2018

Άσκηση 1 : Εύρεση ελαχίστου σε κυρτές συναρτήσεις μιας μεταβλητής

ΓΙΩΡΓΟΣ ΜΠΑΛΑΟΥΡΑΣ (8861)

Τεχνικές Βελτιστοποίησης

Ονοματεπώνυμο: *Μπαλαούρας Γεώργιος*

ΑΕΜ: *8861*

E-mail: *mpalaourg@ece.auth.gr*

Εξάμηνο: *6ο Ηλεκτρονικής*

Άσκηση 1 : *Εύρεση ελαχίστου σε κυρτές συναρτήσεις μιας μεταβλητής*

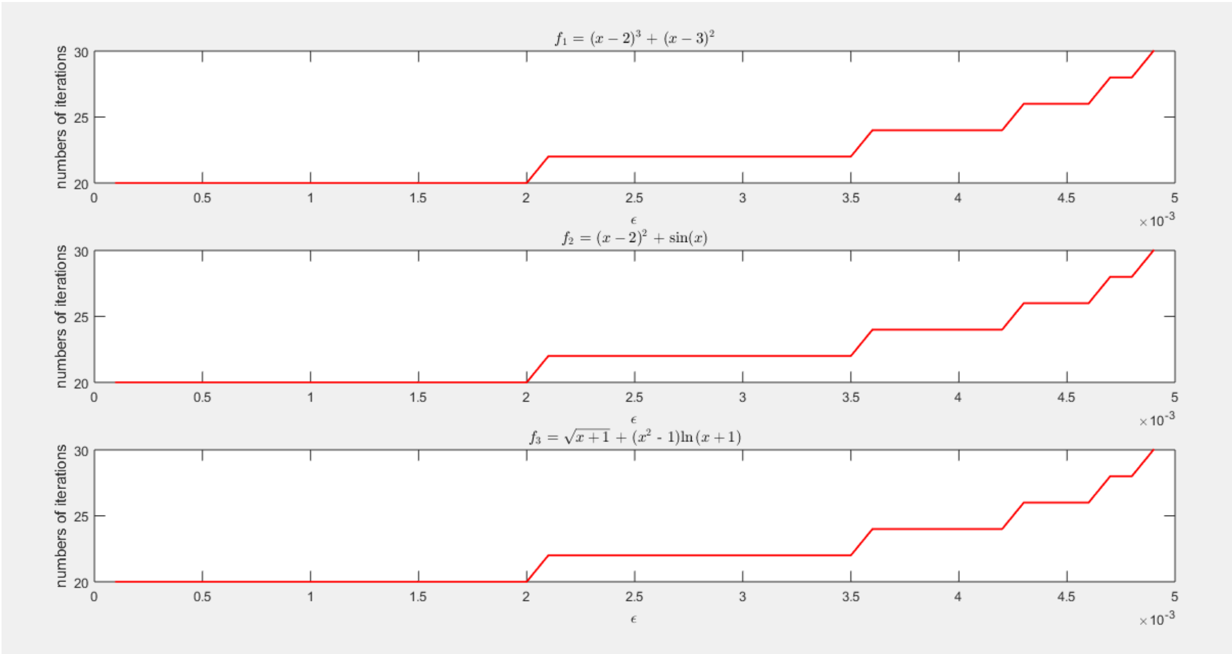
Ο σκοπός αυτής της άσκησης είναι να συγκριθούν τα αποτελέσματα διαφορών αλγορίθμων αναζήτησης, που εφαρμόζονται σε **αυστηρά σχεδόν κυρτές** συναρτήσεις.

Μέθοδος Διχοτόμου: Η μέθοδος αυτή εκμεταλλεύεται ότι οι $f(x)$, είναι **αυστηρά σχεδόν-κυρτές** στο διάστημα $[0,6]$ και έτσι περιορίζει το διάστημα στο $[\alpha, x_2)$ ή το $(x_1, \beta]$ ανάλογα με το $f(x_1) < f(x_2)$ ή $f(x_1) > f(x_2)$ αντίστοιχα. Ο περιορισμός συνεχίζεται έως ότου το διάστημα να έχει εύρος μικρότερο από l. Τέλος, για να χωρίζεται το διάστημα σε ίσα υπο-διαστήματα, τα x_1 , x_2 επιλέγονται συμμετρικά σε απόσταση $\varepsilon > 0$ από την διχοτόμο του.

- Για σταθερό $l = 0.01$:

Δίνεται σαν όρισμα στο `bisection_plot_for_fixed_lamda`, διάφορες τιμές του ε και παρατηρούμε την αύξηση των κλήσεων της αντικειμενικής συνάρτησης. Για τιμές $\varepsilon \geq \frac{l}{2}$ ο αλγόριθμος δεν τερματίζει, ενώ για τιμές κοντά στο $\frac{l}{2}$ οι κλήσεις αυξάνονται σημαντικά.

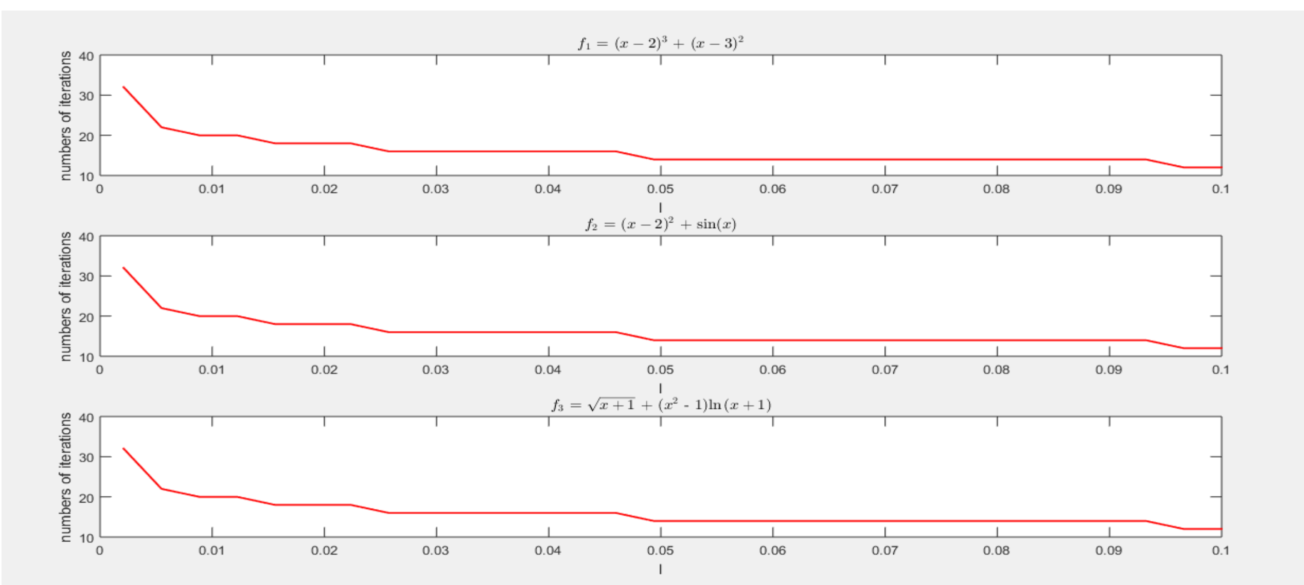
Για το ε, ισχύει $0.0001 \leq \varepsilon \leq 0.0049$ και επιλέγονται 50 σημεία.



Εικόνα 1: Κλήσεις της αντικειμενικής συνάρτησης για διάφορα ε [Μέθοδος Διχοτόμησης]

- Για σταθερό $\varepsilon = 0.001$:

Αντίστοιχα με πριν, δίνεται σαν όρισμα στο `bisection_plot_for_fixed_epsilon`, διάφορες τιμές του l και παρατηρούμε την αύξηση των κλήσεων της αντικειμενικής συνάρτησης.

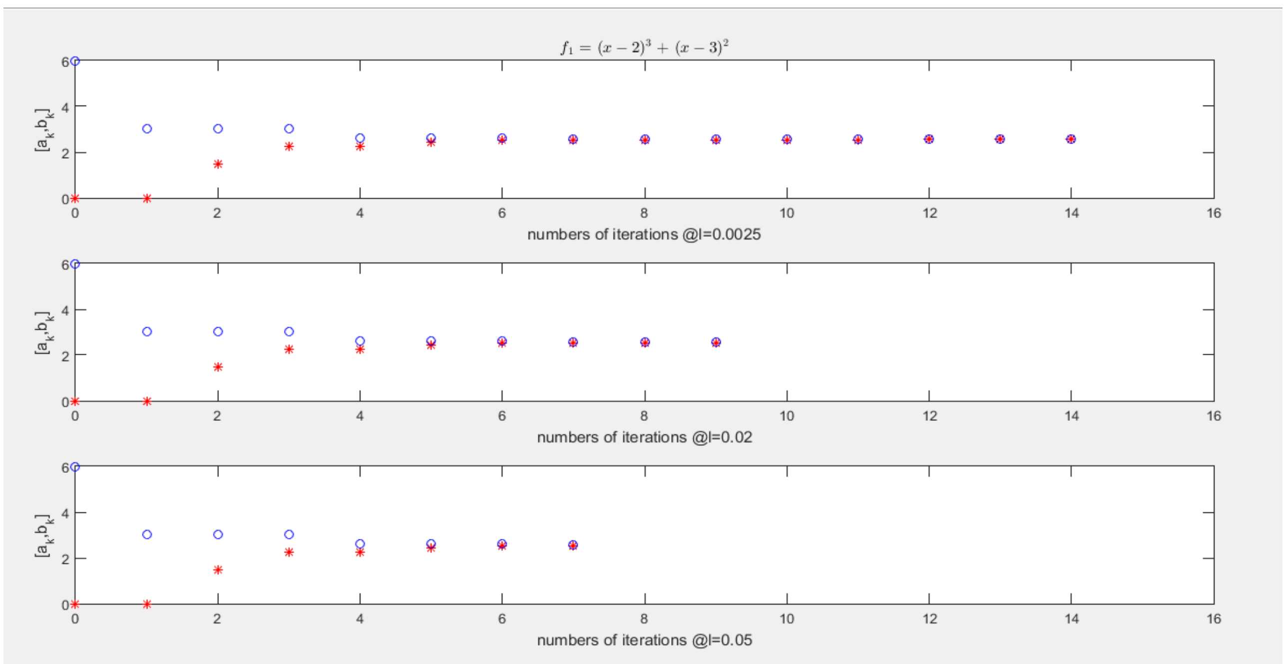


Εικόνα 2: Κλήσεις της αντικειμενικής συνάρτησης για διάφορα l [Μέθοδος Διχοτόμησης]

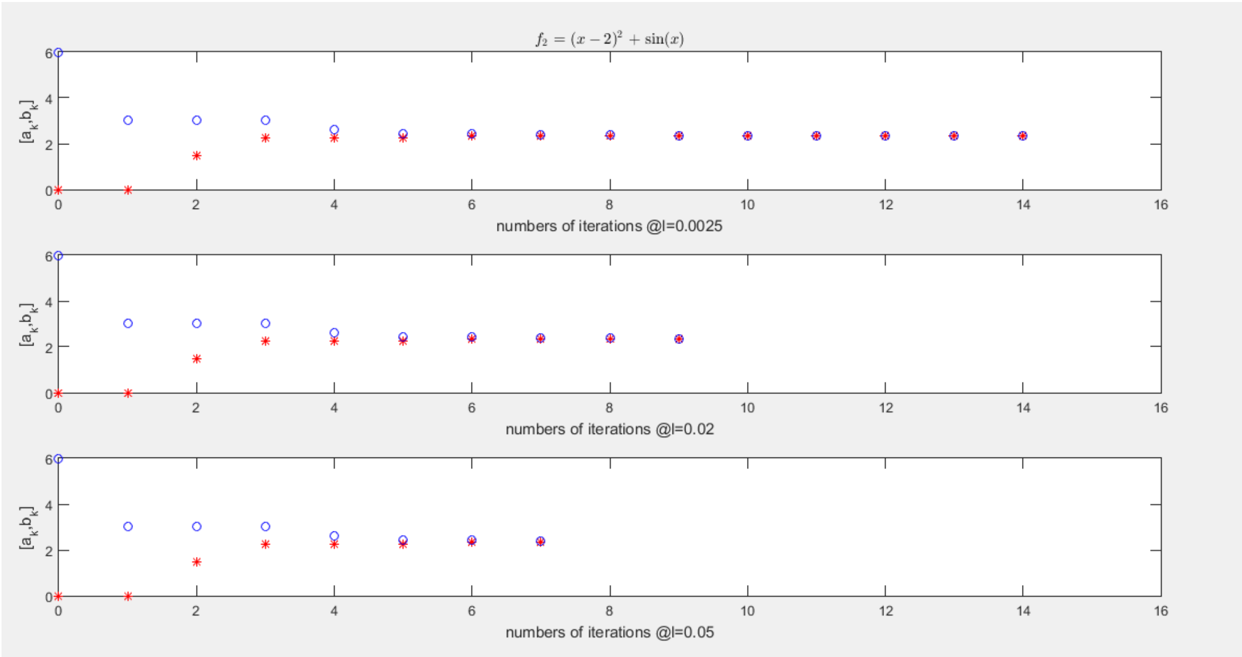
Για τις τιμές του l ισχύει, $0.0021 \leq l \leq 0.1$ και επιβεβαιώνεται ότι ο αριθμός των υπολογισμών της αντικειμενικής συνάρτησης είναι -ο μικρότερο θετικός ακέραιος- που ικανοποιεί την σχέση: $n \geq 2 * \log_2 \frac{b_1 - a_1}{l - 2\varepsilon}$.

Πιο συγκεκριμένα, σε κάθε κλήση της `bisection.m` υπολογίζονται τα x_1, x_2 και στην εντολή `if f(x_1) < f(x_2)`, έχουμε δύο κλήσεις της αντικειμενικής συνάρτησης. Έτσι, ο αριθμός των κλήσεων n συνδέεται με τον αριθμό των επαναλήψεων κ με την σχέση: $n = 2 * (\kappa - 1)$, καθώς αρχικά $\kappa = 1$.

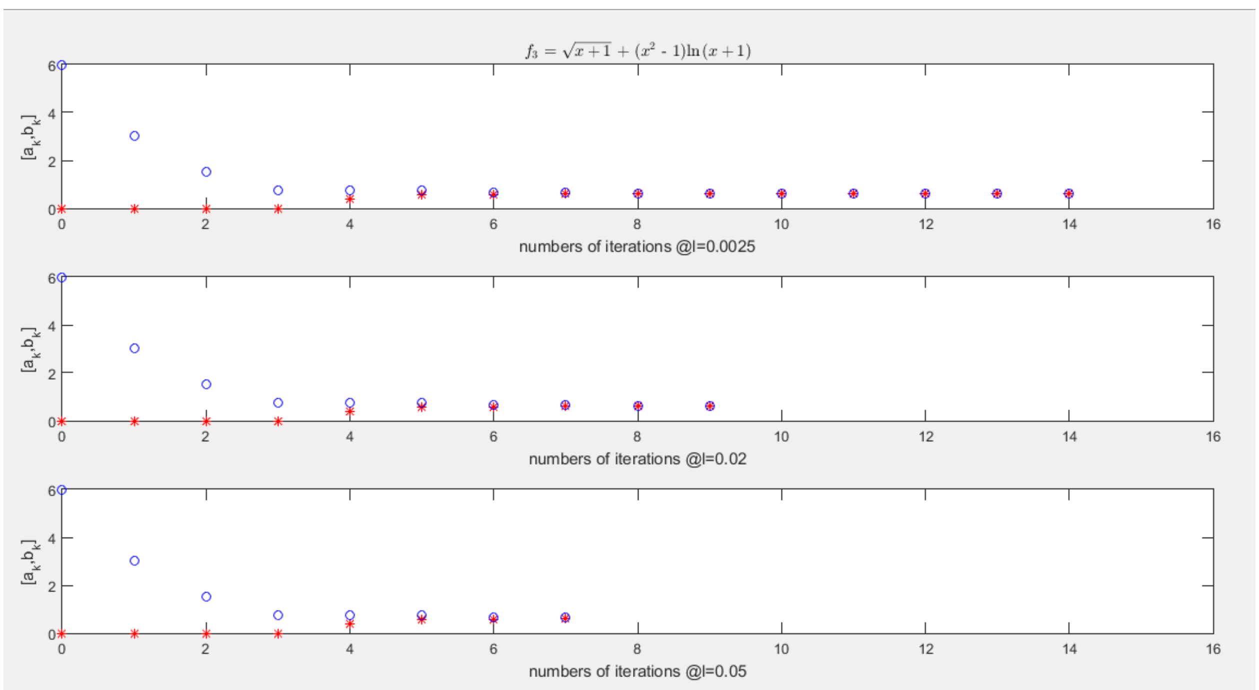
Τέλος, τα διαγράμματα με τον περιορισμό του διαστήματος αναζήτησης $[a_\kappa, b_\kappa]$ ανά επανάληψη κ, αποδεικνύουν ότι το εύρος του εκάστοτε διαστήματος περιορίζεται και συγκλίνει ο αλγόριθμος στην λύση.



Εικόνα 3: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_1 [Μέθοδος Διχοτόμησης]



Εικόνα 4: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_2 [Μέθοδος Διχοτόμησης]



Εικόνα 5: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_2 [Μέθοδος Διχοτόμησης]

Με κόκκινο * αποτυπώνεται το a_κ και με μπλε o το b_κ.

<u>Μέθοδος Χρυσού τομέα</u>: Η μέθοδος του χρυσού τομέα εκμεταλλεύεται την ίδια ιδιότητα των **αυστηρά σχεδόν-κυρτών**, στο διάστημα $[0,6], f(x)$. Ωστόσο, σ' αυτή την μέθοδο τα δύο διαστήματα που προκύπτουν κατά την k-οστή επανάληψη, ικανοποιούν την σχέση:

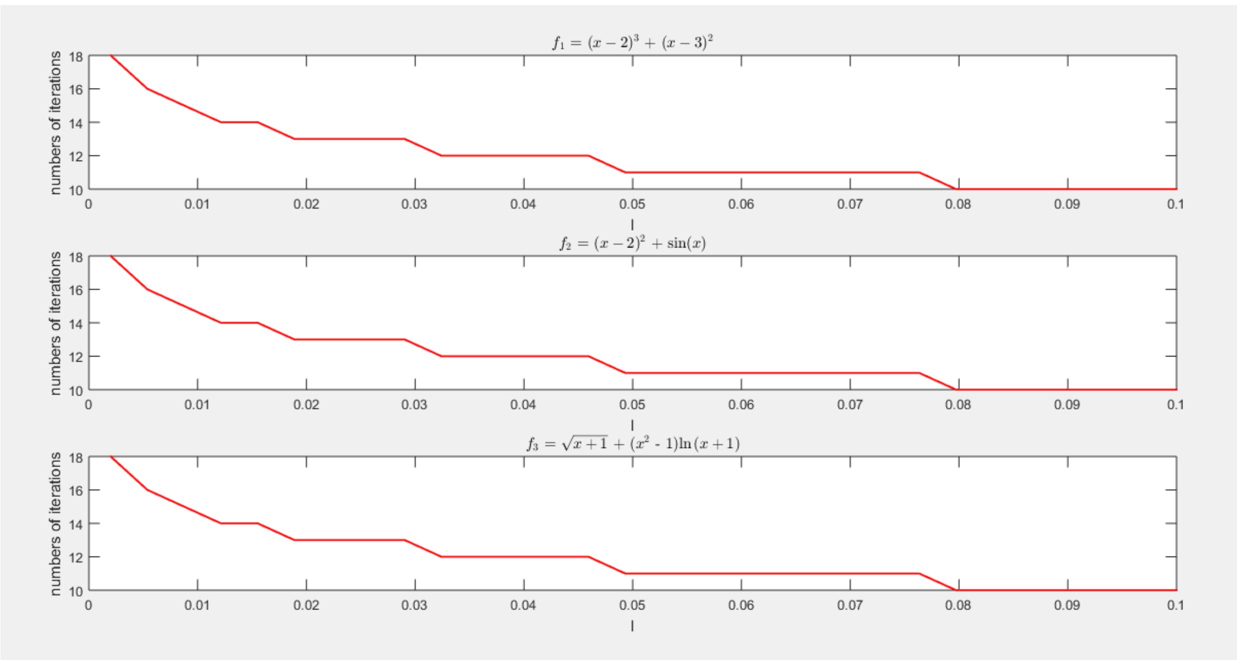
$$\frac{b_{k+1} - a_{k+1}}{b_k - a_k} = \gamma \text{ , } \mu\varepsilon \ \gamma \in (0,1) \qquad (1)$$

Αφού τα $x_{1k}, x_{2k} \in (\alpha_k, b_k)$ –άρα και στην ευθεία που ορίζεται- τότε μπορούμε να επιλέξουμε:

$$x_{1k} = a_k + (1-\gamma)(b_k - a_k) \ \kappa\alpha\iota \ x_{2k} = a_k + \gamma(b_k - a_k)$$

Τέλος, γνωρίζουμε ότι σε κάθε νέα επανάληψη τα x_{1k+1} και x_{2k+1} είναι τέτοια ώστε $x_{1k+1} = x_{2k}$ ή $x_{2k+1} = x_{1k}$. Παράλληλα, ο περιορισμός του διαστήματος σε $[\alpha, x_2)$ ή $(x_1, \beta]$ ανάλογα με το $f(x_1) < f(x_2)$ ή $f(x_1) > f(x_2)$ και η (1) συνεπάγονται ότι $\gamma = 0.618$.

Χρησιμοποιώντας το `golden_section_plot_for_different_lamda`, μελετάω τον αριθμό των κλήσεων της συνάρτησης για διαφορετικά l :



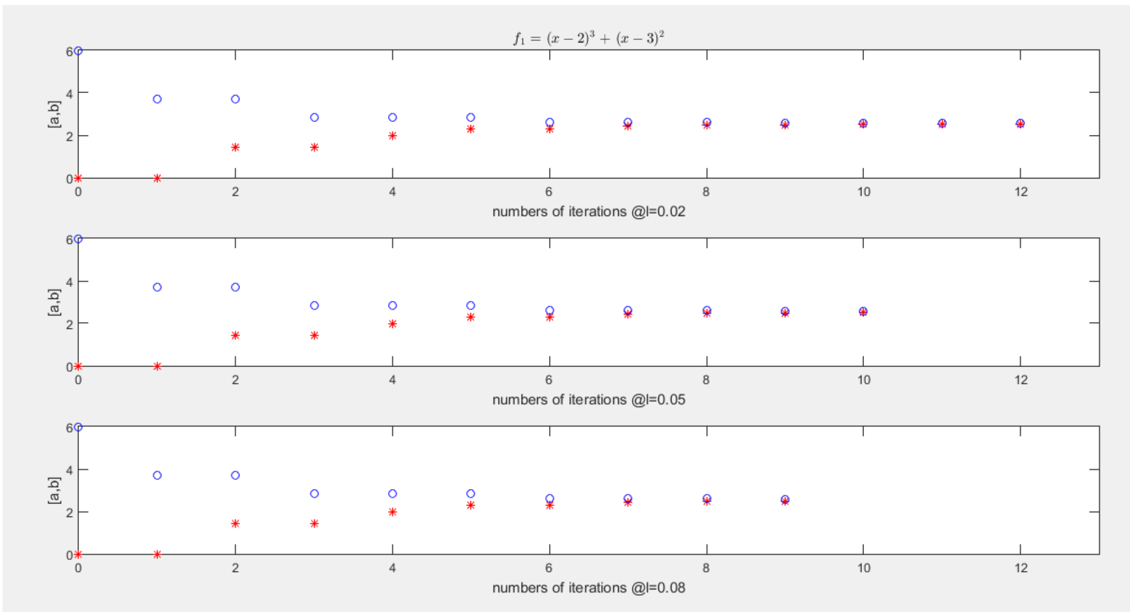
Εικόνα 6:Κλήσεις της αντικειμενικής συνάρτησης για διάφορα l [Μέθοδος Χρυσού Τομέα]

Όπου, επιβεβαιώνεται η θεωρητική σχέση $(0.618)^{n-1} \geq \frac{l}{b_1 - a_1}$ όπου: n ο αριθμός των κλήσεων της αντικειμενικής συνάρτησης και l το τελικό εύρος.

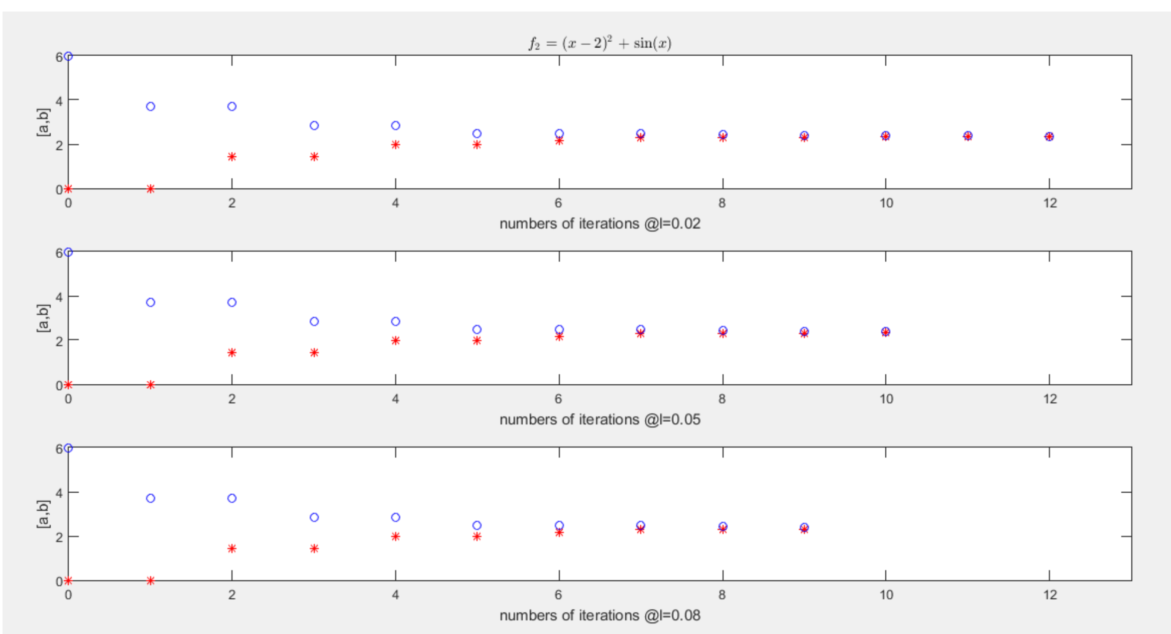
Πιο συγκεκριμένα, την πρώτη φορά που καλείται η `golden_section.m`, υπολογίζονται τα x_1, x_2 και στην εκτέλεση της εντολής `if f(x_1) > f(x_2)` έχουμε 2 κλήσεις τις f. Από την επόμενη, όμως, επανάληψη είτε το x_1 είτε το x_2είναι γνωστά και έτσι έχουμε 1 κλήση της f.

Άρα, $n = 2 + \underbrace{1 + 1 + \cdots + 1}_{\kappa - 1} - 1 = \kappa$, καθώς αρχικά $\kappa = 1$.

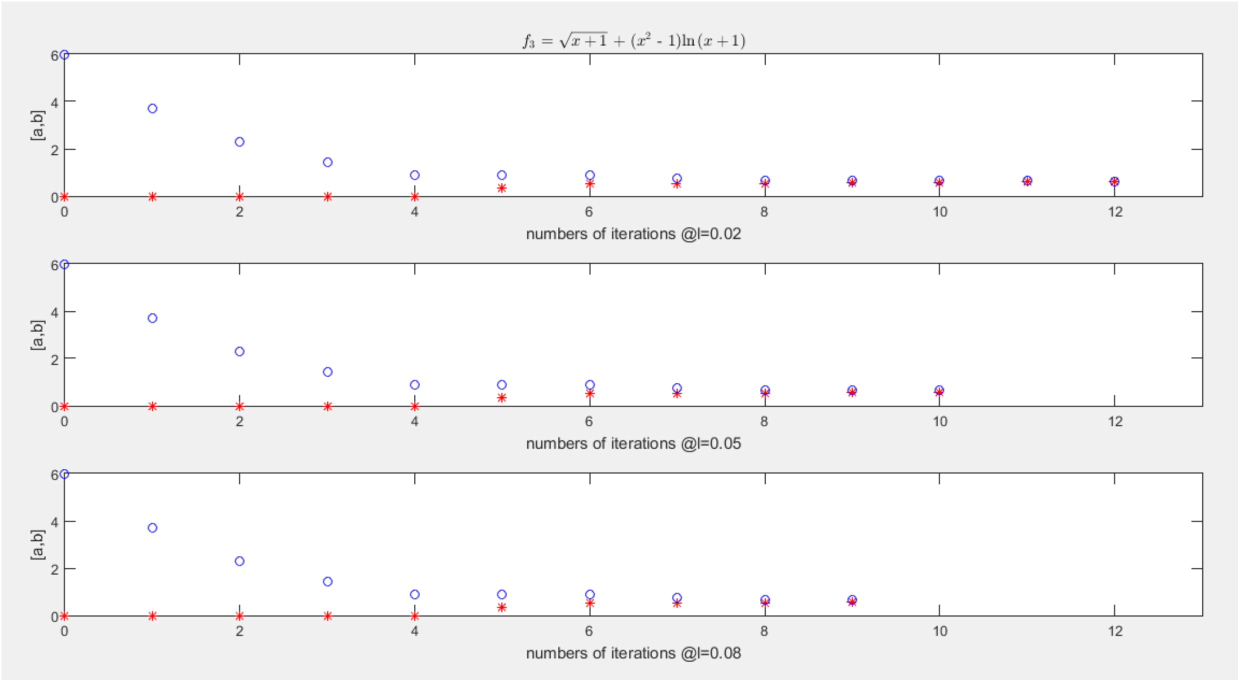
Η σύγκλιση των άκρων του διαστήματος, ανά βήμα επανάληψης για κάθε συνάρτηση:



Εικόνα 7: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_1 [Μέθοδος Χρυσού Τομέα]



Εικόνα 8: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_2 [Μέθοδος Χρυσού Τομέα]



Εικόνα 9: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_3 [Μέθοδος Χρυσού Τομέα]

Με κόκκινο * αποτυπώνεται το a_κ και με μπλε o το b_κ.

<u>Μέθοδος Fibonacci</u>: Μοιάζει σε μεγάλο βαθμό με την μέθοδο του χρυσού τομέα, ωστόσο τα υπο-διαστήματα κατά την k-οστή επανάληψη δεν συνδέονται με κάποια σταθερά, <u>αλλά</u> μεταβάλλονται, από επανάληψη σε επανάληψη, βασιζόμενα στην ακολουθία Fibonacci.

$$F_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F_{n-1} + F_{n-2}, & n \geq 2 \end{cases}$$

Οι βοηθητικές μεταβλητές c, d ισούνται με :

$$c_\kappa = a_\kappa + \left(1 - \frac{F_{n-1-k}}{F_{n-k}}\right) * (b_\kappa - a_\kappa) \;\kappa\alpha\iota\; d_\kappa = a_\kappa + \frac{F_{n-1-k}}{F_{n-k}} * (b_\kappa - a_\kappa)$$

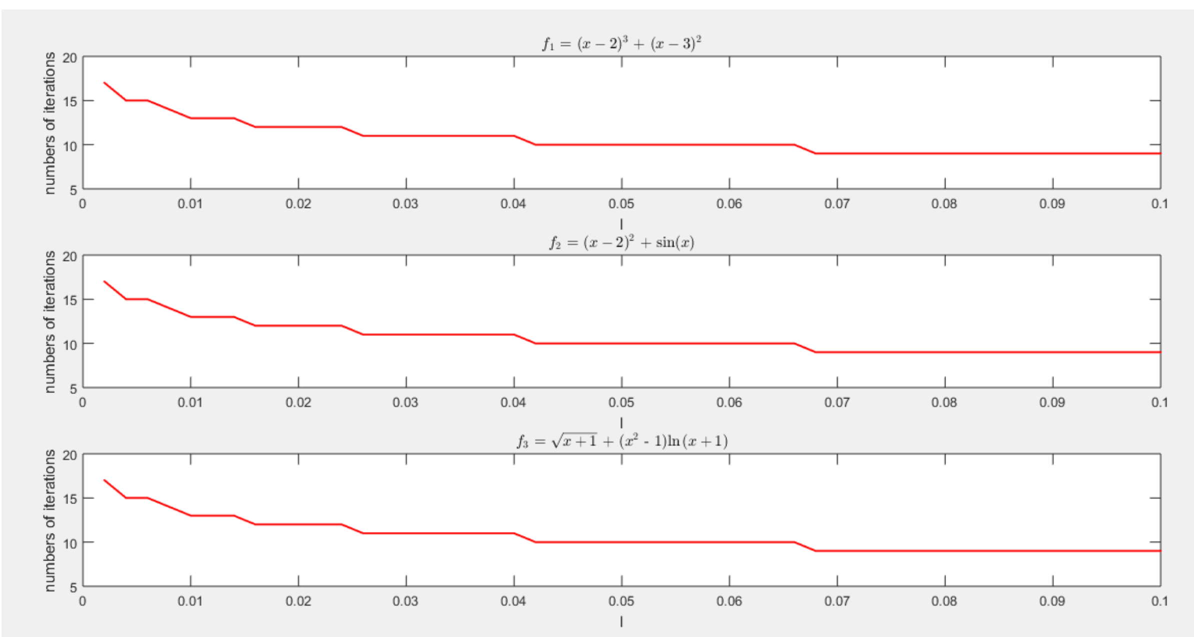
Ακόμα μία σημαντική διαφορά είναι ότι για αυτή την μέθοδο είναι αναγκαίο να γνωρίζουμε εκ των προτέρων των αριθμό n των επαναλήψεων. Για να υπολογιστεί, εκμεταλλευόμαστε το γεγονός ότι σε κάθε επανάληψη το εύρος αναζήτησης μειώνεται κατά $\frac{F_{n-1-k}}{F_{n-k}}$. Άρα,

$$F_n \leq \frac{b-a}{l} \leq F_{n+1} \,.$$

<u>Σημείωση για τον κώδικα</u>: Για τον υπολογισμό των αριθμών Fibonacci, δημιουργήθηκε το αρχείο `fib.m`. Για να μειωθεί ο χρόνος εκτέλεσης και για να μην γίνει χρήση πίνακα για την αποθήκευση, εκμεταλλεύομαι το γεγονός ότι η τιμή που ψάχνω εξαρτάται από τις δυο προηγούμενες, οι οποίες βρίσκονται στις μεταβλητές `old_value` και `pre_old_value`.

Στο <u>`fibonacci_mine.m`</u>, αρχικά υπολογίζεται ο αριθμός των επαναλήψεων n. Έπειτα, σύμφωνα με τις τιμές των $f(c)$ και $f(d)$ το διάστημα περιορίζεται. Για παράδειγμα, αν$f(c) \leq f(d)$, το a παραμένει ίδιο και γνωρίζοντας ότι η f είναι **αυστηρά σχεδόν-κυρτή** το b_{k+1} γίνεται ίσο με το d_k, αφού η τιμή που ψάχνουμε είναι σίγουρα στο $[\alpha, d)$. Το d_{k+1} εξισώνεται με το c_k και το c υπολογίζεται εκ νέου , για $k+1$. Αντίστοιχα για την άλλη περίπτωση.

Χρησιμοποιώντας το `fibonacci_mine_plot_for_different_lamda`, μελετάω τον αριθμό των κλήσεων της συνάρτησης για διαφορετικά l :



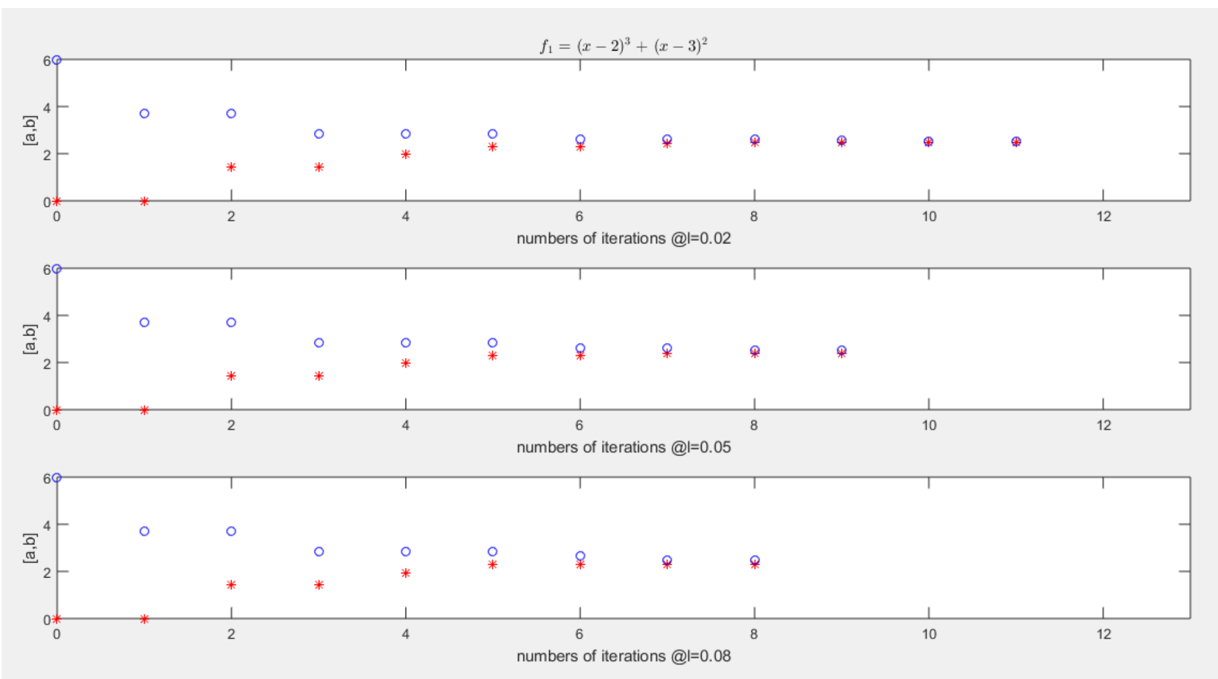
Εικόνα 10: Κλήσεις της αντικειμενικής συνάρτησης για διάφορα l [Μέθοδος Fibonacci]

Σημείωση: Για την ακολουθία Fibonacci, δεν ακολούθησα την σύμβαση του βιβλίου (ότι $F(0) = 1$), αλλά ότι $F(0) = 0$. Έτσι, για τον αριθμό των υπολογισμών της αντικειμενικής συνάρτησης ισχύει ότι $F_{n+2} > \frac{b_1 - a_1}{l}$, σχέση που επιβεβαιώνεται και από το παραπάνω διάγραμμα.

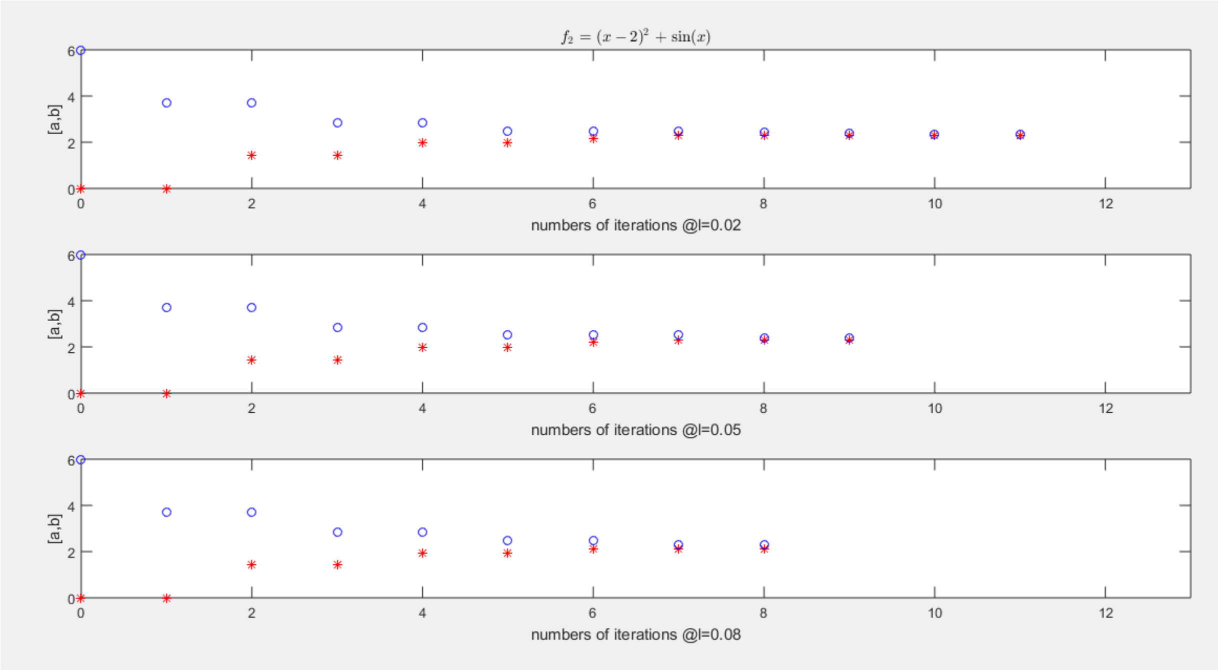
Πχ, για $l = 0.1 \Rightarrow n = 9$ και $F_{11} = 89 > \frac{b_1 - a_1}{l} = 60$

Πιο συγκεκριμένα, την πρώτη φορά που καλείται η `fibonacci_mine.m`, υπολογίζονται τα x_1, x_2 και στην εκτέλεση της εντολής `if f(x_1) > f(x_2)` έχουμε 2 κλήσεις τις f. Από την επόμενη, όμως, επανάληψη είτε το x_1 είτε το x_2είναι γνωστά και έτσι έχουμε 1 κλήση της f. Άρα, $2 + \underbrace{1 + 1 + \cdots + 1}_{\kappa - 1} - 1 = \kappa$, καθώς αρχικά $\kappa = 1$. Ωστόσο, το block των εντολών του `for k = 1:n-1`, θα εκτελεστεί έως ότου $\kappa = n - 1$

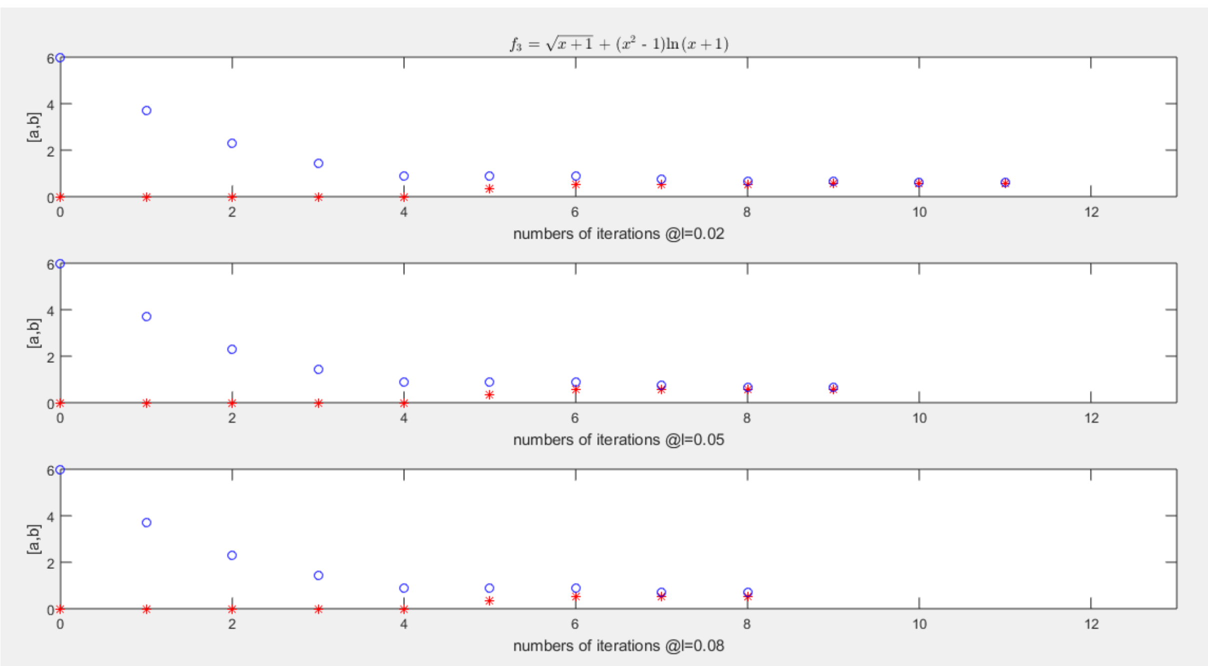
Η σύγκλιση των άκρων του διαστήματος, ανά βήμα επανάληψης για κάθε συνάρτηση:



Εικόνα 11: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_1 [Μέθοδος Fibonacci]



Εικόνα 12: Περιορισμός του $[a_\kappa, b_\kappa]$ *για 3 τιμές του* l *στην* f_2 *[Μέθοδος Fibonacci]*

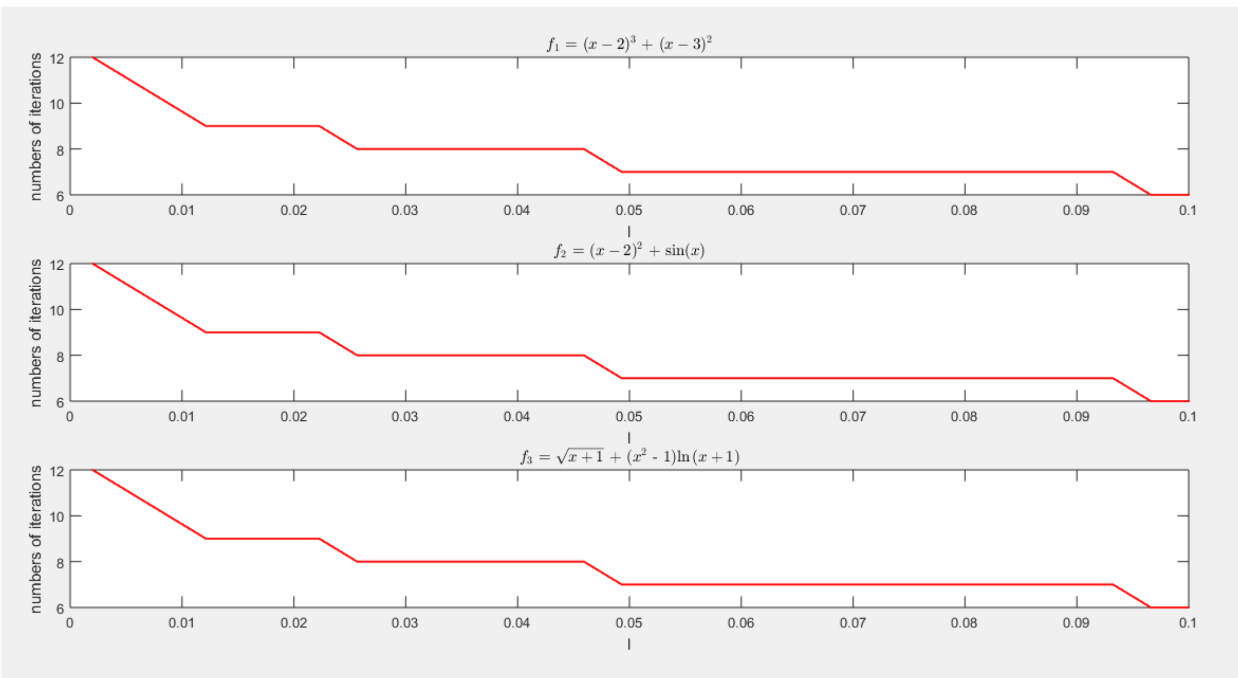


Εικόνα 13: Περιορισμός του $[a_\kappa, b_\kappa]$ *για 3 τιμές του* l *στην* f_3 *[Μέθοδος Fibonacci]*

Με κόκκινο * αποτυπώνεται το a_κ και με μπλε ο το b_κ.

<u>Μέθοδος Διχοτόμου</u> ***<u>(με την χρήση παραγώγου)</u>***: Σ' αυτήν την μέθοδο, εκμεταλλευόμαστε την κλίση της συνάρτησης, στο συγκεκριμένο σημείο x^*. Αν $\left.\frac{df}{dx}\right|_{x=x^*} = 0$, τότε το σημείο είναι το ελάχιστο. Για $\left.\frac{df}{dx}\right|_{x=x^*} > 0$ η συνάρτηση έχει θετική κλίση ("αυξάνει") και έτσι για $x > x^*$, ισχύει $f(x) > f(x^*)$ και το διάστημα θα ισούται με $[a, x^*)$. Τέλος, για $\left.\frac{df}{dx}\right|_{x=x^*} < 0$ η συνάρτηση έχει αρνητική κλίση ("φθίνει") και έτσι για $x < x^*$, ισχύει $f(x) > f(x^*)$ και το διάστημα θα ισούται με $(x^*, b]$.

Χρησιμοποιώντας το `bisection_der_plot_for_different_lamda`, μελετάω τον αριθμό των κλήσεων της συνάρτησης για διαφορετικά l :

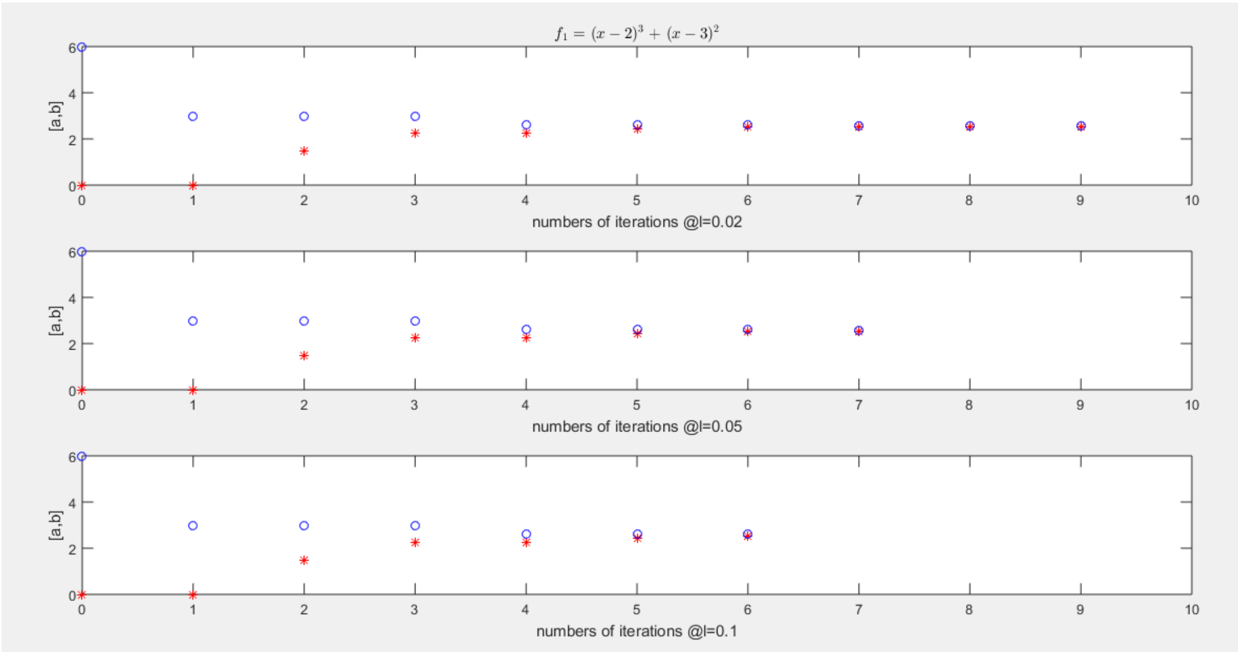


Εικόνα 14:Κλήσεις της αντικειμενικής συνάρτησης για διάφορα l [Μέθοδος Διχοτόμησης με Παράγωγο]

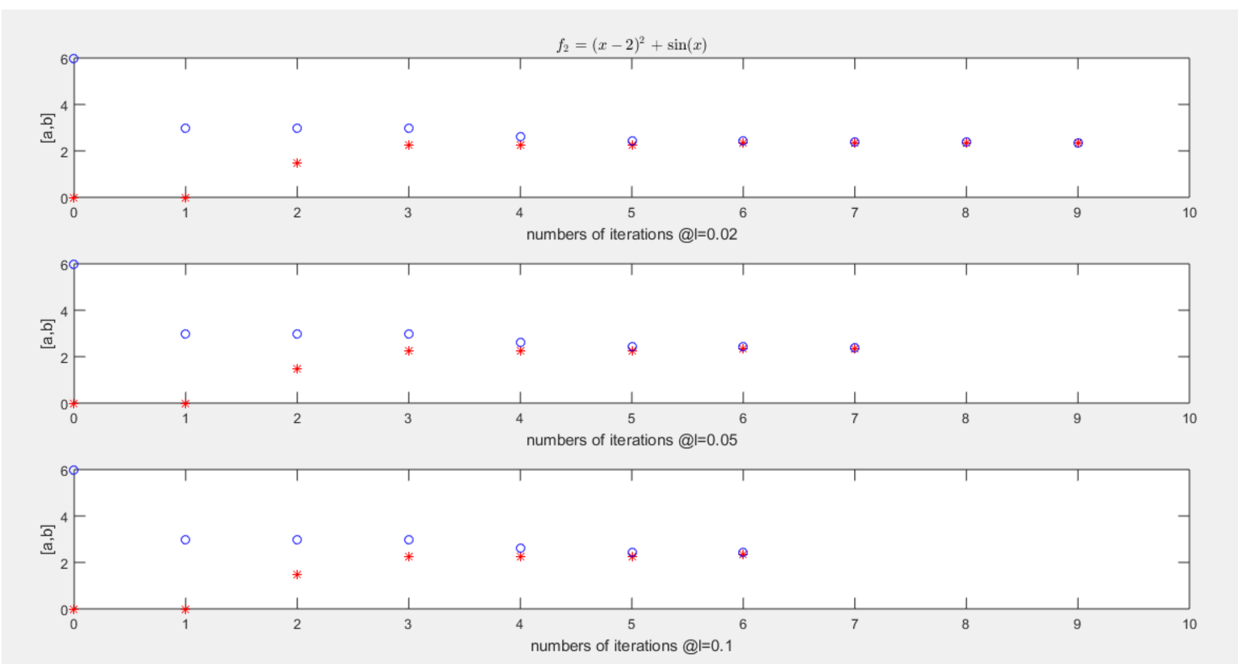
Γνωρίζουμε, ότι σε κάθε επανάληψη το διάστημα αναζήτησης μειώνεται κατά το ήμισυ. Έτσι, για την $i - οστή$ επανάληψη, θα ισχύει: $\Delta_i = \frac{\Delta_0}{2^i}$, όπου Δ_0 το αρχικό διάστημα. Για την τελική επανάληψη n και με εύρος αναζήτησης l, πρέπει $\Delta_n \le l \Rightarrow \frac{b_0 - a_0}{2^n} \le l \Rightarrow n \ge \log_2\left(\frac{b_0 - a_0}{l}\right)$

Πιο συγκεκριμένα, κάθε φορά που καλείται η `bisection_der.m`, υπολογίζεται το x_κ και στην εκτέλεση της εντολής `value = subs(df,`x_1`)` έχουμε 1 κλήση της f.
Άρα, $\underbrace{1 + 1 + \cdots + 1}_{\kappa} = \kappa$. Ωστόσο, το block των εντολών της `for k = 1:n`, θα εκτελεστεί n φορές.

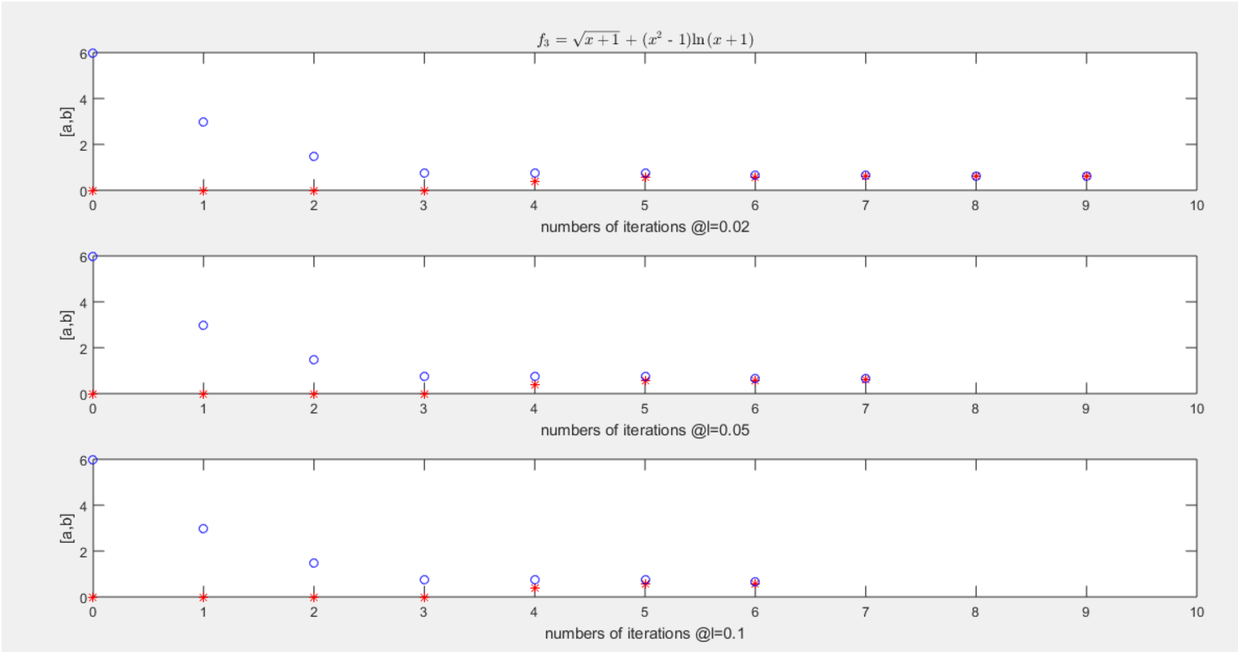
Η σύγκλιση των άκρων του διαστήματος, ανά βήμα επανάληψης για κάθε συνάρτηση:



Εικόνα 15: Περιορισμός του $[a_\kappa, b_\kappa]$ *για 3 τιμές του* l *στην* f_1 *[Μέθοδος Διχοτόμησης με Παράγωγο]*



Εικόνα 16: Περιορισμός του $[a_\kappa, b_\kappa]$ *για 3 τιμές του* l *στην* f_2 *[Μέθοδος Διχοτόμησης με Παράγωγο]*



Εικόνα 17: Περιορισμός του $[a_\kappa, b_\kappa]$ για 3 τιμές του l στην f_3 [Μέθοδος Διχοτόμησης με Παράγωγο]

Με κόκκινο * αποτυπώνεται το a_κ και με μπλε ο το b_κ.

Συμπεράσματα:

Τρέχοντας όλους τους αλγορίθμους για τα ίδια δεδομένα, παρατηρούμε:

n, αριθμός επαναλήψεων και κ, κλήσεις αντικειμενικής συνάρτησης

Μέθοδος	Τελικό εύρος αναζήτησης l	Αριθμός υπολογισμών της αντικειμενικής συνάρτησης
Διχοτόμησης $\kappa = 2 * (n-1)$	0.0021	32
	0.01	20
	0.1	12
Χρυσού Τομέα $\kappa = n$	0.0021	18
	0.01	15
	0.1	10

Fibonacci $\kappa = n - 1$	0.0021	17
	0.01	13
	0.1	9
Διχοτόμησης (με παράγωγο) $\kappa = n$	0.0021	12
	0.01	10
	0.1	6

Ο πιο αποδοτικός αλγόριθμος για την εύρεση του ελαχίστου, σε μια κυρτή συνάρτηση μιας μεταβλητής, είναι ο αλγόριθμος της διχοτόμησης με χρήση παραγώγου. Αν ο υπολογισμός της παραγώγου είναι απαγορευτικός, τότε ακολουθεί ο αλγόριθμος Fibonacci, που αν και πιο περίπλοκος εκμεταλλεύεται είτε το c είτε το d, που είναι γνωστά στην επόμενη επανάληψη, και μειώνει σημαντικά τις πράξεις. Τέλος, ακολουθεί ο αλγόριθμος του Χρυσού τομέα.

Τέλος, ένα μειονέκτημα των αλγορίθμων Fibonacci και Διχοτόμησης με χρήση παραγώγου, μπορεί να θεωρηθεί το γεγονός του υπολογισμού του n, δηλαδή, τον αριθμό των επαναλήψεων.