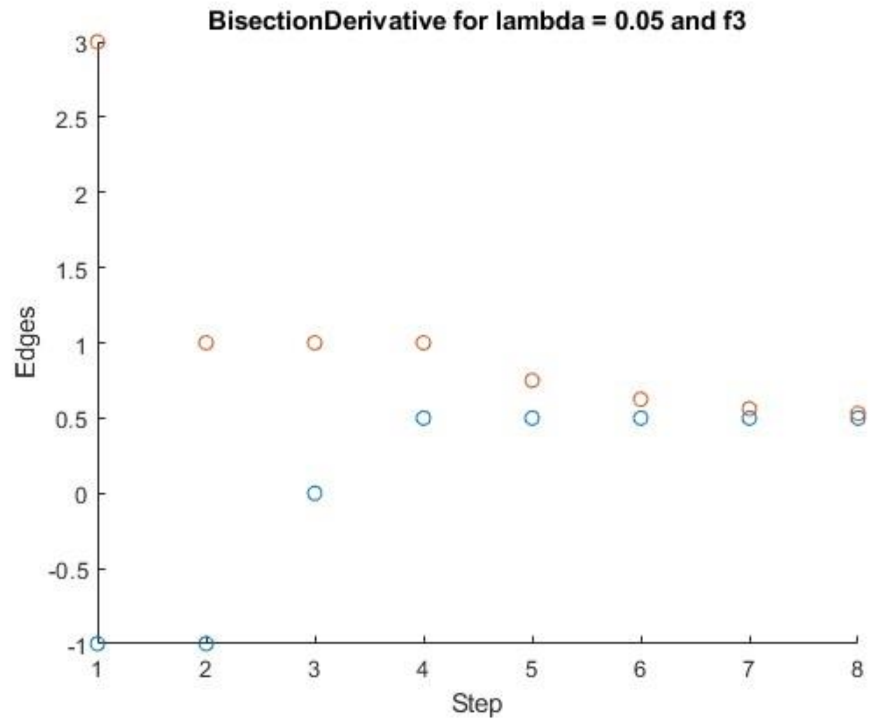



BisectionDerivative for lambda = 0.05 and f1
Edges
Step


BisectionDerivative for lambda = 0.05 and f2
Edges
Step


BisectionDerivative for lambda = 0.01 and f3
Edges
Step


BisectionDerivative for lambda = 0.03 and f3
Edges
Step



Όπως και πριν, για μεγαλύτερα λάμδα, το διάστημα που βρίσκεται το ελάχιστο είναι μεγαλύτερο, αλλά κερδίζουμε μερικές επαναλήψεις στον αλγόριθμο. Οπότε, χάνουμε σε ακρίβεια για να τρέξει λίγο γρηγορότερα ο αλγόριθμος.

Σχόλια: γρήγορος αλγόριθμος με πολύ λίγες κλήσεις και επαναλήψεις. Για την παράγωγο χρησιμοποιήθηκε η Symbolic Math Toolbox του Matlab (χρησιμοποιήθηκε και από τους άλλου αλγόριθμους για να γραφτούν οι συναρτήσεις).

<u>Συμπεράσματα</u>

Βλέπουμε και στους 4 αλγορίθμους πως βγάζουν διαστήματα που περιέχουν τα ελάχιστα των τριων συναρτήσεων, τα οποία είναι για τη πρώτη χ = 1.14991, δεύτερη χ = -0.401405 και τρίτη χ =0.520086. Από άποψη χρόνου, όλες πέρα από τη Fibonacci ήταν στις ίδιες ταχύτητες. Η Μέθοδος Διχοτόμου με χρήση παραγώγων είχε τις λιγότερες κλήσεις τις αντικειμενικής και τις λιγότερες επαναλήψεις μαζί με την Μέθοδο Διχοτόμου χωρίς χρήση παραγώγων.