

Ελαχιστοποίηση κυρτής συνάρτησης μιας μεταβλητής σε δοσμένο διάστημα

Θεολόγης Γεώργιος AEM:10413
email: gtheolog@ece.auth.gr

Νοέμβριος 2023

1 Εισαγωγή

Το αντικείμενο της παρούσας εργασίας είναι η υλοποίηση γνωστών αλγορίθμων ελαχιστοποίησης κυρτών συναρτήσεων μιας μεταβλητής $f(x)$ όταν το x ανήκει σε διάστημα $[a, b]$. Επιπλέον, πραγματοποιείται και πειραματισμός με διάφορες παραμέτρους εισόδου αυτών των αλγορίθμων για να παρατηρηθεί η εξάρτηση της αποδοτικότητας, της σύγκλισης και άλλων χαρακτηριστικών των μεθόδων από αυτές. Οι αλγόριθμοι που θα υλοποιηθούν είναι οι εξής:

1. Μέθοδος της Διχοτόμου.
2. Μέθοδος του Χρυσού Τομέα.
3. Μέθοδος Fibonacci.
4. Μέθοδος της Διχοτόμου με χρήση παραγώγου.

Οι συγκεκριμένοι εφαρμόζονται και σε μη κυρτές συναρτήσεις, αν περιορίσουμε το αρχικό εύρος αναζήτησης $[a, b]$ στο οποίο η $f(x)$ είναι τουλάχιστον σχεδόν κυρτή. Η μέθοδος διχοτόμου με χρήση παραγώγου απαιτεί όμως και την διαφορισιμότητα της $f(x)$ και εφαρμόζεται τουλάχιστον σε ψευδόκυρτες συναρτήσεις. Οι συναρτήσεις που καλούμαστε να ελαχιστοποιήσουμε είναι οι εξής:

- $f_1(x) = (x - 1)^3 + (x - 4)^2 \cos(x)$
- $f_2(x) = e^{-2x} + (x - 2)^2$
- $f_3(x) = x^2 \ln(0.5x) + \sin(0.2x)^2$

Από τα δεδομένα της εργασίας θα υποθέσουμε ότι οι συναρτήσεις αυτές τηρούν τις προϋποθέσεις για την εφαρμογή των μεθόδων. Πράγματι, στο αρχικό εύρος αναζήτησης $[0, 3]$ η $f_2(x)$ είναι κυρτή ενώ οι άλλες δύο είναι κυρτές σε υποδιάστημα της περιοχής $[0, 3]$ το οποίο περιλαμβάνει το ελάχιστο. Αυτό μπορούμε να το συμπεράνουμε από ποιοτικές γραφικές παραστάσεις των συναρτήσεων με δείγματα κάποιες τιμές του x .

Οι κάθε μέθοδος έχει υλοποιηθεί σαν συνάρτηση στο δικό της αρχείο `.m`, ενώ το βασικό script είναι το `optimization_project_1.m` και χρησιμοποιείται βοηθητικά το `Varying_tolerance_plots.m` για άντληση plots κοινής μορφής μεταξύ των θεμάτων.

2 Θέμα 1

2.1 Αλγόριθμος μεθόδου Διχοτόμου

Ακολουθεί ο κώδικας της υλοποίησης της μεθόδου της Διχοτόμου στο Matlab. Πρόκειται για την συνάρτηση Dichotomos με ορίσματα το αρχικό εύρος αναζήτησης [ap,bp] , την συνάρτηση fun που θα ελαχιστοποιηθεί, το τελικό εύρος αναζήτησης l και τέλος την e απόσταση από την διχοτόμο. Ός έξοδο έχουμε τους πίνακες A και B που έχουν τα άκρα αναζήτησης ak και bk αντίστοιχα, συναρτήσει του βήματος k καθώς και τον αριθμό κλήσεων (calls) της αντικειμενικής συνάρτησης κατά την εκτέλεση της μεθόδου. Η διαδικασία είναι αυτή που δίνεται για την μέθοδο από την επιλεγμένη βιβλιογραφία. Χρησιμοποιούμε επίσης τους πίνακες FX1 και FX2 για να αποθηκεύουμε τις τιμές f(x1k) και f(x2k) της αντικειμενικής συνάρτησης αν και δεν είναι απαραίτητο.

```
1 function [A,B,calls] = Dichotomos(ap,bp,fun,l,e)
2 fun=matlabFunction(fun);
3 A=ap;
4 B=bp;
5 FX1=[];
6 FX2=[];
7 k=1;
8 calls=0;
9 while (B(end)-A(end)>l)
10     x1k=(A(end)+B(end))/2-e;
11     x2k=(A(end)+B(end))/2+e;
12     fx1=fun(x1k);
13     fx2=fun(x2k);
14     calls=calls+2;
15     if (fx1<fx2)
16         an=A(end);
17         bn=x2k;
18     else
19         an=x1k;
20         bn=B(end);
21     end
22     k=k+1;
23     A=[A,an];
24     B=[B,bn];
25     FX1=[FX1,fx1];
26     FX2=[FX2,fx2];
27
28 end
29 end
```

2.2 Κλήσεις αντικειμενικής συνάρτησης σε συνάρτηση απόστασης από την διχοτόμο

Σε αυτό το σημείο θα εξετάσουμε την επιρροή που έχει η παράμετρος e στον αριθμό των κλήσεων της αντικειμενικής συνάρτησης όταν έχουμε δεδομένο τελικό διάστημα αναζήτησης l=0.01. Προφανώς πρέπει να ισχύει $\epsilon < \frac{l}{2}$ ώστε το διάστημα αναζήτησης να μικραίνει και συνακολούθως η μέθοδος να συγκλίνει σε αποτέλεσμα και να τερματίζει. Επομένως υλοποιούμε την μέθοδο σε κάθε συνάρτηση για τιμές του e στο [0.001,0.0049]. Ακολουθούν οι γραφικές παραστάσεις:

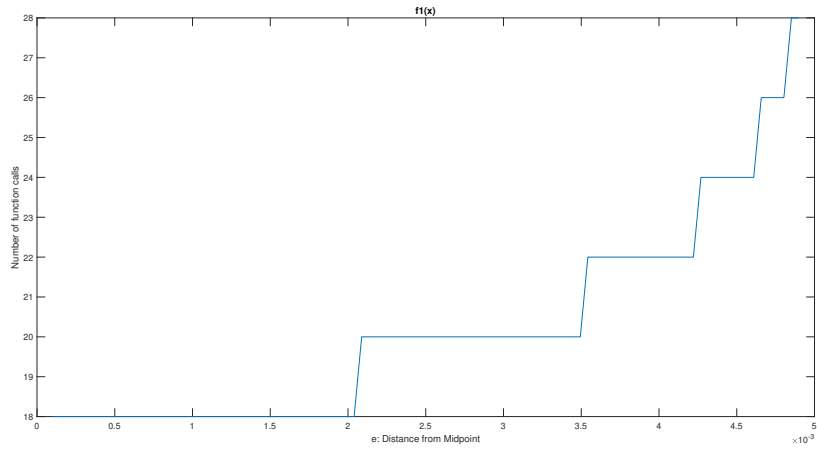


Figure 1: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του ϵ για $l=0.01$ στη μέθοδο της διχοτόμου.

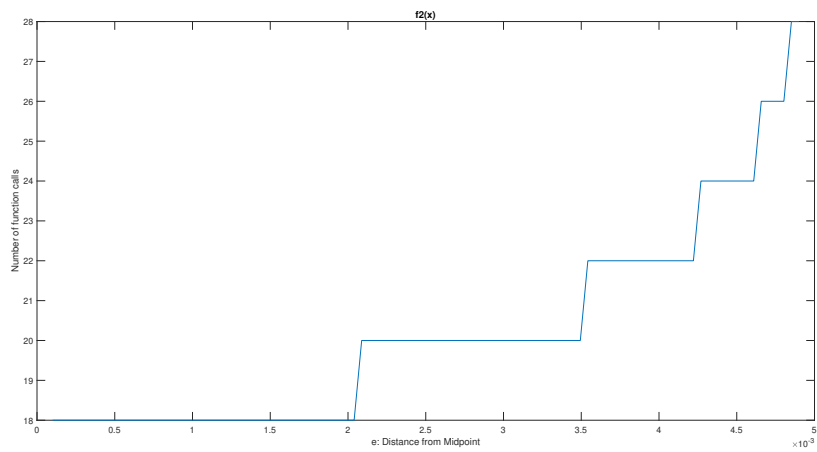


Figure 2: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του ϵ για $l=0.01$ στη μέθοδο της διχοτόμου.

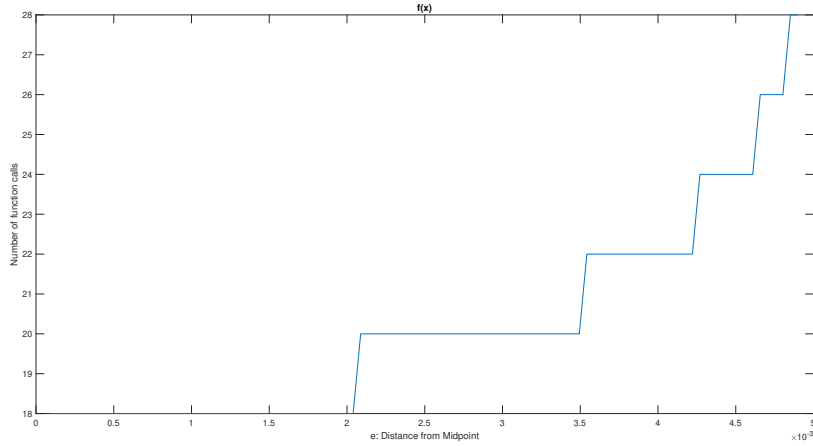


Figure 3: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του ϵ για $l=0.01$ στη μέθοδο της διχοτόμου.

Καταρχάς παρατηρούμε ότι και για τις 3 συναρτήσεις τα διαγράμματα είναι ταυτόσημα. Αυτό δεν αποτελεί έκπληξη αφού και για τις τρεις συναρτήσεις το σημείου ελαχίστου είναι στο διάστημα $[1,2.1]$ και πιο σημαντικά ότι το μήκος κάθε διαδοχικό διάστημα αναζήτησης συνδέεται με το προηγούμενο με την εξής σχέση: $L_{n+1} = \frac{L_n}{2} + \epsilon$. Συνεπώς, ανεξαρτήτως κυρτής συνάρτησης για ίδιο l, ϵ και αρχικό εύρος αναζήτησης, τα μήκη των διαδοχικών διαστημάτων είναι ίδια στη μέθοδο της Διχοτόμου και επομένως είναι ίδιος ο αριθμός κλήσεων μέχρι το τελικό διάστημα αναζήτησης να γίνει μικρότερο από 1 και συνεπώς να τερματίσει ο αλγόριθμος.

Διερευνώντας την συμπεριφορά του αριθμού των κλήσεων εξάγουμε κάποια σημαντικά συμπεράσματα. Για σχετικά μικρό ϵ σε σχέση με το 1 (π.χ. $\epsilon = 0.1 \cdot l$) καταλαβαίνουμε ότι τα διαδοχικά διαστήματα αναζήτησης είναι μειωμένα κατά μισό από το προηγούμενο τους αφού:

$$L_{n+1} = \frac{L_n}{2} + \epsilon \approx \frac{L_n}{2}$$

Οπότε, κάθε επανάληψη το διάστημα αναζήτησης γίνεται συνέχεια το μισό και έχουμε 2 κλήσεις της αντικειμενικής συνάρτησης. Άρα για το τελικό διάστημα αναζήτησης, αν n είναι ο αριθμός των κλήσεων της αντικειμενικής συνάρτησης και $[a1, b1]$ το αρχικό διάστημα αναζήτησης θα έχουμε:

$$L \approx \left(\frac{1}{2}\right)^{\frac{n}{2}} \cdot (b1 - a1)$$

Και γιατί πρέπει να ισχύει $L < 1$ έχουμε:

$$\left(\frac{1}{2}\right)^{\frac{n}{2}} < \frac{l}{b1 - a1}$$

Πράγματι αυτή η σχέση επαληθεύεται για μικρά ϵ σε σχέση με το 1 αφού για $a1=0, b1=3$ και $l=0.01$ έχουμε:

$$\left(\frac{1}{2}\right)^{\frac{n}{2}} < \frac{l}{(b1 - a1)} \Rightarrow \left(\frac{1}{2}\right)^{\frac{n}{2}} < \frac{0.01}{3}$$

$$\frac{n}{2} > 8.22 \Rightarrow n > 16.44$$

Άρα ο ελάχιστος άρτιος ακέραιος n που ικανοποιεί την ανίσωση:

$$n = 18$$

Αυτός είναι και ο αριθμός των κλήσεων της αντικειμενικής συνάρτησης που παρατηρούμε στα διαγράμματα για μικρό ϵ . Όσο μεγαλώνουμε όμως την τιμή του ϵ , τα διαδοχικά διαστήματα αναζήτησης είναι αύξουσα συνάρτηση του ϵ καθώς:

$$L_{n+1} = \frac{L_n}{2} + \epsilon$$

Συνεπώς, όσο το ϵ μεγαλώνει, τόσο λιγότερο μικραίνει κατά ποσοστό το διάστημα αναζήτησης (άρα για δεδομένη επανάληψη το εύρος αναζήτησης είναι μεγαλύτερο από ότι θα ήταν για μικρότερο ϵ) και επομένως χρειαζόμαστε περισσότερες επαναλήψεις άρα και κλήσεις της αντικειμενικής συνάρτησης για να τερματίσει ο αλγόριθμος. Αυτό είναι κάτι που πράγματι παρατηρούμε και στις παραπάνω γραφικές παραστάσεις. Είναι προφανές ότι το ϵ δεν μπορεί να ξεπεράσει το $\frac{1}{2}$ γιατί τότε το διάστημα αναζήτησης δεν θα μπορέσει να γίνει ποτέ μικρότερο του 1 και συνακολούτως ο αλγόριθμος δεν θα μπορεί να τερματίσει.

2.3 Κλήσεις αντικειμενικής συνάρτησης σε συνάρτηση με τελικό εύρος αναζήτησης l

Συνεχίζοντας θα εξετάσουμε την επιρροή που έχει το τελικός εύρος αναζήτησης στον αριθμό των κλήσεων της αντικειμενικής συνάρτησης όταν έχουμε δεδομένη τιμή παραμέτρου ϵ και αρχικό διάστημα αναζήτησης $[0,3]$. Προφανώς πρέπει να ισχύει $2 \cdot \epsilon < l \Rightarrow 0.002 < l$ ώστε το διάστημα αναζήτησης να μικραίνει και συνακολούτως η μέθοδος να συγκλίνει σε αποτέλεσμα και να τερματίζει. Επομένως υλοποιούμε την μέθοδο σε κάθε συνάρτηση για τιμές του l στο $[0.0025, 0.1]$.

Ακολουθούν οι γραφικές παραστάσεις:

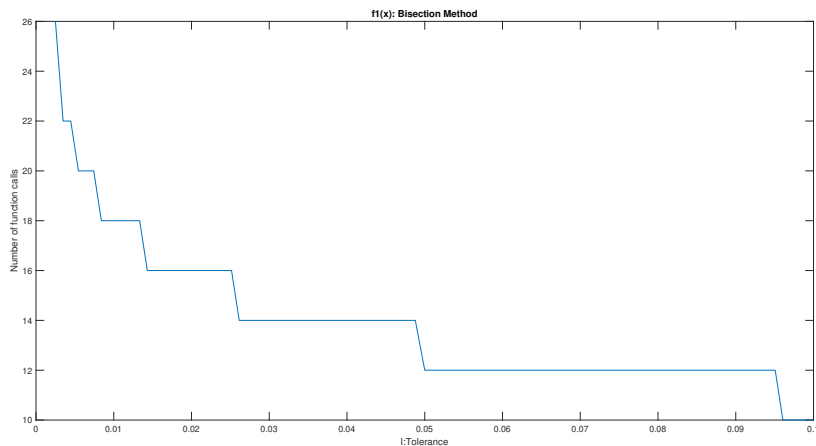


Figure 4: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συνάρτησει του l για $\epsilon=0.001$ στη μέθοδο της διχοτόμου.

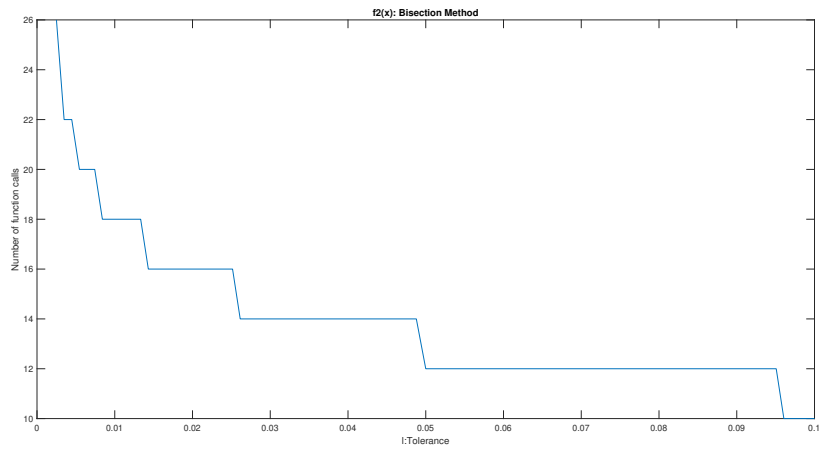


Figure 5: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συνάρτησει του l για $\varepsilon=0.001$ στη μέθοδο της διχοτόμου.

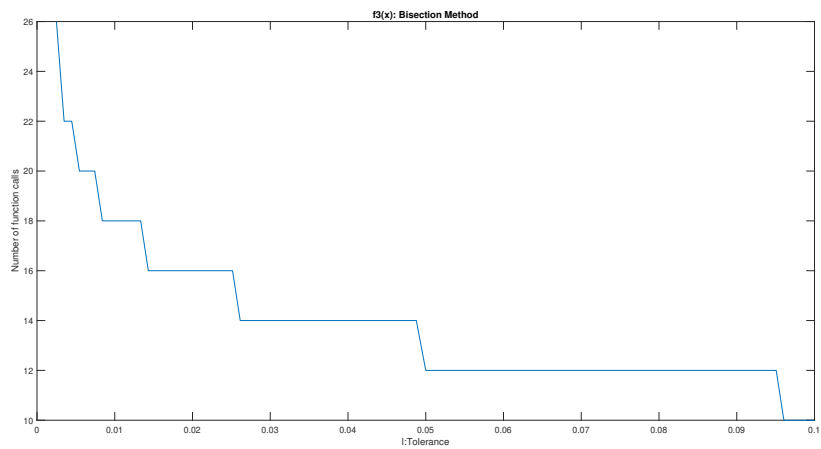


Figure 6: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συνάρτησει του l για $\varepsilon=0.001$ στη μέθοδο της διχοτόμου.

Από τις γραφικές παραστάσεις παρατηρούμε ότι η συμπεριφορά του αριθμού των κλήσεων της αντικειμενικής συνάρτησης σε σχέση με το l μοιάζει με λογαριθμική και είναι φθίνουσα. Όσο αυξάνουμε το εύρος τελικής αναζήτησης l τόσο μικραίνει ο αριθμός κλήσεων της αντικειμενικής συνάρτησης. Αυτό είναι απολύτως λογικό καθώς, για μεγαλύτερο l ο αλγόριθμος κάνει λιγότερες επαναλήψεις για να ισχύει $b_k - a_k < l$ και συνεπώς έχουμε λιγότερες κλήσεις της $f_i(x)$ για να τερματίσει ο αλγόριθμος. Μάλιστα για $l > 0.007$ η σχέση $(\frac{1}{2})^{\frac{n}{2}} < \frac{l}{b_1 - a_1}$ για n αριθμό κλήσεων επαληθεύεται από τα διαγράμματα. Όπως αναφέρθηκε προηγουμένως οι γραφικές παραστάσεις για τις τρεις συναρτήσεις είναι ταυτόσημες καθώς ο αριθμός των κλήσεων της συνάρτησης εξαρτάται μόνο από το μήκος των διαδοχικών διαστημάτων αναζήτησης και την συνθήκη τερματισμού (άρα από τα a_1, b_1, l, ϵ) και όχι από την εκάστοτε κυρτή συνάρτηση.

2.4 Διαγράμματα διαστημάτων αναζήτησης $[a_k, b_k]$ συναρτήσε του δείκτη βήματος

Συνεχίζοντας την συλλογιστική της προηγούμενης ενότητας θα αναπαραστήσουμε γραφικά το διάστημα αναζήτησης $[a_k, b_k]$ για δείκτη βήματος k , για διάφορες τιμές του l . Για αισθητικό λόγο τα διαγράμματα αυτής της μορφής αν και αποτελούνται από διάκριτα σημεία (κύκλο για a_k , και x για b_k), συμπληρώνονται με συνεχή γραμμή αν και δεν υπάρχουν οι τιμές πάνω σε αυτή.

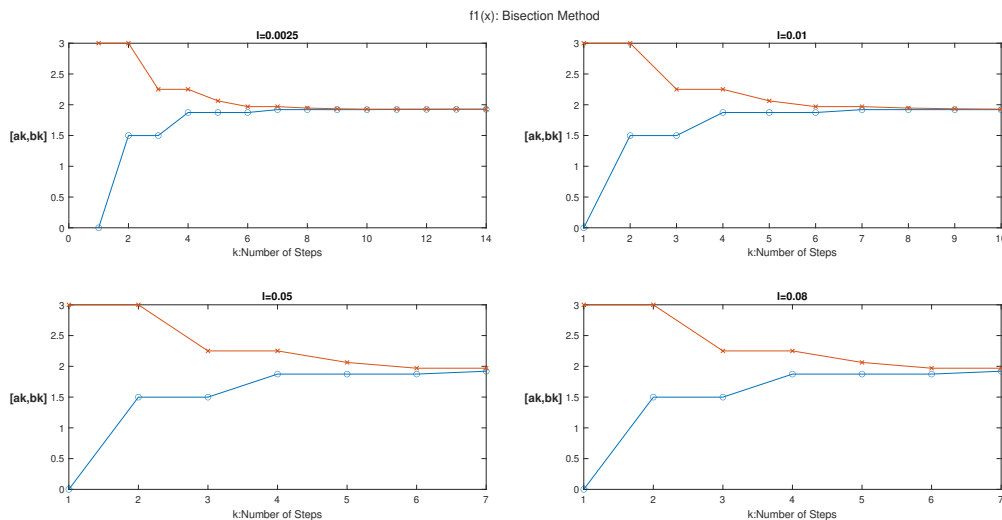


Figure 7: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσε του δείκτη βήματος k για $\epsilon=0.001$ στη μέθοδο της διχοτόμου.

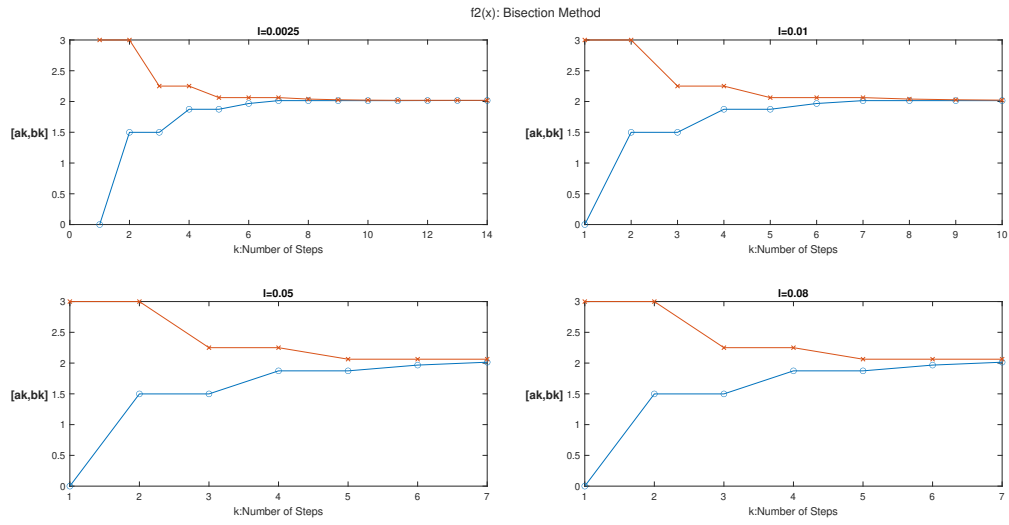


Figure 8: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συνάρτησει του δείκτη βήματος k για $\epsilon=0.001$ στη μέθοδο της διχοτόμου.

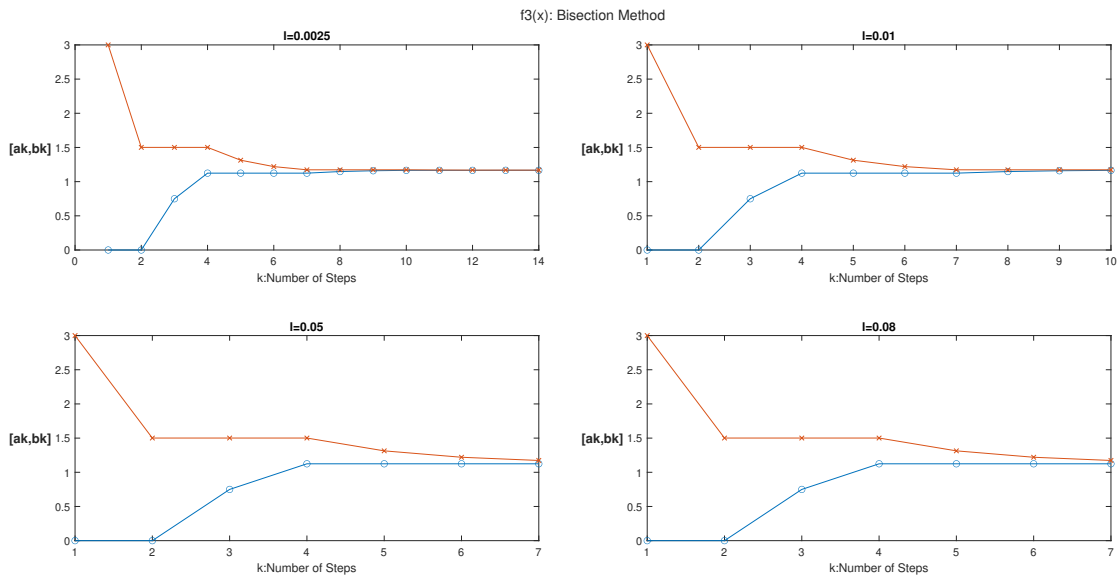


Figure 9: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συνάρτησει του δείκτη βήματος k για $\epsilon=0.001$ στη μέθοδο της διχοτόμου.

Παρατηρούμε σαφώς από τα διαγράμματα ότι όσο μικρότερο είναι το l , τόσα περισσότερα βήματα απαιτούνται για τον τερματισμό του αλγορίθμου αλλά και τόσο πιο κοντά είναι μεταξύ τους τα τελικά άκρα του διαστήματος $[a_k, b_k]$ όταν η μέθοδος ολοκληρώνεται. Άρα έχουμε καλύτερη προσέγγιση για το σημείου ελαχίστου για το μικρότερο l . Λόγω της φύσης του αλγορίθμου, για τις διάφορες τιμές του l , το διάγραμμα $[k, [a_k, b_k]]$ είναι ταυτόσημο ως προς κάθε κοινό βήμα k για μια συνάρτηση ενώ όταν απλά έχουμε μεγαλύτερο l τερματίζεται ο αλγόριθμος πιο γρήγορα.

3 Θέμα 2

3.1 Αλγόριθμος μεθόδου Χρυσής Τομής

Ακολουθεί ο κώδικας της υλοποίησης της μεθόδου της Χρυσής Τομής στο Matlab. Πρόκειται για την συνάρτηση GoldenRatio με ορίσματα το αρχικό εύρος αναζήτησης [ap,bp] , την συνάρτηση fun που θα ελαχιστοποιήθει, το τελικό εύρος αναζήτησης l και τέλος μια κενή παράμετρο e που δεν αξιοποιείται στη συνάρτηση αλλά έχει προστεθεί για λόγους συμβατότητας του κώδικα. Ός έξοδο έχουμε τους πίνακες A και B που έχουν τα άκρα αναζήτησης ak και bk αντίστοιχα, συναρτήσει του βήματος k καθώς και τον αριθμό κλήσεων (calls) της αντικειμενικής συνάρτησης κατά την εκτέλεση της μεθόδου. Η διαδικασία είναι αυτή που δίνεται για την μέθοδο απο την επιλεγμένη βιβλιογραφία. Χρησιμοποιούμε επίσης τους πίνακες FX1 και FX2 για να αποθηκεύουμε τις τιμές f(x1k) και f(x2k) της αντικειμενικής συνάρτησης, και τους πίνακες X1,X2 για να αποθηκεύουμε τα x1k,x2k αντίστοιχα που υπολογίζονται σε κάθε βήμα k.

Ακολουθεί ο κώδικας:

```
1 function [A,B,calls] = GoldenRatio(ap,bp,fun,l,e)
2 fun=matlabFunction(fun);
3 g=(sqrt(5)-1)/2;
4 x11=ap+(1-g)*(bp-ap);
5 x21=ap+g*(bp-ap);
6 calls=0;
7 fX11=fun(x11);
8 fX21=fun(x21);
9 calls=calls+2;
10 k=1;
11 A=ap;
12 B=bp;
13 FX1=fX11;
14 FX2=fX21;
15 X1=x11;
16 X2=x21;
17 while (B(k)-A(k)>=1)
18     if (FX1(k)<FX2(k))
19         an=A(k);
20         bn=X2(k);
21         x1n=an+(1-g)*(bn-an);
22         x2n=X1(k);
23         fx1n=fun(x1n);
24         fx2n=FX1(k);
25         calls=calls+1;
26     else
27         an=X1(k);
28         bn=B(k);
29         x1n=X2(k);
30         x2n=an+g*(bn-an);
31         fx1n=FX2(k);
32         fx2n=fun(x2n);
33         calls=calls+1;
34     end
35     k=k+1;
36     A=[A,an];
37     B=[B,bn];
38     FX1=[FX1,fx1n];
39     FX2=[FX2,fx2n];
40     X1=[X1,x1n];
```

```

41     X2=[X2 , x2n] ;
42 end
43 end

```

3.2 Κλήσεις αντικειμενικής συνάρτησης σε συνάρτηση με τελικό εύρος αναζήτησης 1

Συνεχίζοντας θα εξετάσουμε την επιρροή που έχει το τελικό εύρος αναζήτησης στον αριθμό των κλήσεων της αντικειμενικής συνάρτησης όταν έχουμε αρχικό διάστημα αναζήτησης $[0,3]$. Επομένως υλοποιούμε την μέθοδο σε κάθε συνάρτηση για τιμές του 1 στο $[0.0025,0.1]$.

Ακολουθούν οι γραφικές παραστάσεις:

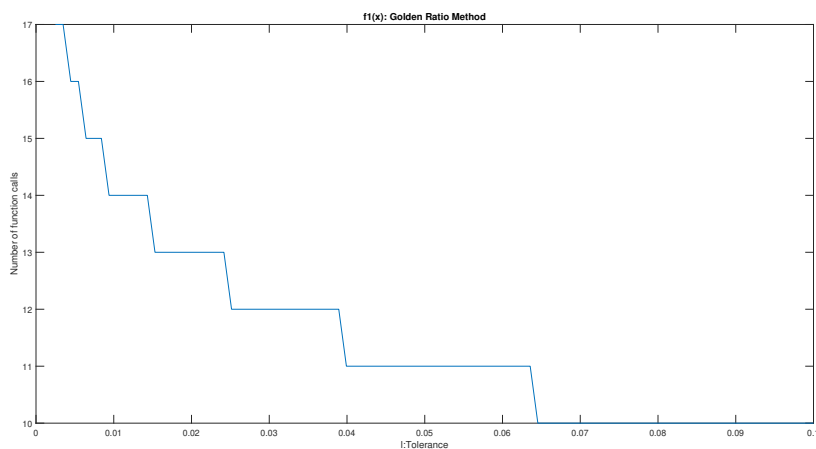


Figure 10: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του 1 στη μέθοδο της χρυσής τομής.

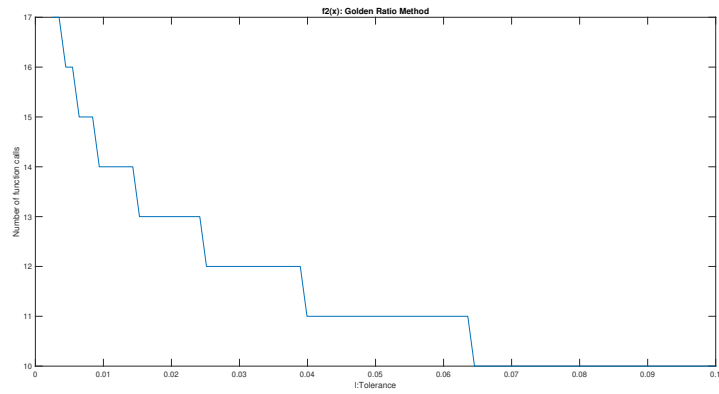


Figure 11: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του 1 στη μέθοδο της χρυσής τομής.

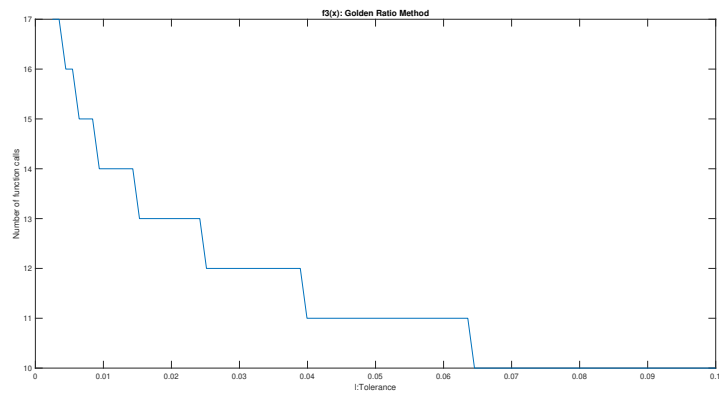


Figure 12: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του 1 στη μέθοδο της χρυσής τομής.

Από τις γραφικές παραστάσεις των τριών συναρτήσεων που είναι πάλι ταυτόσημες, παρατηρούμε ότι η συμπεριφορά του αριθμού των κλήσεων της αντικειμενικής συνάρτησης σε σχέση με το l μοιάζει με λογαριθμική ($-\log x$) και είναι φθίνουσα, όπως ήταν και της διχοτόμου. Όσο αυξάνουμε το εύρος τελικής αναζήτησης l τόσο μικραίνει ο αριθμός κλήσεων της αντικειμενικής συνάρτησης. Αυτό είναι λογικό καθώς σε κάθε βήμα το διάστημα αναζήτησης μειώνεται κατά:

$$b_{k+1} - a_{k+1} = \gamma(b_k - a_k) (\gamma = 0.618)$$

. Άρα άμα έχουν γίνει m επαναλήψεις μέχρι τον τερματισμό το τελικό εύρος αναζήτησης είναι:

$$L_m < \gamma^m(b_1 - a_1)$$

Όμως αφού πρέπει να ισχύει $L_m \leq 1$ και γνωρίζουμε ότι σε κάθε επανάληψη πριν τον τερματισμό γίνεται 1 κλήση της $f_i(x)$ ενώ στην αρχικοποίηση έχουμε δύο κλήσεις της, τότε αν n είναι ο αριθμός των συνολικών κλήσεων τότε πρέπει να ισχύει ότι:

$$(0.618)^{n-2} < \frac{l}{b_1 - a_1}$$

Πράγματι παρατηρούμε ότι αυτή η σχέση επιβεβαιώνεται από τα γραφήματα. Επίσης καθίσταται προφανές ότι το πλήθος των βημάτων άρα και κλήσεων της αντικειμενικής συνάρτησης εξαρτάται μόνο από το αρχικό διάστημα $[a_1, b_1]$ και το τελικό εύρος αναζήτησης l , και όχι από την εκάστοτε κυρτή συνάρτηση, κάτι που ερμηνεύει γιατί τα τρία διαγράμματα ταυτίζονται. Παρατηρούμε ότι για δεδομένο αρχικό εύρος αναζήτησης $[a_1, b_1] = [0, 3]$ και τελικό εύρος αναζήτησης l , η μέθοδος της χρυσής τομής απαιτεί λιγότερες κλήσεις της αντικειμενικής συνάρτησης από ότι απαιτεί η μέθοδος του διχοτόμου. Για παράδειγμα, για $l = 0.01$ η μέθοδος διχοτόμου απαιτεί 18 κλήσεις ενώ της χρυσής τομής μόλις 14. Σαφώς αυτό σημαίνει ότι η μέθοδος της χρυσής τομής είναι πιο αποδοτική από αυτή της διχοτόμου.

3.3 Διαγράμματα διαστημάτων αναζήτησης $[a_k, b_k]$ συναρτήσει του δείκτη βήματος

Στη συνέχεια θα παραστήσουμε γραφικά το διάστημα αναζήτησης $[a_k, b_k]$ για δείκτη βήματος k στην εκτέλεση της μεθόδου, για διάφορες τιμές του l έχοντας το ίδιο αρχικό εύρος αναζήτησης $[0, 3]$.

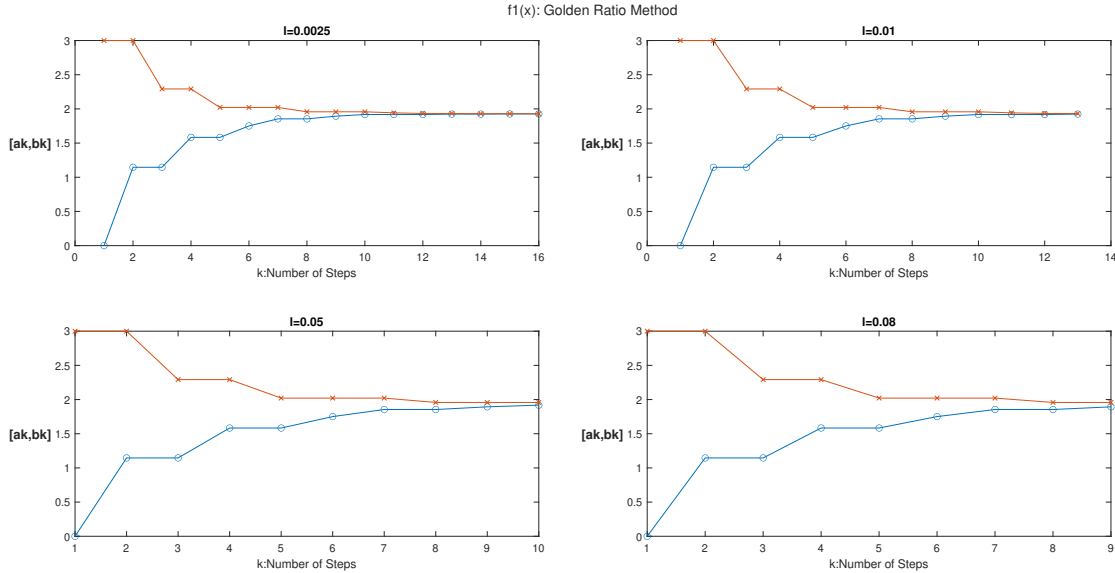


Figure 13: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσει του δείκτη βήματος k στη μέθοδο της χρυσής τομής.

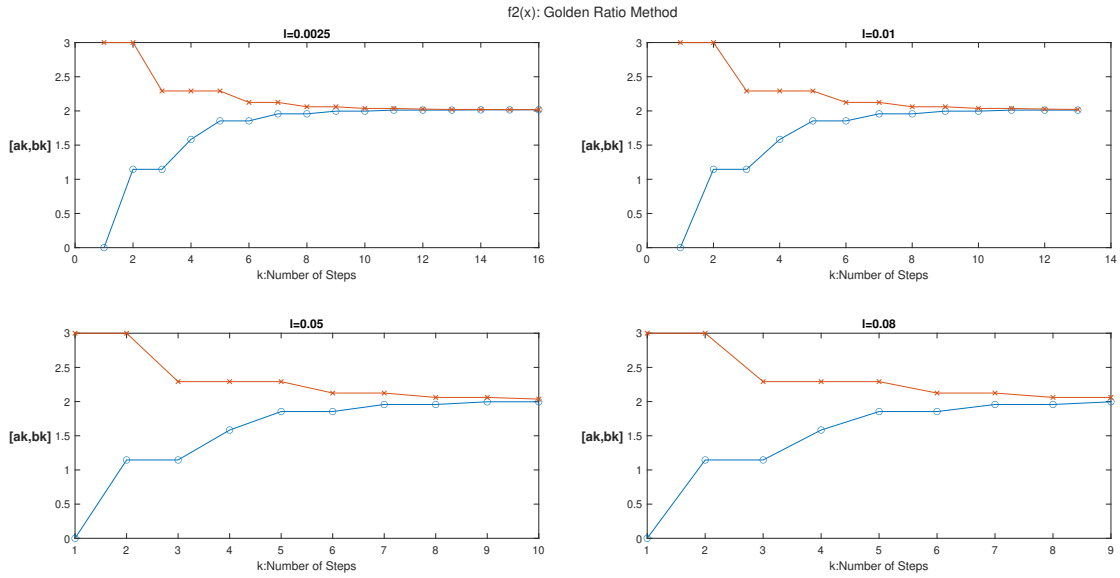


Figure 14: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συνάρτησει του δείκτη βήματος k στη μέθοδο της χρυσής τομής.

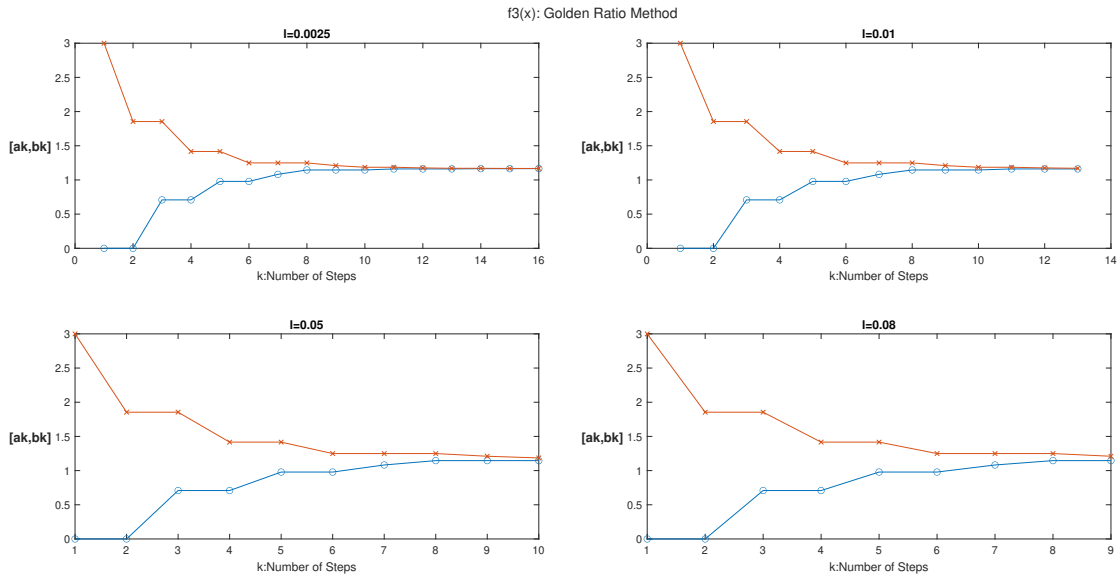


Figure 15: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συνάρτησει του δείκτη βήματος k στη μέθοδο της χρυσής τομής.

Παρατηρούμε όπως και στη προηγούμενη μέθοδο από τα διαγράμματα ότι όσο μικρότερο είναι το l , τόσα περισσότερα βήματα απαιτούνται για τον τερματισμό του αλγορίθμου άλλα και τόσο πιο κοντά είναι μεταξύ τους τα τελικά άκρα του διαστήματος $[a_k, b_k]$ όταν η μέθοδος ολοκληρώνεται. Άρα έχουμε καλύτερη προσέγγιση για το σημείου ελαχίστου για το μικρότερο l . Λόγω της φύσης του αλγορίθμου, για τις διάφορες τιμές του l , το διάγραμμα $[k, [a_k, b_k]]$ είναι ταυτόσημο ως προς κάθε κοινό βήμα k για μια συνάρτηση ενώ όταν απλά έχουμε μεγαλύτερο l τερματίζεται ο αλγόριθμος πιο γρήγορα. Επίσης στη μέθοδο αυτή το σημείο τερματισμού είναι προδιαγεγραμμένο από την σχέση που παρουσιάστηκε παραπάνω και σαφώς εξαρτάται μόνο από τα l και το αρχικό εύρος αναζήτησης $[a_1, b_1] = [0, 3]$ και όχι από το ποια είναι η κυρτή συνάρτηση. Παρόλα αυτά απαιτεί παραπάνω βήματα k από ότι της διχοτόμου για να τερματίσει.

4 Θέμα 3

4.1 Αλγόριθμος μεθόδου Fibonacci

Ακολουθεί ο κώδικας της υλοποίησης της μεθόδου του Fibonacci στο Matlab. Πρόκειται για την συνάρτηση Fibonacci με ορίσματα το αρχικό εύρος αναζήτησης [ap,bp] , την συνάρτηση fun που θα ελαχιστοποιηθεί, το τελικό εύρος αναζήτησης l και τέλος μια κενή παράμετρο jk που δεν αξιοποιείται στη συνάρτηση αλλά έχει προστεθεί για λόγους συμβατότητας του κώδικα. Ός έξοδο έχουμε τους πίνακες A και B που έχουν τα άκρα αναζήτησης ak και bk αντίστοιχα, συναρτήσε του βήματος k καθώς και τον αριθμό κλήσεων (calls) της αντικειμενικής συνάρτησης κατά την εκτέλεση της μεθόδου. Η διαδικασία είναι αυτή που δίνεται για την μέθοδο απο την επιλεγμένη βιβλιογραφία και θυμίζει στην λογική της αυτή του χρυσού τομέα. Χρησιμοποιούμε επίσης τους πίνακες FX1 και FX2 για να αποθηκεύουμε τις τιμές f(x1k) και f(x2k) της αντικειμενικής συνάρτησης, και τους πίνακες X1,X2 για να αποθηκεύουμε τα x1k,x2k αντίστοιχα που υπολογίζονται σε κάθε βήμα k. Για τον υπολογισμό του δείκτη τερματισμού n, σχηματίσαμε μια δικιά μας λίστα Farr που περιέχει τις τιμές της ακολουθίας Fibonacci από την F_0 έως και την F_n . Λόγω της δεικτοδότησης στη Matlab όπου το πρώτο στοιχείο της λίστας έχει δείκτη 1 , οι συνθήκες του βιβλίου έχουν προσαρμοστεί κατάλληλα ώστε να αξιοποιούν την λίστα Farr για άντληση των τιμών της ακολουθίας Fibonacci και αυτό έγινε με την προσθήκη του +1 στον δείκτη της F. Επίσης λόγω της φύσης του αλγορίθμου στην επανάληψη $k=n-2$, υπολογίζονται τα a_{n-1} , b_{n-1} και προκύπτει ότι $x_{1n-1} = x_{2n-1}$ το οποίο βρίσκεται στη μέση του διαστήματος $[a_{n-1}, b_{n-1}]$ (απο (5.1.7) και (5.1.8. βιβλίου για $k=n-1$). Αυτό είναι αρκετά σημαντικό διότι εισάγοντας μια ικανά μικρή ποσότητα ϵ , ώστε το σημείο x_{1n} να διαφοροποιηθεί από το x_{2n} ως $x_{2n} = x_{1n} + \epsilon$ και το τελικό εύρος αναζήτησης να προκύψει το $[a_n, x_{2n}]$ ή το $[x_{1n}, b_n]$. Επίσης απο την σχέση (5.1.10) του βιβλίου για $k=n-1$ ισχύει:

$$b_n - a_n = \frac{F_1}{F_2}(b_{n-1} - a_{n-1})$$

$$b_n - a_n = \frac{1}{2}(b_{n-1} - a_{n-1})$$

Έχοντας υπόψιν ότι πρέπει να ισχύει $b_n - a_n < l$, τότε πρέπει να ισχύει και :

$$b_{n-1} - a_{n-1} < 2l$$

Επομένως ξέροντας ότι το $x_{1n-1} = \frac{a_{n-1}+b_{n-1}}{2}$, συμπεραίνουμε ότι πράγματι το μήκος του διαστήματος $[x_{1n}, b_n]$ είναι μικρότερο του l. Άρα η επιλογή του ϵ πρέπει να είναι τέτοια ώστε το μήκος του διαστήματος $[a_n, x_{1n} + \epsilon]$ να είναι και αυτό μικρότερο του l. Άρα

$$x_{1n} + \epsilon - a_{n-1} < l \Rightarrow \frac{a_{n-1} + b_{n-1}}{2} + \epsilon - a_{n-1} < l$$

$$\epsilon < l - \frac{a_{n-1} + b_{n-1}}{2}$$

Αυτή η συνθήκη για $l \in [0.001, 0.1]$ ικανοποιείται από την σχέση $\epsilon=0.01l$, ωστόσο μπορούμε να προσδώσουμε ασφάλεια καθορίζοντας δυναμικά το ϵ στο τελευταίο βήμα ως :

$$\epsilon = l - \frac{a_{n-1} + b_{n-1}}{2}$$

Εμείς θα αρκεστούμε στο $\epsilon=0.01l$. Είναι σημαντικό να επισημανθεί ότι όταν $k=n-2$, η εκτίμηση της x_{1n-1} και x_{2n-1} καταλήγει σε $x_{1n-1}=x_{2n-1}$. Γιατί ισχύει αυτό και γνωρίζοντας ότι είτε ισχύει $x_{1n-1}=x_{2n-2}$ ή $x_{1n-1}=x_{2n-1}=x_{1n-2}$, είναι προφανές ότι ήδη έχει υπολογιστεί η τιμή f(x1n-1) και δεν χρειάζεται να ξαναυπολογιστεί.

```
1 function [A,B,calls] = Fibonacci(ap,bp,fun,l,jk)
2 e=0.01*l;
3 fun=matlabFunction(fun);
4 A=ap;
5 B=bp;
6 F0=1;
```



```

7  F1=1;
8  n=1;
9  F=1;
10 pp=F0;
11 Farr=[F0,F1];
12 while (F<=(bp-ap)/l)
13     n=n+1;
14     temp=F;
15     F=F+pp;
16     pp=temp;
17     Farr=[Farr,F];
18 end
19
20 calls=0;
21 x11=ap+(Farr(n-1)/Farr(n+1))*(bp-ap);
22 x21=ap+(Farr(n)/Farr(n+1))*(bp-ap);
23 X1=x11;
24 X2=x21;
25 FX1=fun(x11);
26 FX2=fun(x21);
27 calls=calls+2;
28
29 k=1;
30
31 for k=1:(n-3)
32     if (FX1(k)>FX2(k))
33         an=X1(k);
34         bn=B(k);
35         x1n=X2(k);
36         x2n=an+(Farr(n-k)/Farr(n-k+1))*(bn-an);
37         fx1n=FX2(k);
38         fx2n=fun(x2n);
39         calls=calls+1;
40
41     else
42         an=A(k);
43         bn=X2(k);
44         x1n=an+(Farr(n-k-1)/Farr(n-k+1))*(bn-an);
45         x2n=X1(k);
46         fx2n=FX1(k);
47         fx1n=fun(x1n);
48         calls=calls+1;
49
50
51     end
52
53     A=[A,an];
54     B=[B,bn];
55     FX1=[FX1,fx1n];
56     FX2=[FX2,fx2n];
57     X1=[X1,x1n];
58     X2=[X2,x2n];
59
60 end

```

```

61 k=n-2;
62 if(FX1(k)>FX2(k))
63     an=X1(k);
64     bn=B(k);
65     x1n=X2(k);
66     x2n=an+(Farr(n-k)/Farr(n-k+1))*(bn-an);
67     fx1n=FX2(k);
68     fx2n=fx1n;
69 else
70     an=A(k);
71     bn=X2(k);
72     x1n=an+(Farr(n-k-1)/Farr(n-k+1))*(bn-an);
73     x2n=X1(k);
74     fx2n=FX1(k);
75     fx1n=fx2n;
76
77
78 end
79
80 A=[A,an];
81 B=[B,bn];
82 FX1=[FX1,fx1n];
83 FX2=[FX2,fx2n];
84 X1=[X1,x1n];
85 X2=[X2,x2n];
86
87
88
89
90
91 x1n=X1(end);
92 x2n=x1n+e;
93 X1=[X1,x1n];
94 X2=[X2,x2n];
95 FX1=[FX1,FX1(end)];
96 FX2=[FX2,fun(x2n)];
97 calls=calls+1;
98     if(FX1(end)>FX2(end))
99         an=X1(end);
100         bn=B(end);
101     else
102         an=A(end);
103         bn=X2(end);
104     end
105
106 A=[A,an];
107 B=[B,bn];
108
109
110 end

```

4.2 Κλήσεις αντικειμενικής συνάρτησης σε συνάρτηση με τελικό εύρος αναζήτησης 1

Συνεχίζοντας θα εξετάσουμε την επιρροή που έχει το τελικό εύρος αναζήτησης στον αριθμό των κλήσεων της αντικειμενικής συνάρτησης όταν έχουμε αρχικό διάστημα αναζήτησης $[0,3]$ στη μέθοδο Fibonacci. Υλοποιούμε την μέθοδο σε κάθε συνάρτηση για τιμές του 1 στο $[0.0025,0.1]$.

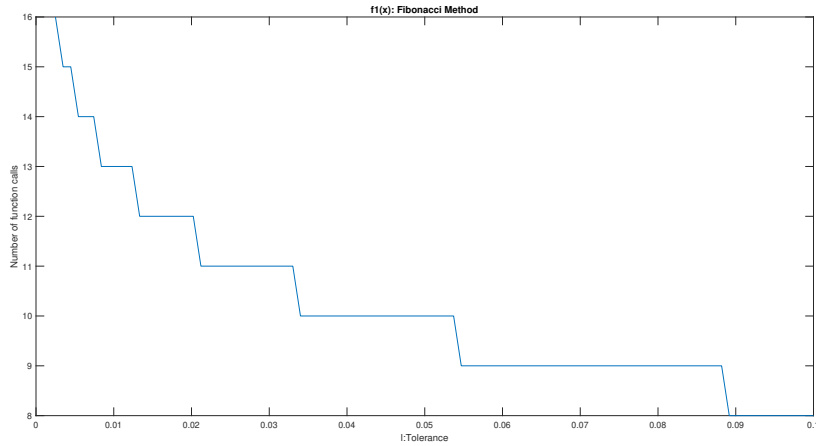


Figure 16: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του 1 στη μέθοδο Fibonacci.

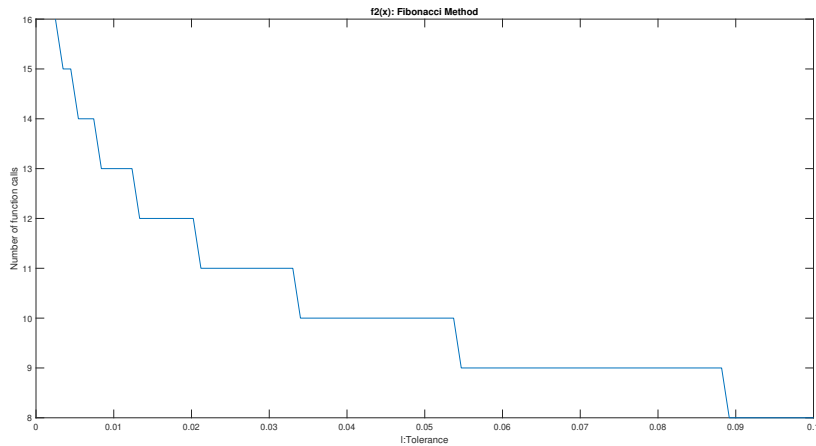


Figure 17: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του 1 στη μέθοδο Fibonacci.

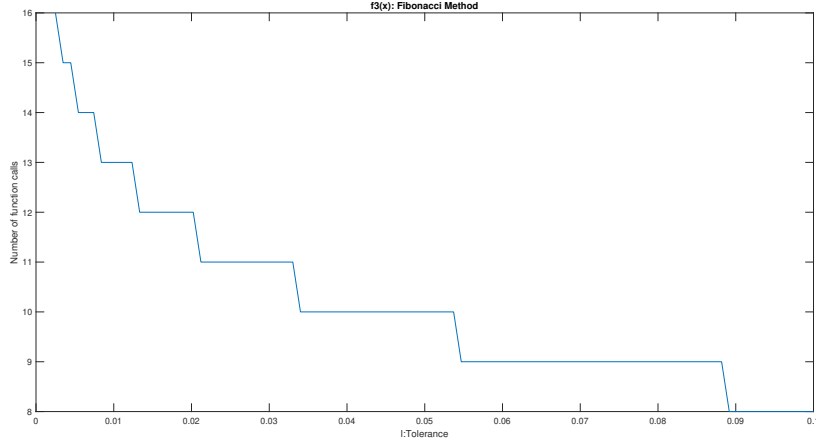


Figure 18: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του l στη μέθοδο Fibonacci.

Παρατηρούμε ότι οι γραφικές παραστάσεις των τριών συναρτήσεων είναι πάλι ταυτόσημες, όπως και στις προηγούμενες μεθόδους και ότι η συμπεριφορά του αριθμού των κλήσεων της αντικειμενικής συνάρτησης σε σχέση με το l μοιάζει με λογαριθμική και είναι φθίνουσα. Όσο αυξάνουμε το εύρος τελικής αναζήτησης l τόσο μικραίνει ο αριθμός κλήσεων της αντικειμενικής συνάρτησης. Αυτό είναι λογικό καθώς σε κάθε βήμα το διάστημα αναζήτησης μειώνεται κατά:

$$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)$$

. Συνεπώς στο τελευταίο βήμα του αλγορίθμου έχουμε:

$$b_n - a_n = \frac{F_1}{F_2}(b_{n-1} - a_{n-1}) = \frac{F_1}{F_2} \cdot \frac{F_2}{F_3}(b_{n-2} - a_{n-2})$$

$$b_n - a_n = \frac{F_1}{F_2} \cdot \frac{F_2}{F_3} \cdot \dots \cdot \frac{F_{n-1}}{F_n}(b_1 - a_1)$$

$$b_n - a_n = \frac{b_1 - a_1}{F_n}$$

Γιατί πρέπει να ισχύει ότι :

$$b_n - a_n < l$$

Καταλήγουμε στο ότι πρέπει να ισχύει για το n δείκτη τερματισμού του αλγορίθμου:

$$F_n \geq \frac{b_1 - a_1}{l}$$

Ο n είναι ο ελάχιστος θετικός ακέραιος που ικανοποιεί την ανίσωση το οποίο υπολογίζουμε και χρησιμοποιούμε μέσα στον αλγόριθμο. Επίσης έχοντας υπόψη ότι έχουμε 2 κλήσεις της αντικειμενικής συνάρτησης στην αρχικοποίηση, $n-3$ κλήσεις στα ενδιάμεσα βήματα και μια τελευταία κλήση στο τελευταίο βήμα. Άρα ο αριθμός των κλήσεων της αντικειμενικής συνάρτησης είναι ίδιος με τον αριθμό n που ικανοποιεί την παραπάνω συνάρτηση. Πράγματι παρατηρούμε ότι αυτή η ανίσωση επαληθεύεται από τις άνωθεν γραφικές παραστάσεις. Επίσης καθίσταται προφανές ότι το πλήθος των βημάτων άρα και κλήσεων της αντικειμενικής συνάρτησης εξαρτάται μόνο από το αρχικό διάστημα $[a_1, b_1]$ και το τελικό εύρος αναζήτησης l , και όχι από την εκάστοτε κυρτή συνάρτηση, κάτι που ερμηνεύει γιατί τα τρία διαγράμματα ταυτίζονται. Παρατηρούμε ότι για δεδομένο αρχικό εύρος αναζήτησης $[a_1, b_1] = [0, 3]$ και τελικό εύρος αναζήτησης l , η μέθοδος Fibonacci απαιτεί λιγότερες κλήσεις της αντικειμενικής συνάρτησης από ότι απαιτεί η μέθοδος του διχοτόμου και η μέθοδος της χρυσής τομής. Για παράδειγμα, για $l=0.01$ η μέθοδος διχοτόμου απαιτεί 18 κλήσεις, της χρυσής τομής 14 κλήσεις, ενώ του Fibonacci. Σαφώς αυτό σημαίνει ότι η μέθοδος του Fibonacci είναι πιο αποδοτική από αυτές της διχοτόμου και της χρυσής τομής.

4.3 Διαγράμματα διαστημάτων αναζήτησης $[a_k, b_k]$ συναρτήσεως του δείκτη βήματος

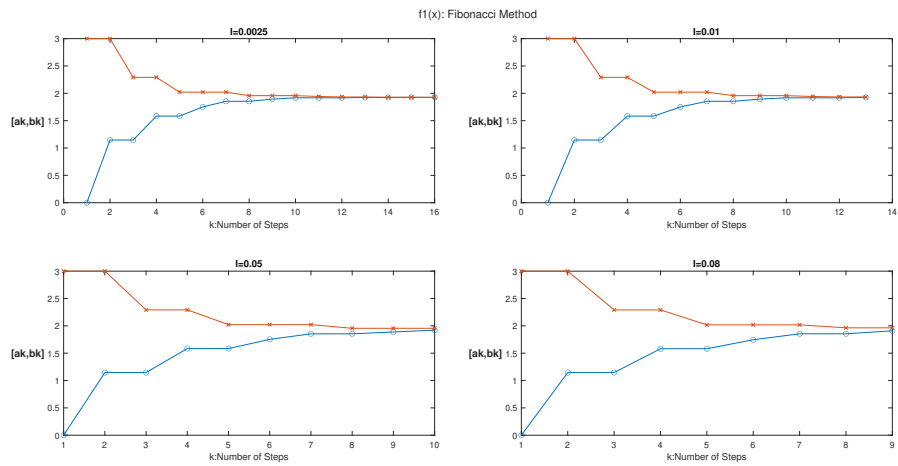


Figure 19: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσεως του δείκτη βήματος k στη μέθοδο Fibonacci.

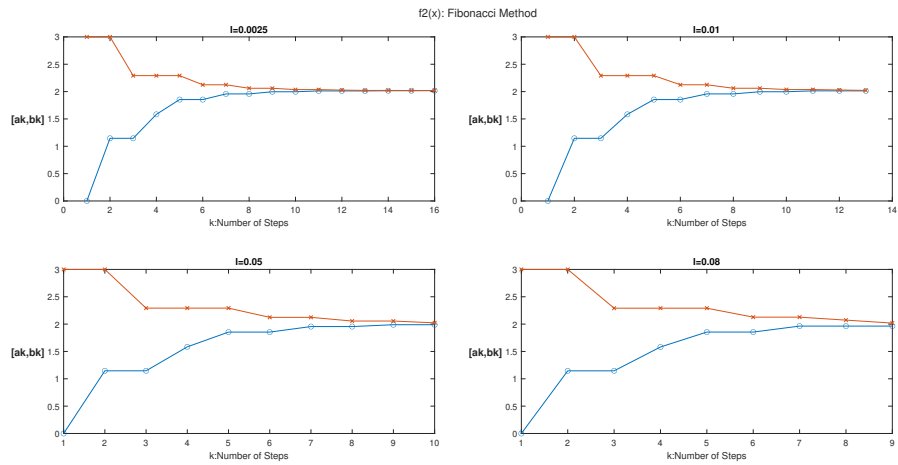


Figure 20: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσεως του δείκτη βήματος k στη μέθοδο Fibonacci.

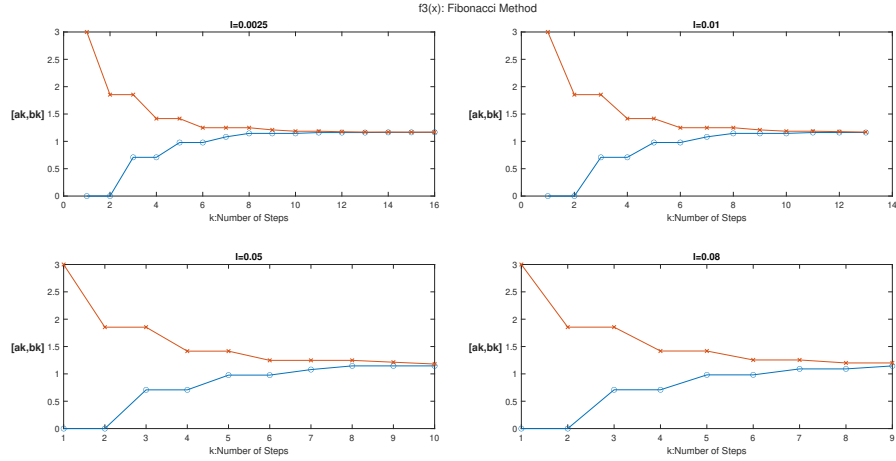


Figure 21: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσεϊ του δείκτη βήματος k στη μέθοδο Fibonacci.

Παρατηρούμε όπως και στη προηγούμενη μέθοδο από τα διαγράμματα ότι όσο μικρότερο είναι το l , τόσα περισσότερα βήματα απαιτούνται για τον τερματισμό του αλγορίθμου άλλα και τόσο πιο κοντά είναι μεταξύ τους τα τελικά άκρα του διαστήματος $[a_k, b_k]$ όταν η μέθοδος ολοκληρώνεται. Άρα έχουμε καλύτερη προσέγγιση για το σημείου ελαχίστου για το μικρότερο l . Λόγω της φύσης του αλγορίθμου, για τις διάφορες τιμές του l , το διάγραμμα $[k, [a_k, b_k]]$ είναι ταυτόσημο ως προς κάθε κοινό βήμα k για την ίδια συνάρτηση ενώ όταν απλά έχουμε μεγαλύτερο l τερματίζεται ο αλγόριθμος πιο γρήγορα. Επίσης στη μέθοδο αυτή το σημείο τερματισμού είναι προδιαγεγραμμένο από την σχέση που παρουσιάστηκε παραπάνω και σαφώς εξαρτάται μόνο από τα l και το αρχικό εύρος αναζήτησης $[a_1, b_1] = [0, 3]$ και όχι από το ποια είναι η κυρτή συνάρτηση. Απαιτεί περίπου τα ίδια βήματα k με χρυσή τομή. Άρα από τις 3 μεθόδους ελαχιστοποίησης κυρτής συνάρτησης χωρίς την χρήση παραγώγου είναι η πιο αποδοτική η Fibonacci.

5 Θέμα 4

5.1 Αλγόριθμος μεθόδου Διχοτόμου με χρήση παραγώγου

Ακολουθεί ο κώδικας της υλοποίησης της μεθόδου της Διχοτόμου με χρήση παραγώγου στο Matlab. Πρόκειται για την συνάρτηση `DichotomosDerivative` με ορίσματα το αρχικό εύρος αναζήτησης `[ap,bp]`, την συνάρτηση `fun` που θα ελαχιστοποιηθεί, το τελικό εύρος αναζήτησης `l` και τέλος την `e` που δεν αξιοποιείται στη συνάρτηση αλλά έχει προστεθεί για λόγους συμβατότητας του κώδικα. Ός έξοδο έχουμε τους πίνακες `A` και `B` που έχουν τα άκρα αναζήτησης `ak` και `bk` αντίστοιχα, συναρτήσε του βήματος `k` καθώς και τον αριθμό κλήσεων (`calls`) της αντικειμενικής συνάρτησης κατά την εκτέλεση της μεθόδου. Η διαδικασία είναι αυτή που δίνεται για την μέθοδο απο την επιλεγμένη βιβλιογραφία. Χρησιμοποιούμε επίσης τους πίνακες `DF` για να αποθηκεύουμε τις τιμές της παραγώγου της αντικειμενικής συνάρτησης $f_i(x)$. Η διαφορά αυτής της μεθόδου από τις προηγούμενες απαιτεί την διαφορισιμότητα της συνάρτησης στο αρχικό εύρος αναζήτησης `[0,3]` σε αντίθεση με την προηγούμενη. Μάλιστα η παράγωγος αυτή πρέπει να μας είναι γνωστή. Σαφώς είναι πιο περιορισμένη για εφαρμογές από τις προηγούμενες.

```
1 function [A,B,calls] = DichotomosDerivative(ap,bp,fun,l,e)
2 syms x;
3 fun(x)=diff(fun,x);
4 fun=matlabFunction(fun);
5 A=ap;
6 B=bp;
7 DF=[];
8 n=0;
9 calls=0;
10 while ((1/2) ^n>1/(bp-ap))
11     n=n+1;
12 end
13 k=1;
14
15 for i=1:n
16     xk=(A(end)+B(end))/2;
17     df=fun(xk);
18     calls=calls+1;
19     if (df==0)
20         an=xk;
21         bn=xk;
22         break;
23     elseif (df>0)
24         an=A(end);
25         bn=xk;
26     else
27         an=xk;
28         bn=B(end);
29     end
30     A=[A,an];
31     B=[B,bn];
32     DF=[DF,df];
33     xf=xk;
34 end
35
36
37
38 end
```

5.2 Κλήσεις αντικειμενικής συνάρτησης σε συνάρτηση με τελικό εύρος αναζήτησης

Συνεχίζοντας θα εξετάσουμε την επιρροή που έχει το τελικό εύρος αναζήτησης στον αριθμό των κλήσεων της παραγώγου της αντικειμενικής συνάρτησης όταν έχουμε αρχικό διάστημα αναζήτησης $[0,3]$ στη μέθοδο της Διχοτόμου με την χρήση παραγώγου. Υλοποιούμε την μέθοδο σε κάθε συνάρτηση για τιμές του l στο $[0.0025, 0.1]$.

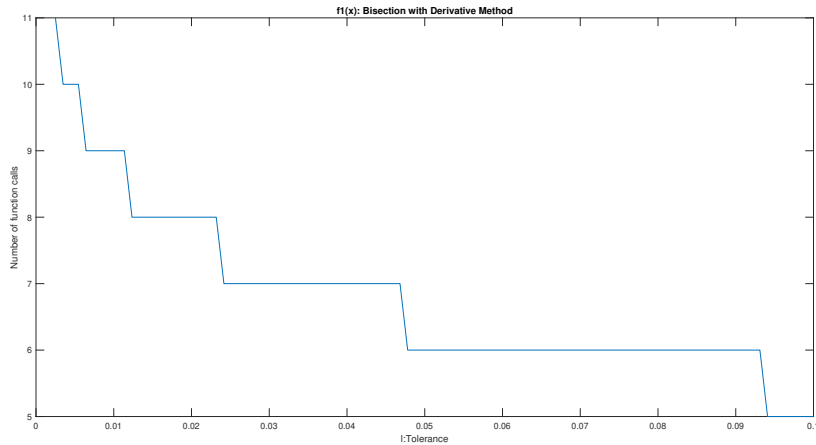


Figure 22: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του l στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

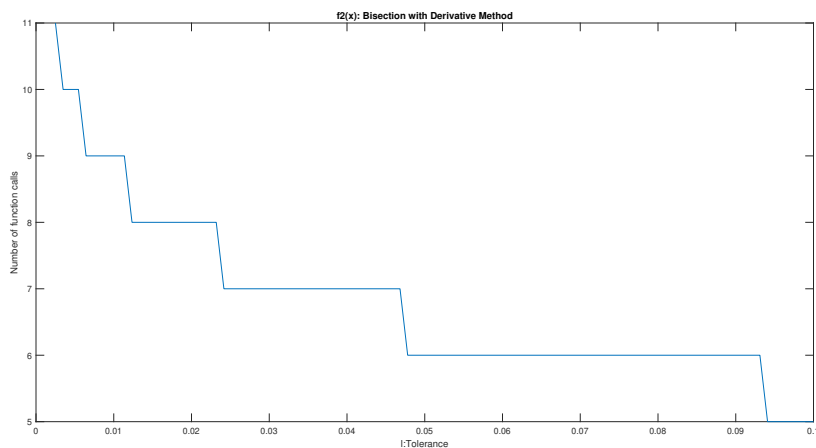


Figure 23: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του l στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

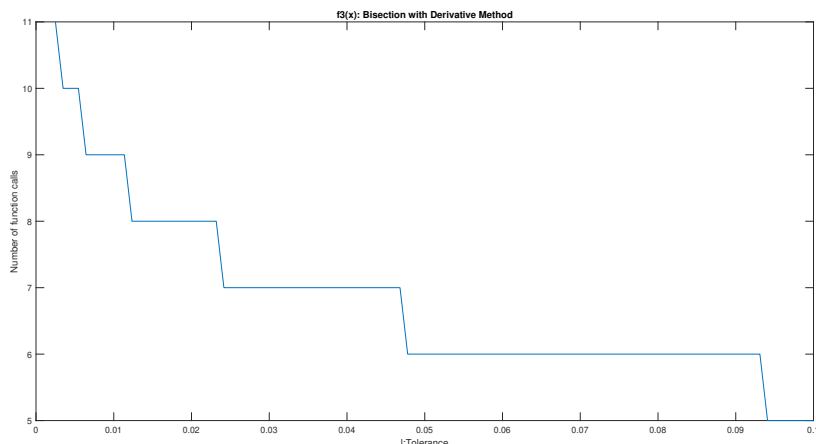


Figure 24: Γραφική Παράσταση του αριθμού κλήσεων της αντικειμενικής συνάρτησης εν συναρτήσει του l στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

Παρατηρούμε ότι οι γραφικές παραστάσεις των τριών συναρτήσεων είναι πάλι ταυτόσημες, όπως και στις προηγούμενες μεθόδους και ότι η συμπεριφορά του αριθμού των κλήσεων της αντικειμενικής συνάρτησης σε σχέση με το l μοιάζει με λογαριθμική και είναι φθίνουσα. Όσο αυξάνουμε το εύρος τελικής αναζήτησης l τόσο μικραίνει ο αριθμός κλήσεων της παραγώγου της αντικειμενικής συνάρτησης πράγμα που είναι λογικό όπως θα αποδείξουμε στην συνέχεια. Κατ' αρχάς, κάθε διάστημα αναζήτησης συνδέεται με το προηγούμενο με την σχέση $L_{k+1} = \frac{L_k}{2}$. Συνεπώς το τελικό διάστημα αναζήτησης είναι:

$$L_{n+1} = \left(\frac{1}{2}\right)^n (b1 - a1)$$

Άρα πρέπει να ισχύει ότι :

$$\left(\frac{1}{2}\right)^n \leq \frac{l}{b1 - a1}$$

Ο n είναι ο ελάχιστος θετικός ακέραιος που ικανοποιεί την ανίσωση και επομένως ο αριθμός των κλήσεων της παραγώγου της αντικειμενικής συνάρτησης. Πράγματι παρατηρούμε ότι αυτή η ανίσωση επαληθεύεται από τις άνωθεν γραφικές παραστάσεις. Επίσης καθίσταται προφανές ότι το πλήθος των βημάτων άρα και κλήσεων της αντικειμενικής συνάρτησης εξαρτάται μόνο από το αρχικό διάστημα $[a1, b1]$ και το τελικό εύρος αναζήτησης l , και όχι από την εκάστοτε κυρτή συνάρτηση, κάτι που ερμηνεύει γιατί τα τρία διαγράμματα ταυτίζονται. Βέβαια ο αλγόριθμος τερματίζει νωρίτερα εάν βρεθούμε σε x_k τέτοιο που μηδενίζει την παράγωγο. Αλλά αυτό αποτελεί μια πολύ ειδική περίπτωση καθώς τότε το ελάχιστο πρέπει να ανήκει σε κάποιο από τα σημεία διχοτόμησης που είναι πεπερασμένα για γνωστό l . Τέλος, για δεδομένο αρχικό εύρος αναζήτησης $[a1, b1] = [0, 3]$ και τελικό εύρος αναζήτησης l , η μέθοδος Διχοτόμου με χρήση παραγώγου απαιτεί λιγότερους υπολογισμούς συνάρτησης από ότι απαιτούν οι 3 μέθοδοι χωρίς την χρήση παραγώγων που εξετάσαμε προηγουμένως. Μάλιστα είναι μόλις οι μισές από αυτούς που χρειάζεται η μέθοδος της διχοτόμου χωρίς παράγωγο για το ίδιο πλητικό $\frac{l}{b1 - a1}$. Άρα είναι η πιο αποδοτική από τις μεθόδους.

5.3 Διαγράμματα διαστημάτων αναζήτησης $[a_k, b_k]$ συναρτήσεως του δείκτη βήματος

Στη συνέχεια θα ανάπαρστήσουμε γραφικά το διάστημα αναζήτησης $[a_k, b_k]$ για δείκτη βήματος k στην εκτέλεση της μεθόδου, για διάφορες τιμές του l έχοντας το ίδιο αρχικό εύρος αναζήτησης $[0, 3]$.

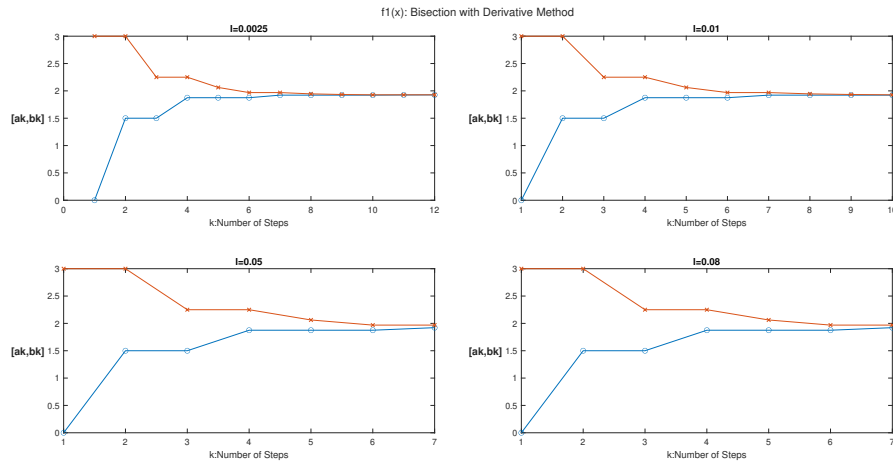


Figure 25: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσεως του δείκτη βήματος k στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

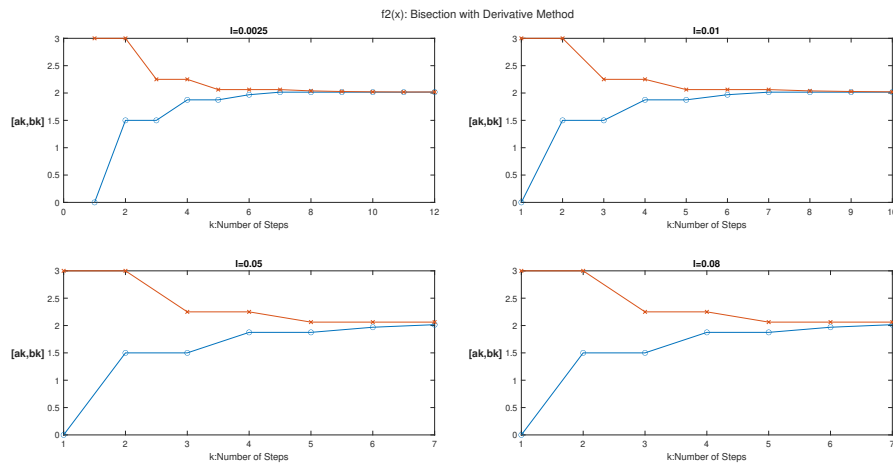


Figure 26: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συναρτήσεως του δείκτη βήματος k στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

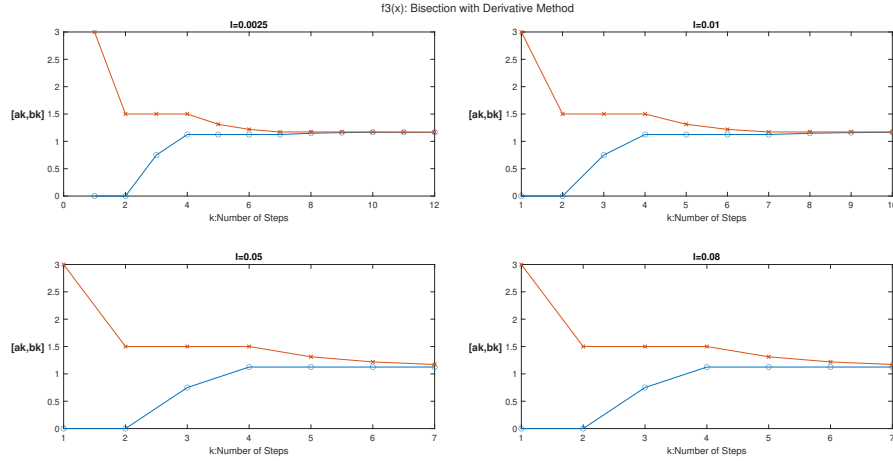


Figure 27: Γραφική Παράσταση των άκρων $[a_k, b_k]$ εν συνάρτησει του δείκτη βήματος k στη μέθοδο Διχοτόμου με την χρήση παραγώγου.

Παρατηρούμε όπως και στη προηγούμενη μέθοδο από τα διαγράμματα ότι όσο μικρότερο είναι το l , τόσο περισσότερα βήματα απαιτούνται για τον τερματισμό του αλγορίθμου άλλα και τόσο πιο κοντά είναι μεταξύ τους τα τελικά άκρα του διαστήματος $[a_k, b_k]$ όταν η μέθοδος ολοκληρώνεται. Άρα έχουμε καλύτερη προσέγγιση για το σημείου ελαχίστου για το μικρότερο l . Λόγω της φύσης του αλγορίθμου, για τις διάφορες τιμές του l , το διάγραμμα $[k, [a_k, b_k]]$ είναι ταυτόσημο ως προς κάθε κοινό βήμα k για την ίδια συνάρτηση ενώ όταν απλά έχουμε μεγαλύτερο l τερματίζεται ο αλγόριθμος πιο γρήγορα. Επίσης στη μέθοδο αυτή το σημείο τερματισμού είναι προδιαγεγραμμένο από την σχέση που παρουσιάστηκε παραπάνω και σαφώς εξαρτάται μόνο από τα l και το αρχικό εύρος αναζήτησης $[a_1, b_1] = [0, 3]$ και όχι από το ποιά είναι η κυρτή συνάρτηση, εκτός από την ειδική περίπτωση όπου το σημείο ελαχίστου εμφανίζεται σε σημείο διχοτόμησης. Συγκρίνοντας με τα αντίστοιχα διαγράμματα των προηγούμενων μεθόδων, για το ίδιο πηλίκο $\frac{l}{b_1 - a_1}$ η μέθοδος αυτής της ενότητας έχει λιγότερα βήματα εκτός από της απλής διχοτόμου για $\varepsilon \ll 1$ που έχουν τα ίδια βήματα k .

6 Σύγκριση μεθόδων

Έχοντας πραγματοποιήσει την επίλυση του κάθε θέματος της εργασίας, τώρα μπορούμε να συγκεντρώσουμε όλα τα άνωθεν συμπεράσματα για δώσουμε μια πιο ολοκληρωμένη εικόνα περί της συγκριτικής αποδοτικότητας των 4 μεθόδων. Καταρχάς, όπως αναλύθηκε επανειλημμένα σε κάθε θέμα οι υπολογισμοί τιμών της αντικειμενικής συνάρτησης (ή της παραγώγου της) άρα και ο αριθμός των επαναλήψεων, εξαρτάται αποκλειστικά από το αρχικό εύρος αναζήτησης $[0,3]$, το τελικό εύρος αναζήτησης l και προφανώς από την μέθοδο που αξιοποιείται. Συνεπώς αφού και για τις τρεις συναρτήσεις εφαρμόζουμε αναζήτηση ελαχίστου μέσα στο αρχικό διάστημα $[0,3]$, για να έχουμε δικαία σύγκριση πρέπει να καταμετρήσουμε τον αριθμό υπολογισμών της αντικειμενικής συναρτήσεων που προκύπτει σε κάθε μέθοδο, για το ίδιο l . Προφανώς η πιο αποδοτική μέθοδος είναι αυτή που απαιτεί μικρότερο αριθμό υπολογισμών. Ακολουθεί πίνακας που δείχνει τον αριθμό των υπολογισμών της αντικειμενικής συνάρτησης για συγκεκριμένη μέθοδο και τιμή l .

| Μέθοδος | $l=0.01$ | $l=0.05$ | $l=0.1$ |
|--------------------------------|----------|----------|---------|
| Διχοτόμου ($\epsilon=0.001$) | 18 | 12 | 10 |
| Χρυσής Τομής | 14 | 11 | 10 |
| Fibonacci | 13 | 10 | 8 |
| Διχοτόμου με χρήση παραγώγου | 9 | 6 | 5 |

Table 1: Πίνακας σύγκρισης του αριθμού υπολογισμών συνάρτησης για κάθε μέθοδο, για διαφορετικές τιμές του l .

Σε αυτό τον συγκεντρωτικό πίνακα τα συμπεράσματα που προαναφέραμε γίνονται εμφανή. Η πιο αποδοτική μέθοδος είναι αυτή της διχοτόμου με χρήση παραγώγων καθώς απαιτεί τις λιγότερες κλήσεις. Η πιο αποδοτική από τις μεθόδους δίχως την χρήση παραγώγων είναι η μέθοδος του Fibonacci και η αμέσως επόμενη είναι η μέθοδος του χρυσού τομέα. Οι δυο μέθοδοι μοιάζουν αρκετά η μια με την άλλη και αυτό είναι κατί που περιμέναμε από την θεωρία του βιβλίου. Τέλος η λιγότερη αποδοτική μέθοδος είναι αυτή της διχοτόμου χωρίς παράγωγο η οποία απαιτεί τον μεγαλύτερο αριθμό κλήσεων και μάλιστα για $\epsilon \ll l$ (άρα π.χ. για $\epsilon=0.001$ και για $l \geq 0.01$) απαιτεί ακριβώς διπλάσιο αριθμό υπολογισμών από την μέθοδο της διχοτόμου με την χρήση παραγώγου, όπως προκύπτει από την ανάλυση στο θέμα 1. Παρόμοια συμπεράσματα εξάγονται από τα γραφήματα και για τον αριθμό βημάτων k έως τον τερματισμό του αλγορίθμου.