Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής

Τεχνικές Βελτιστοποίησης - 1η Εργασία

Θεόδωρος Παπαφωτίου
ΑΕΜ: 9708
papafotit@ece.auth.gr

18 Νοεμβρίου 2021

Περιεχόμενα

1 Περιγραφή 1ης Εργασίας **2**
- 1.1 Μέθοδοι . . . 2
- 1.2 Υλοποίηση . . . 2
- 1.3 Μελέτη . . . 2

2 Μελέτη Μεθόδων **4**
- 2.1 Μέθοδος Διχοτόμου - Χωρίς χρήση παραγώγων . . . 4
  - 2.1.1 Σύντομη περιγραφή της μεθόδου . . . 4
  - 2.1.2 Γραφήματα . . . 5
- 2.2 Μέθοδος Χρυσού Τομέα . . . 12
  - 2.2.1 Σύντομη περιγραφή της μεθόδου . . . 12
  - 2.2.2 Γραφήματα . . . 12
- 2.3 Μέθοδος Fibonacci . . . 18
  - 2.3.1 Σύντομη περιγραφή της μεθόδου . . . 18
  - 2.3.2 Γραφήματα . . . 19
- 2.4 Μέθοδος Διχοτόμου - Με χρήση παραγώγων . . . 25
  - 2.4.1 Σύντομη περιγραφή της μεθόδου . . . 25
  - 2.4.2 Γραφήματα . . . 25
- 2.5 Γενικά Συμπεράσματα . . . 32

3 Σύγκριση Μεθόδων **33**

Κεφάλαιο 1

Περιγραφή 1ης Εργασίας

Μια αναφορά [**;**]

Στόχος της παρούσας εργασίας είναι η υλοποίηση και μελέτη μεθόδων για την **εύρεση ελαχίστου** μια δωσμένης συνάρτησης σε συγκεκριμένο εύρος [A, B].

1.1 Μέθοδοι

Οι μέθοδοι που καλούμαστε να υλοποιήσουμε και να μελετήσουμε είναι οι παρακάτω:

- Μέθοδος Διχοτόμου - Χωρίς χρήση παραγώγων
- Μέθοδος Χρυσού Τομέα
- Μέθοδος Fibonacci
- Μέθοδος Διχοτόμου - Με χρήση παραγώγων

Σε όλες τις παραπάνω μεθόδους ξεκινάμε από ένα αρχικό διάστημα $[a_0, b_0]$ μέσα στο οποίο βρίσκεται το ελάχιστο x^* της $f(x)$ Με τη χρησιμοποίηση ενός ακολουθιακού αλγόριθμου καταλήγουμε σε ένα διάστημα $[a_k,\ b_k]$ με προδιαγεγραμμένη ακρίβεια l, δηλαδή $b_k - a_k \leq l$.

1.2 Υλοποίηση

Η υλοποίηση των παραπάνω μεθόδων πραγματοποιήθηκε με τη χρήση του λογισμικού **MATLAB** (v.2019b). Για κάθε μέθοδο έχει υλοποιηθεί η αντίστοιχη συνάρτηση (function), καθώς και ένα αρχείο (main) που καλεί τη συνάρτηση αυτή. Όλα τα αρχεία επισυνάπτονται με την παρούσα τεχνική αναφορά στο αποθετήριο της πλατφόρμας eLearning.

1.3 Μελέτη

Οι συναρτήσεις προς ελαχιστοποίηση είναι οι παρακάτω:

- $f_1(x) = (x-3)^2 + (\sin(x+3))^2$

- $f_2(x) = (x-1) \cdot \cos\left(\frac{x}{2}\right) + x^2$
- $f_3(x) = (x+2)^2 + e^{(x-2)} \cdot \sin(x+3)$

Για κάθε μία συνάρτηση, καλούμαστε στο διάστημα $[a_0, b_0] = [-4, 4]$ να αναζητήσουμε το ελάχιστό τους, χρησιμοποιώντας της παραπάνω μεθόδους. Τη συγκεκριμένη διαδικασία την επαναλαμβάνουμε για διαφορετικές τιμές του l, ενώ στην περίπτωση της *Μεθόδου Διχοτόμου χωρίς παραγώγους* μεταβάλλουμε και το e.

Έπειτα από την αναλυτική μελέτη που θα πραγματοποιηθεί στα επόμενα κεφάλαια για κάθε μέθοδο με τη χρήση διαγραμμάτων και αποτελεσμάτων, η αναφορά θα ολοκληρωθεί με έναν συγκριτικό σχολιασμό πάνω στην αποδοτικότητα των υπό μελέτη μεθόδων για τις 3 συναρτήσεις.

Κεφάλαιο 2

Μελέτη Μεθόδων

2.1 Μέθοδος Διχοτόμου - Χωρίς χρήση παραγώγων

2.1.1 Σύντομη περιγραφή της μεθόδου

Στη μέθοδο διχοτόμου, χωρίς χρήση παραγώγων, ορίζουμε τα x_{1_k} και x_{2_k} ως εξής:

$$
\begin{aligned}
x_{1_k} &= \frac{a_k + b_k}{2} - e \\
x_{2_k} &= \frac{a_k + b_k}{2} + e
\end{aligned}
\tag{2.1}
$$

όπου k η επανάληψη στην οποία βρισκόμαστε κάθε στιγμή. Στη συνέχεια ελέγχουμε στη συνάρτηση f, που μας έχει δοθεί, τις τιμές $f(x_{1_k})$ και $f(x_{2_k})$ και μετακινούμε τα αρχικά όρια $[a_0, b_0]$ προς το σημείο ελαχιστοποίησης της συνάρτησης με τον παρακάτω αλγόριθμο:

Algorithm 1 Set a_k, b_k in Bisection Method without derivatives

```
k ← 0
while (b_k − a_k) > l do
    if f(x_{1_k}) < f(x_{2_k}) then
        b_{k+1} ← x_{2_k}
    else if f(x_{1_k}) > f(x_{2_k}) then
        a_{k+1} ← x_{1_k}
    else
        a_{k+1} ← x_{1_k}
        b_{k+1} ← x_{2_k}
    end if
    x_{1_{k+1}} ← (a_{k+1} + b_{k+1})/2 − e
    x_{2_{k+1}} ← (a_{k+1} + b_{k+1})/2 + e
    k ← k + 1
end while
```

Μόλις ο αλγόριθμος βγει από το βρόχο, γνωρίζουμε ότι έχει βρεθεί προσεγγιστικά το ελάχιστο της συνάρτησης και είναι ίσο με $min = \frac{a_n+b_n}{2}$, όπου n είναι ο συνολικός αριθμός των επαναλήψεων του βρόχου.

Για τη συγκεκριμένη μέθοδο απαιτούμε $l < 2e$, ώστε να μπορεί να τερματίζει ο κώδικας. Ειδάλλως, η διαφορά των x_{1_k} και x_{2_k} θα είναι πάντα μεγαλύτερη του l και ο βρόχος δεν θα τερματίσει ποτέ.

2.1.2 Γραφήματα

Παρακάτω παραθέτουμε τα γραφήματα και τους πίνακες που εξήχθησαν από το πρόγραμμα στο MATLAB για τη συγκεκριμένη μέθοδο, τα οποία χρησιμοποιούνται για την εξαγωγή των συμπερασμάτων. Θεωρούμε ότι με l θέτουμε την ακρίβεια και με n_{calc} τους υπολογισμούς της αντικειμενικής συνάρτησης.

Σύγκλιση στο ελάχιστο

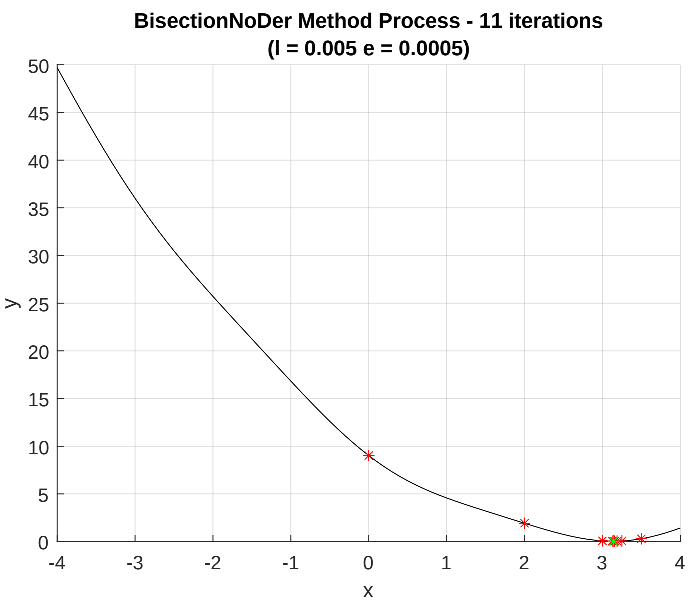

Figure 2.1: [F1] Convergence to minimum

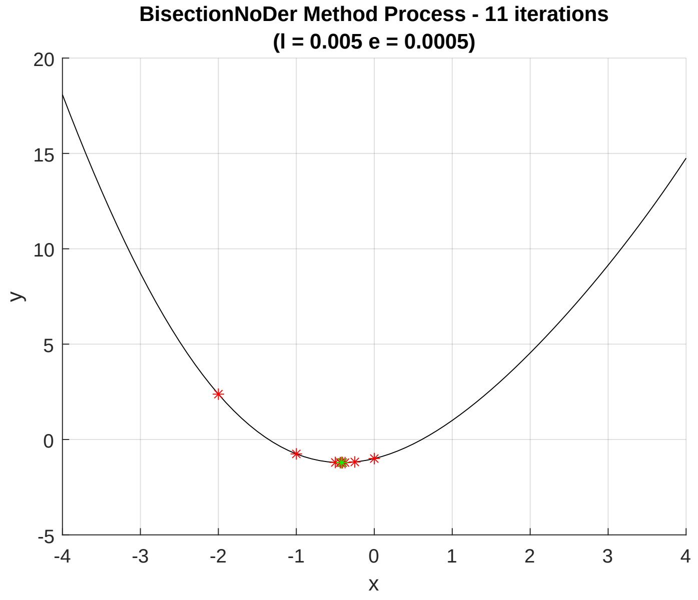

Figure 2.2: [F2] Convergence to minimum

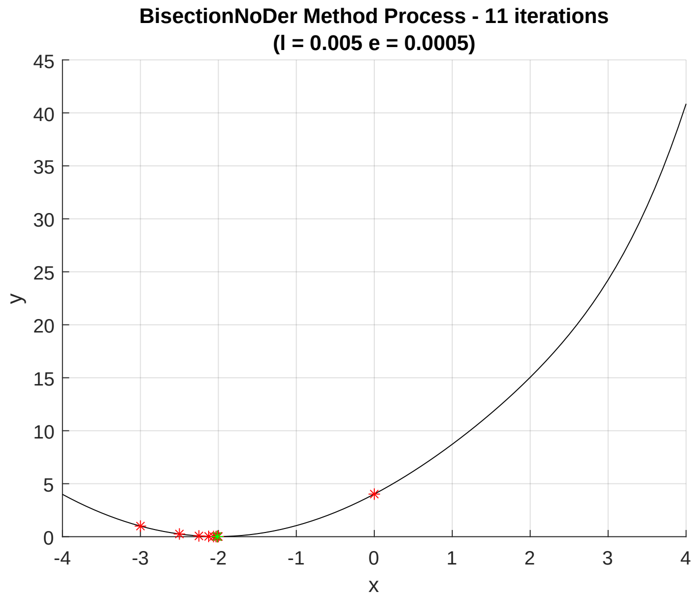

Figure 2.3: [F3] Convergence to minimum

Μεταβολή του l διατηρώντας σταθερό το e

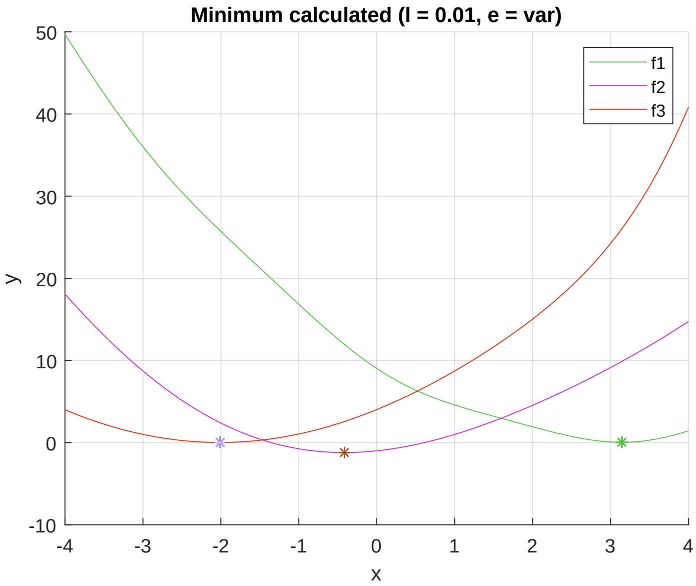

Figure 2.4: [F1, F2, F3] Variation of l while $e = 0.001$

l	e	f1		f2		f3	
		x_min	**N**	**x_min**	**N**	**x_min**	**N**
0.005	0.001	3.14082	13	-0.416890	13	-2.01219	13
0.01	0.001	3.14375	11	-0.417860	11	-2.01122	11
0.015	0.001	3.14375	11	-0.417860	11	-2.01122	11
0.02	0.001	3.14765	10	-0.413960	10	-2.00731	10
0.025	0.001	3.14765	10	-0.413960	10	-2.00731	10
0.03	0.001	3.14765	10	-0.413960	10	-2.00731	10
0.035	0.001	3.13984	9	-0.421770	9	-2.01512	9
0.04	0.001	3.13984	9	-0.421770	9	-2.01512	9

Table 2.1: Variation of l

Μεταβολή του e διατηρώντας σταθερό το l

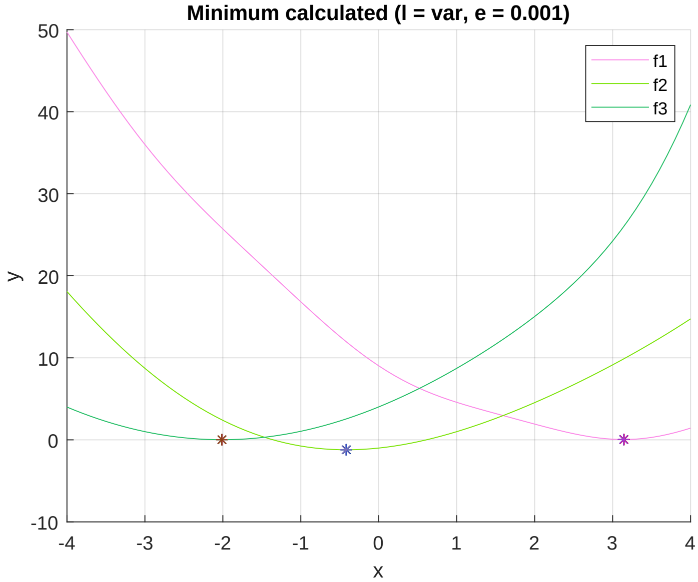

Figure 2.5: [F1, F2, F3] Variation of e while $l = 0.01$

l	e	f1		f2		f3	
		x_min	**N**	**x_min**	**N**	**x_min**	**N**
0.01	**0.0005**	3.14414	11	-0.417920	11	-2.01147	11
0.01	**0.001**	3.14375	11	-0.417860	11	-2.01122	11
0.01	**0.0015**	3.14140	12	-0.415860	12	-2.01292	12
0.01	**0.002**	3.14101	12	-0.415810	12	-2.01267	12
0.01	**0.0025**	3.14061	12	-0.415760	12	-2.01241	12
0.01	**0.003**	3.14022	12	-0.415700	12	-2.01216	12
0.01	**0.0035**	3.14080	13	-0.416630	13	-2.01289	13
0.01	**0.004**	3.14041	13	-0.416580	13	-2.01263	13

Table 2.2: Variation of e

Σύγκλιση του διαστήματος $[a_k, b_k]$ στο ελάχιστο

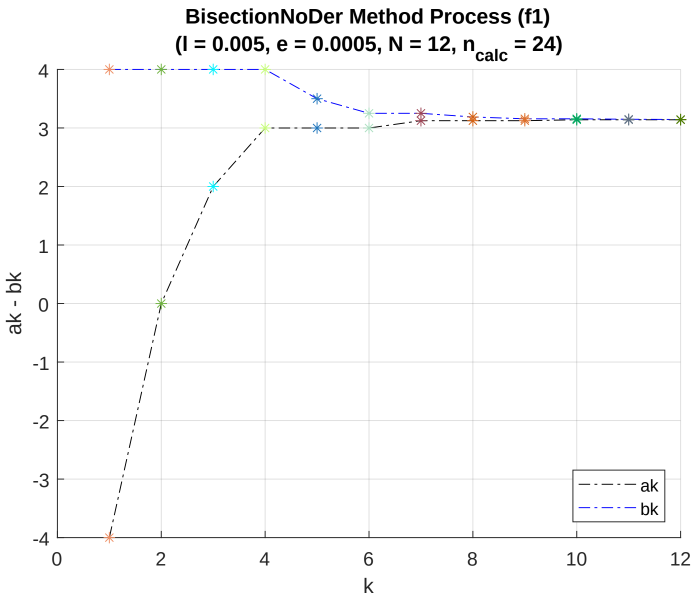

Figure 2.6: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

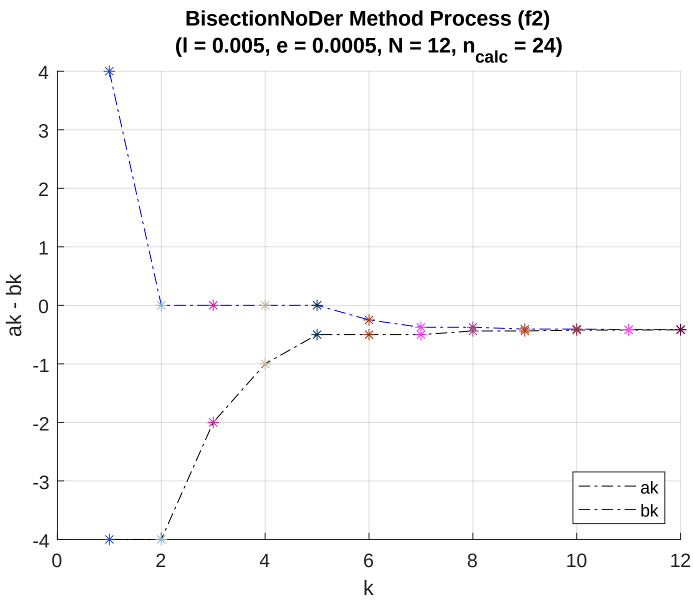

Figure 2.7: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

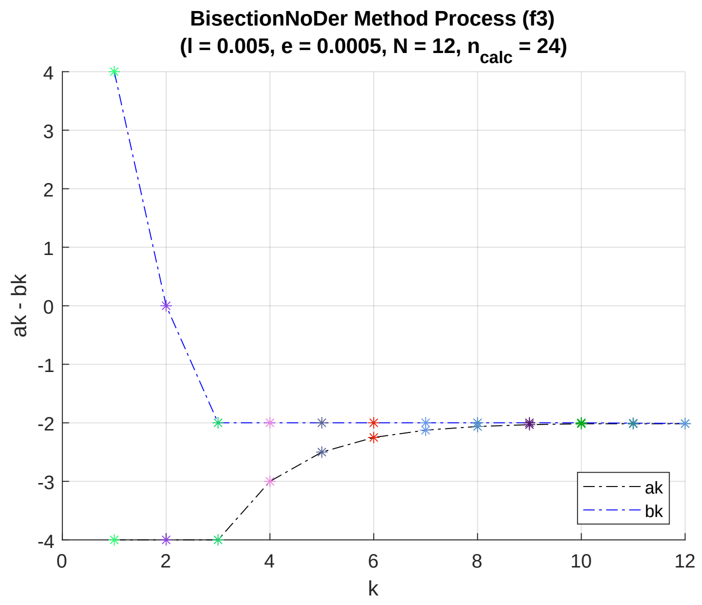

Figure 2.8: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

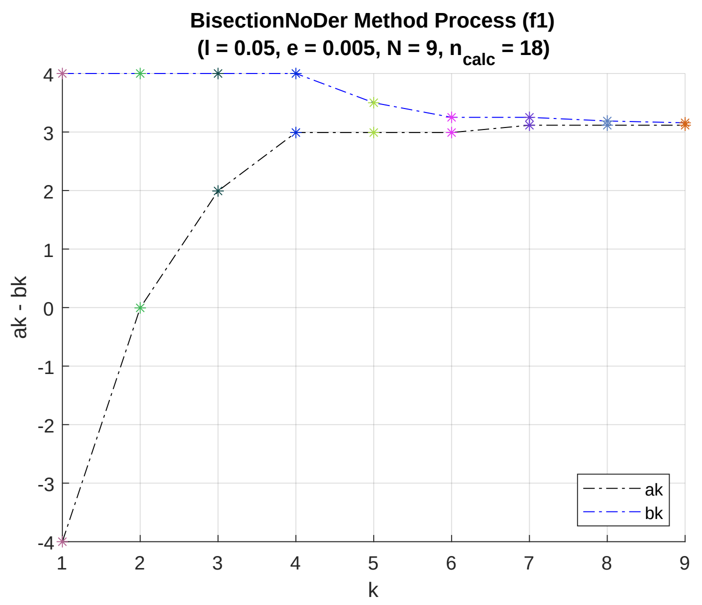

Figure 2.9: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

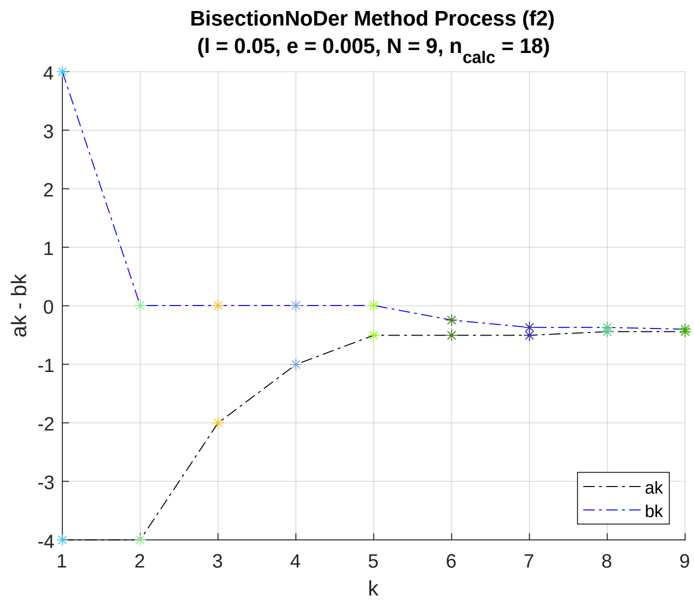

Figure 2.10: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

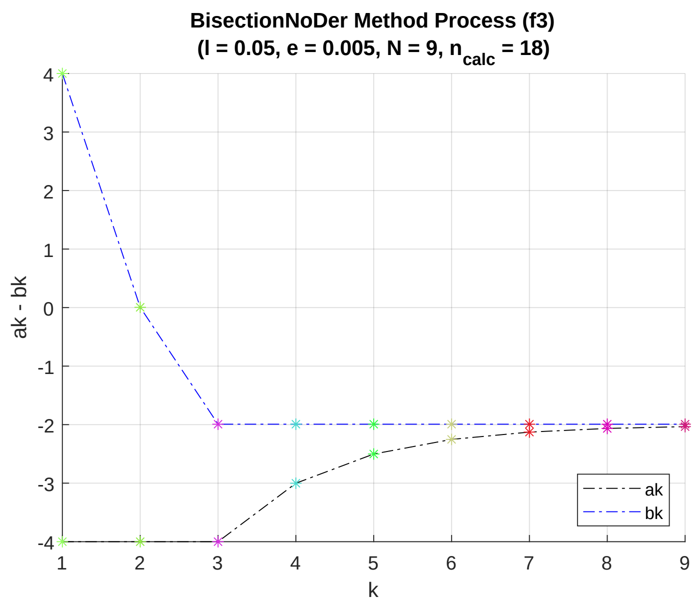

Figure 2.11: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

2.2 Μέθοδος Χρυσού Τομέα

2.2.1 Σύντομη περιγραφή της μεθόδου

Στη μέθοδο χρυσού τομέα ορίζουμε τα x_{1_k} και x_{2_k} ως εξής:

$$
\begin{aligned}
x_{1_k} &= a_k + (1-\gamma)\cdot(b_k - a_k) \\
x_{2_k} &= a_k + \gamma\cdot(b_k - a_k)
\end{aligned}
\tag{2.2}
$$

όπου k η επανάληψη στην οποία βρισκόμαστε κάθε στιγμή και γ ένας παράγοντας ίσος με 0.618. Φυσικά μπορούμε να χρησιμοποιήσουμε και διαφορετικό λόγο, ωστόσο αυτός είναι ο βέλτιστος, έπειτα από την απαραίτητη μελέτη που έχει πραγματοποιηθεί. Στη συνέχεια ελέγχουμε στη συνάρτηση f, που μας έχει δοθεί, τις τιμές $f(x_{1_k})$ και $f(x_{2_k})$ και μετακινούμε τα αρχικά όρια $[a_0, b_0]$ προς το σημείο ελαχιστοποίησης της συνάρτησης με τον παρακάτω αλγόριθμο:

Algorithm 2 Set a_k, b_k in Golden Section method

$k \leftarrow 0$
while $(b_k - a_k) > l$ **do**
  if $f(x_{1_k}) < f(x_{2_k})$ **then**
    $b_{k+1} \leftarrow x_{2_k}$
  else if $f(x_{1_k}) > f(x_{2_k})$ **then**
    $a_{k+1} \leftarrow x_{1_k}$
  else
    $a_{k+1} \leftarrow x_{1_k}$
    $b_{k+1} \leftarrow x_{2_k}$
  end if
  $x_{1_{k+1}} \leftarrow a_{k+1} + (1-\gamma)\cdot(b_{k+1} - a_{k+1})$
  $x_{2_{k+1}} \leftarrow a_{k+1} + \gamma\cdot(b_{k+1} - a_{k+1})$
  $k \leftarrow k+1$
end while

Μόλις ο αλγόριθμος βγει από το βρόχο, γνωρίζουμε ότι έχει βρεθεί προσεγγιστικά το ελάχιστο της συνάρτησης και είναι ίσο με $min = \frac{a_n+b_n}{2}$, όπου n είναι ο συνολικός αριθμός των επαναλήψεων του βρόχου.

2.2.2 Γραφήματα

Παρακάτω παραθέτουμε τα γραφήματα και τους πίνακες που εξήχθησαν από το πρόγραμμα στο MATLAB για τη συγκεκριμένη μέθοδο, τα οποία χρησιμοποιούνται για την εξαγωγή των συμπερασμάτων. Θεωρούμε ότι με l θέτουμε την ακρίβεια και με n_{calc} τους υπολογισμούς της αντικειμενικής συνάρτησης.

Σύγκλιση στο ελάχιστο

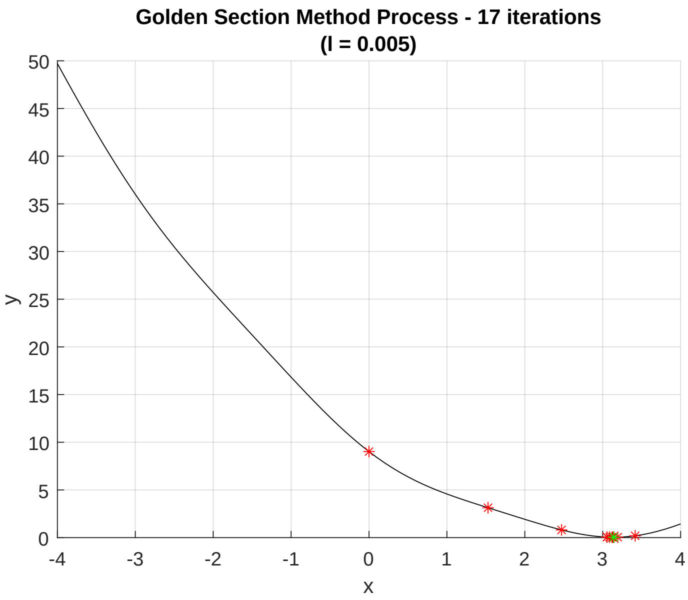

Figure 2.12: [F1] Convergence to minimum

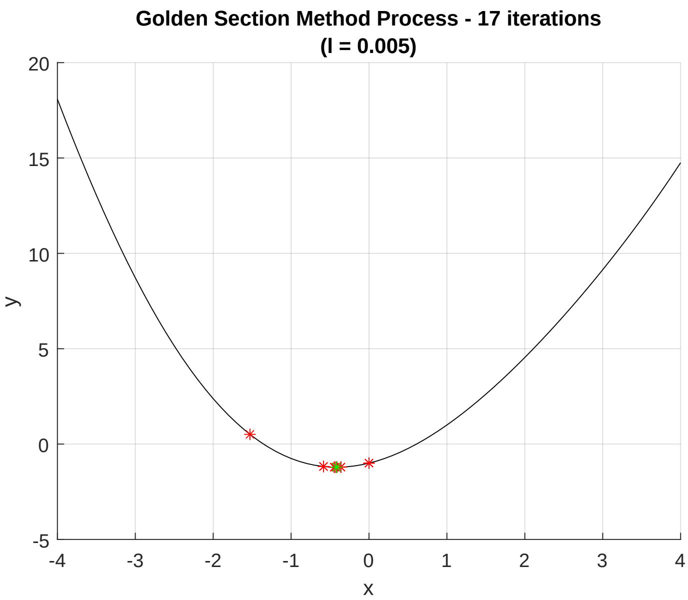

Figure 2.13: [F2] Convergence to minimum

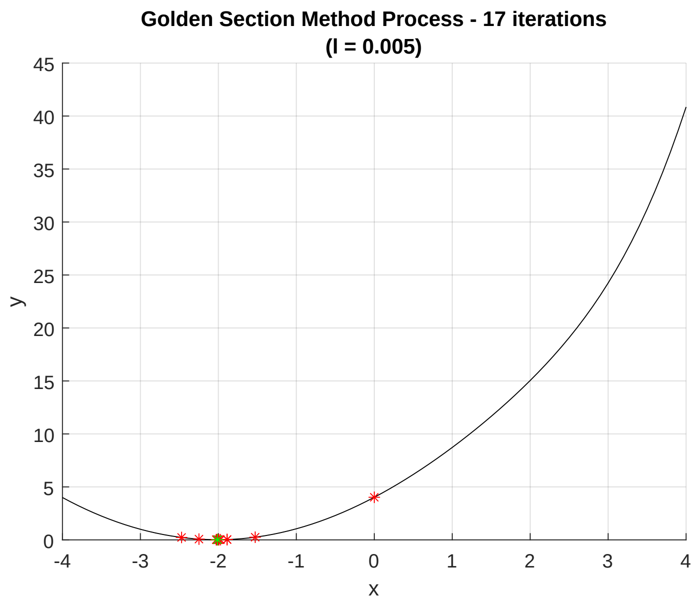

Figure 2.14: [F3] Convergence to minimum

Μεταβολή του l

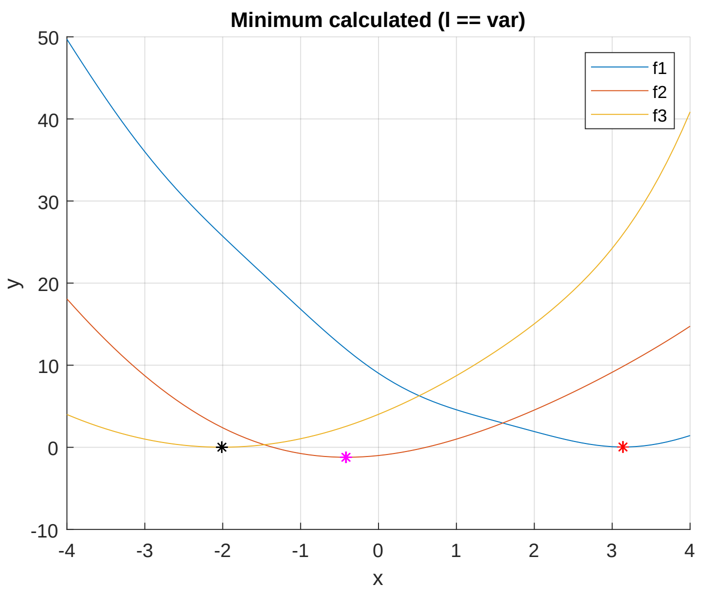

Figure 2.15: [F1, F2, F3] Variation of l

l	e	f1		f2		f3	
		x_min	**N**	**x_min**	**N**	**x_min**	**N**
0.005	0.001	3.14103	17	-0.416330	17	-2.01182	17
0.01	0.001	3.13810	15	-0.413400	15	-2.01113	15
0.015	0.001	3.13810	15	-0.413400	15	-2.01113	15
0.02	0.001	3.14103	14	-0.416330	14	-2.01406	14
0.025	0.001	3.13629	13	-0.421070	13	-2.00932	13
0.03	0.001	3.13629	13	-0.421070	13	-2.00932	13
0.035	0.001	3.13629	13	-0.421070	13	-2.00932	13
0.04	0.001	3.13629	13	-0.421070	13	-2.00932	13

Table 2.3: Variation of l

Σύγκλιση του διαστήματος $[a_k, b_k]$ στο ελάχιστο

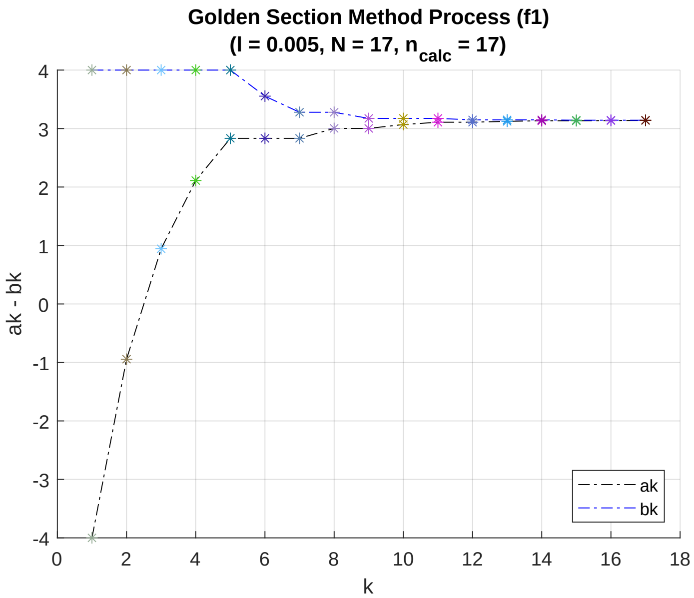

Figure 2.16: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

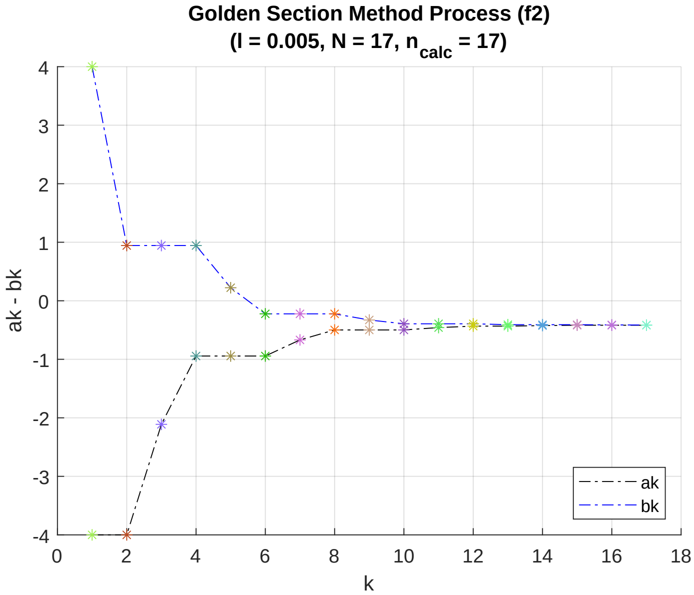

Figure 2.17: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

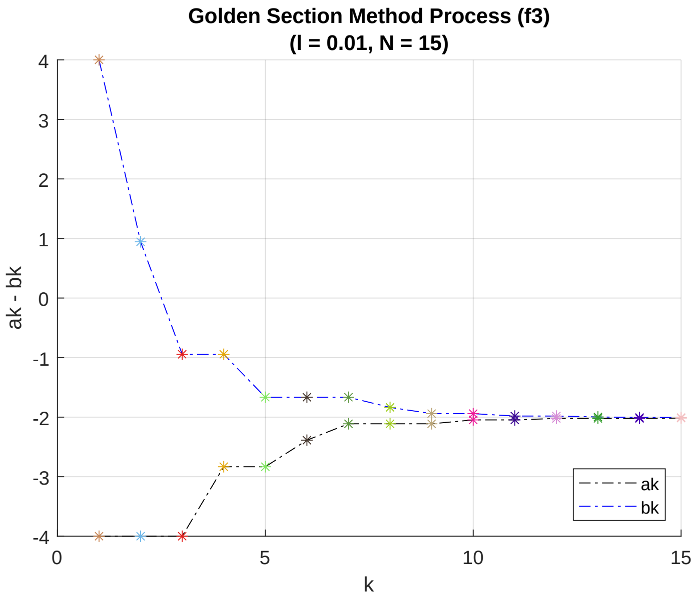

Figure 2.18: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

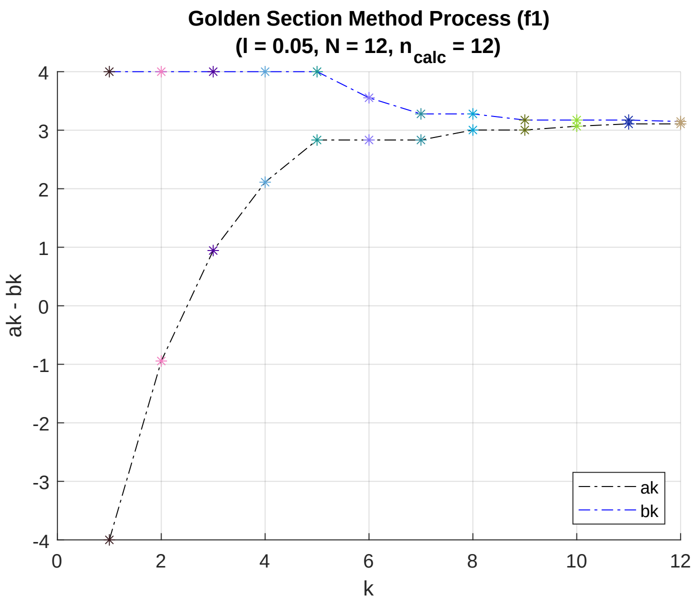

Figure 2.19: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

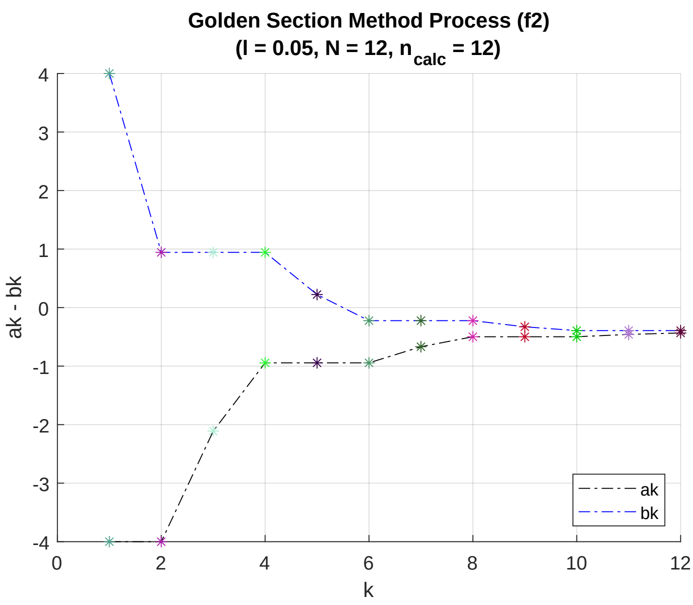

Figure 2.20: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

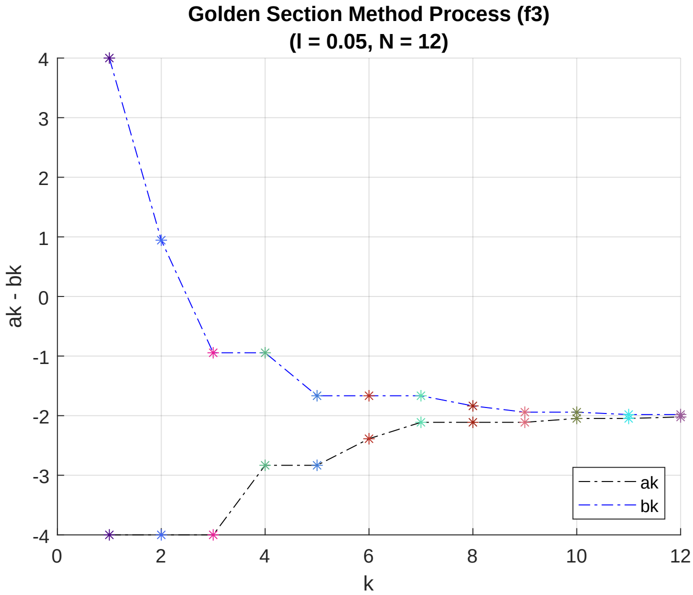

Figure 2.21: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

2.3 Μέθοδος Fibonacci

2.3.1 Σύντομη περιγραφή της μεθόδου

Στη μέθοδο Fibonacci ορίζουμε τα x_{1_k} και x_{2_k} ως εξής:

$$x_{1_k} = a_k + \frac{fibonacci(n-k-1)}{fibonacci(n-k+1)} \cdot (b_k - a_k)$$
$$x_{2_k} = a_k + \frac{fibonacci(n-k)}{fibonacci(n-k+1)} \cdot (b_k - a_k) \tag{2.3}$$

όπου k η επανάληψη στην οποία βρισκόμαστε κάθε στιγμή. Στη συνέχεια ελέγχουμε στη συνάρτηση f, που μας έχει δοθεί, τις τιμές $f(x_{1_k})$ και $f(x_{2_k})$ και μετακινούμε τα αρχικά όρια $[a_0, b_0]$ προς το σημείο ελαχιστοποίησης της συνάρτησης με τον παρακάτω αλγόριθμο:

Μόλις ο αλγόριθμος βγει από το βρόχο, γνωρίζουμε ότι έχει βρεθεί προσεγγιστικά το ελάχιστο της συνάρτησης και είναι ίσο με $min = \frac{a_j+b_j}{2}$, όπου j είναι ο συνολικός αριθμός των επαναλήψεων του βρόχου. Να σημειώσουμε εδώ, ότι το n που εμφανίζεται στον παραπάνω αλγόριθμο 3 αποτελεί τον δείκτη της πρώτης τιμής Φιβοναςςι που είναι μεγαλύτερη από το λόγο $\frac{b_0-a_0}{l}$, δηλαδή:

$$F_n > \frac{b_0 - a_0}{l} \implies n = \text{ceil}\left(\text{fibonacci}^{-1}\left(\frac{b_0 - a_0}{l}\right)\right) \tag{2.4}$$

Παρατηρούμε ότι η μέθοδος Fibonacci φέρει πολλά κοινά με τη μέθοδο Χρυσού Τομέα. Αυτό συμβαίνει επειδή πρακτικά αυτό που αλλάζει είναι ο παράγοντας γ, ο οποίος πλέον

Algorithm 3 Set a_k, b_k in Fibonacci method

$k \leftarrow 0$
while $(b_k - a_k) > l$ **do**
  if $f(x_{1_k}) < f(x_{2_k})$ **then**
    $b_{k+1} \leftarrow x_{2_k}$
  else if $f(x_{1_k}) > f(x_{2_k})$ **then**
    $a_{k+1} \leftarrow x_{1_k}$
  else
    $a_{k+1} \leftarrow x_{1_k}$
    $b_{k+1} \leftarrow x_{2_k}$
  end if
  $x_{1_{k+1}} \leftarrow a_{k+1} + \frac{fibonacci(n-(k+1)-1)}{fibonacci(n-(k+1)+1)} \cdot (b_{k+1} - a_{k+1})$
  $x_{2_{k+1}} \leftarrow a_{k+1} + \frac{fibonacci(n-(k+1))}{fibonacci(n-(k+1)+1)} \cdot (b_{k+1} - a_{k+1})$
  $k \leftarrow k + 1$
end while

αντικαθίσταται από τον λόγο $\frac{fibonacci(n-k)}{fibonacci(n-k+1)}$. *Για το λόγο αυτό αξίζει να συμειώσουμε ότι το* $\frac{1}{fibonacci(n)}$ *συγκλίνει στο* $\gamma = 0.618$, *που είναι και η βέλτιστη τη τιμή που λαμβάνει το* γ *στη μέθοδο του Χρυσού Τομέα.*

2.3.2 Γραφήματα

Παρακάτω παραθέτουμε τα γραφήματα και τους πίνακες που εξήχθησαν από το πρόγραμμα στο MATLAB για τη συγκεκριμένη μέθοδο, τα οποία χρησιμοποιούνται για την εξαγωγή των συμπερασμάτων. Θεωρούμε ότι με l θέτουμε την ακρίβεια και με n_{calc} τους υπολογισμούς της αντικειμενικής συνάρτησης.

Σύγκλιση στο ελάχιστο

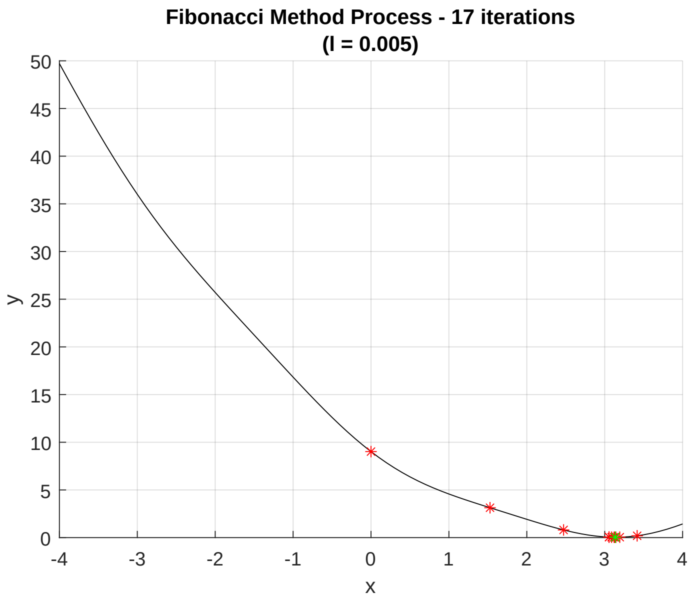

Figure 2.22: [F1] Convergence to minimum

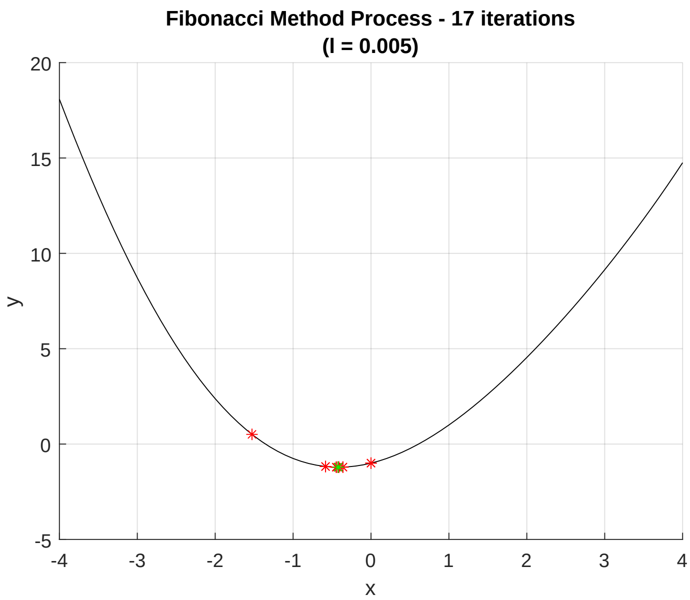

Figure 2.23: [F2] Convergence to minimum

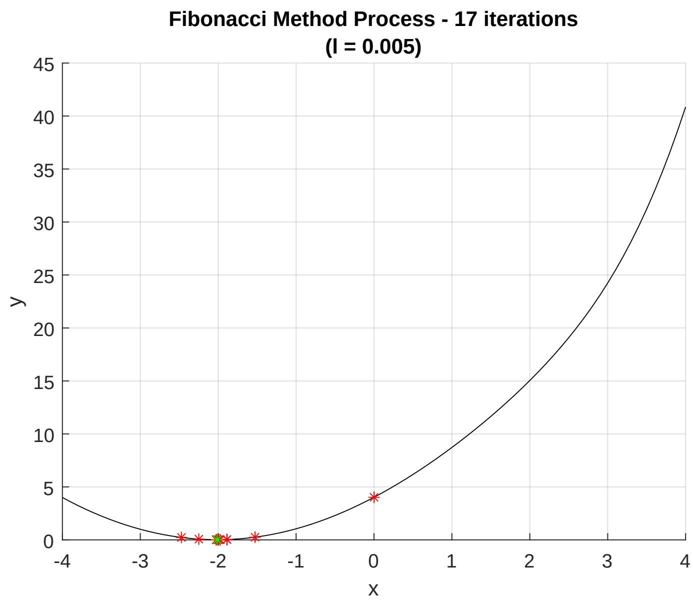

Figure 2.24: [F3] Convergence to minimum

Μεταβολή του l

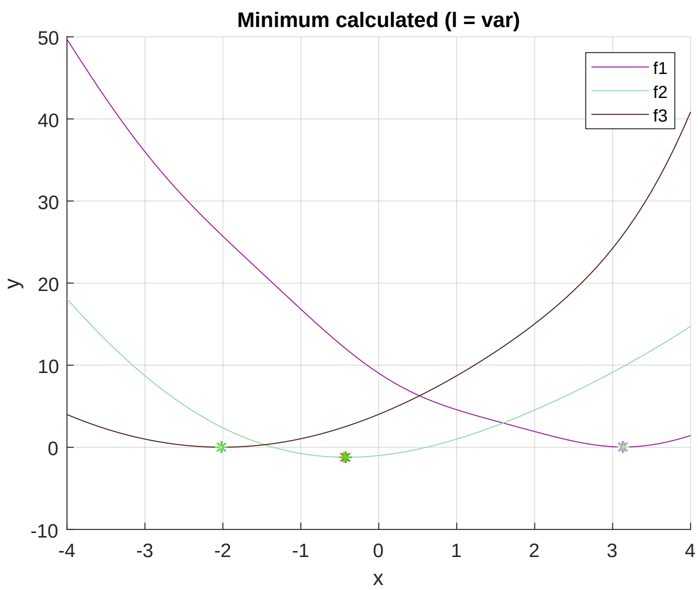

Figure 2.25: [F1, F2, F3] Variation of l

l	e	f1		f2		f3	
		x_min	N	x_min	N	x_min	N
0.005	0.001	3.13932	17	-0.414860	17	-2.01238	17
0.01	0.001	3.14083	15	-0.417430	15	-2.01418	15
0.015	0.001	3.13443	14	-0.419670	14	-2.00656	14
0.02	0.001	3.13443	14	-0.419670	14	-2.00656	14
0.025	0.001	3.15119	13	-0.413790	13	-2.00531	13
0.03	0.001	3.15119	13	-0.413790	13	-2.00531	13
0.035	0.001	3.14163	12	-0.429180	12	-2.00858	12
0.04	0.001	3.14163	12	-0.429180	12	-2.00858	12

Table 2.4: Variation of l

Σύγκλιση του διαστήματος $[a_k, b_k]$ στο ελάχιστο

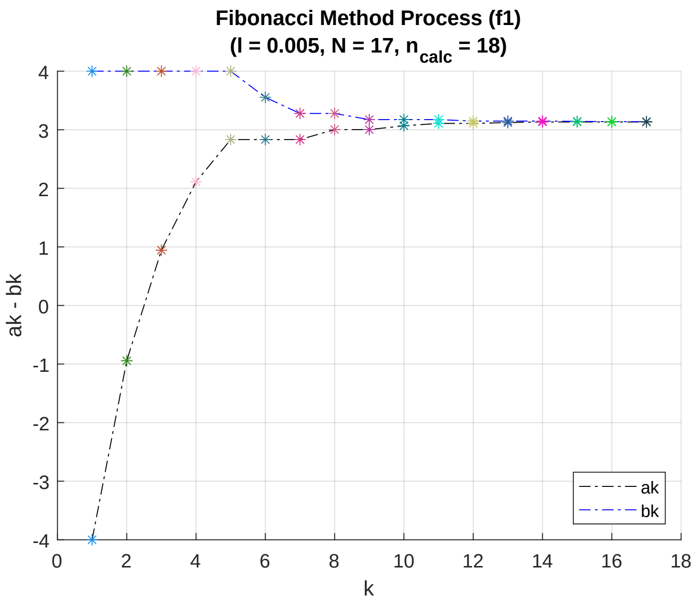

Figure 2.26: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

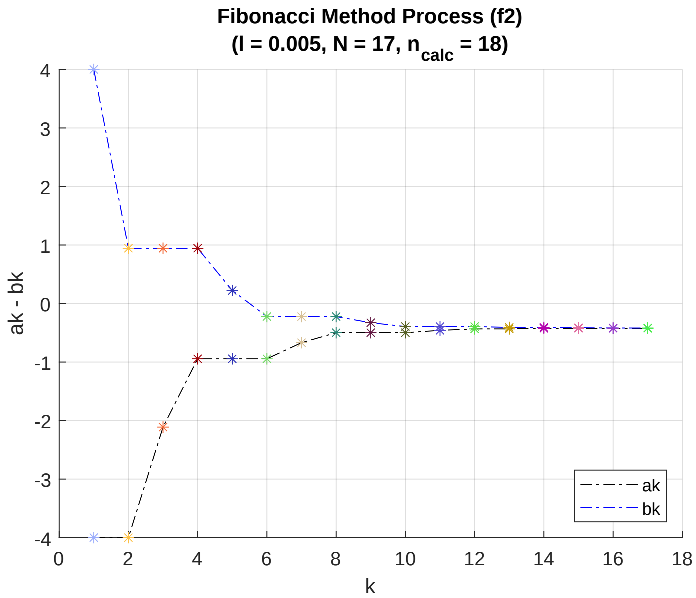

Figure 2.27: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

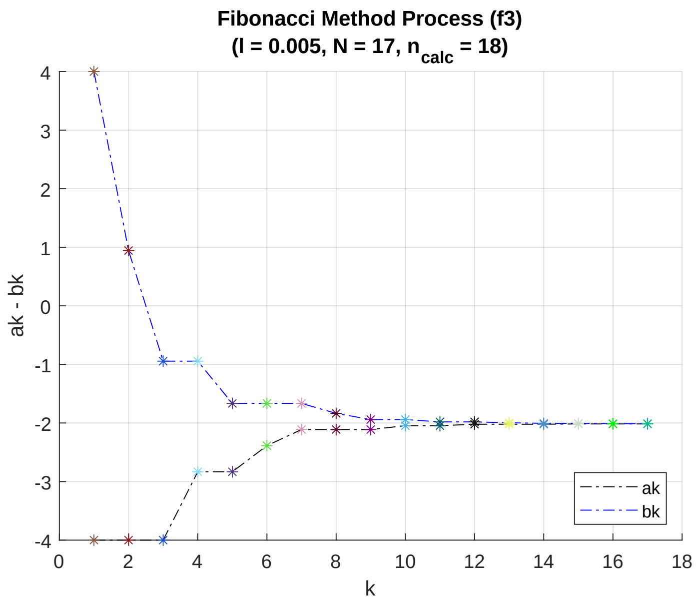

Figure 2.28: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

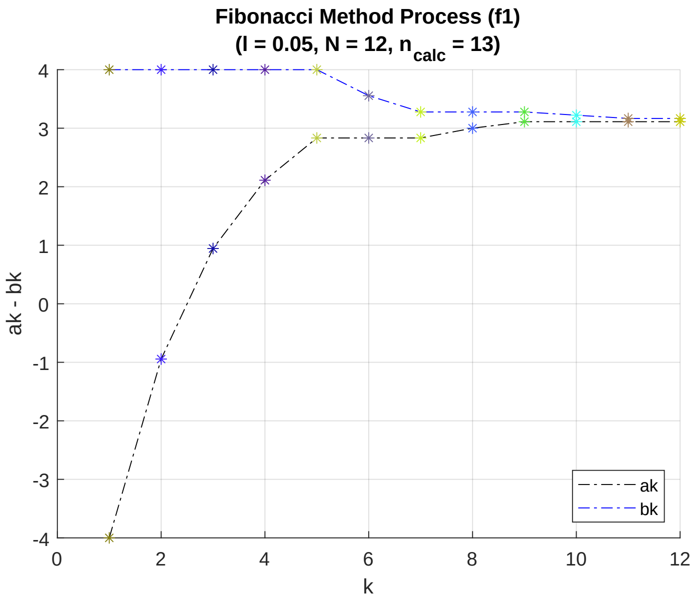

Figure 2.29: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

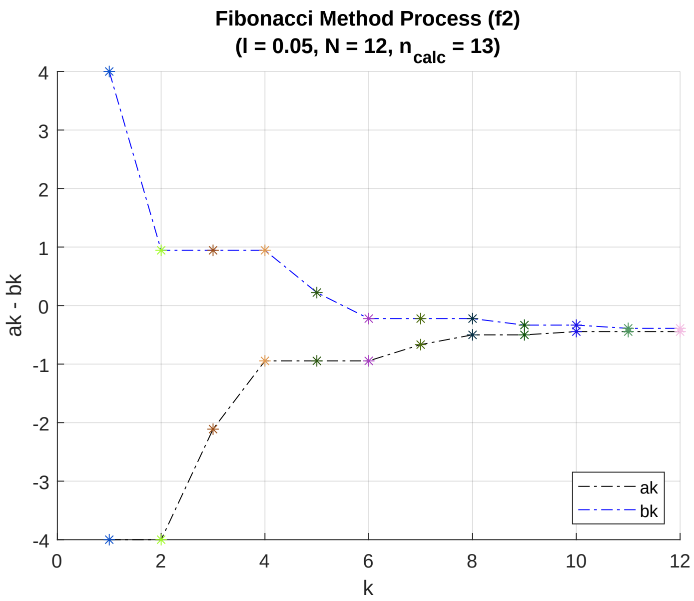

Figure 2.30: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

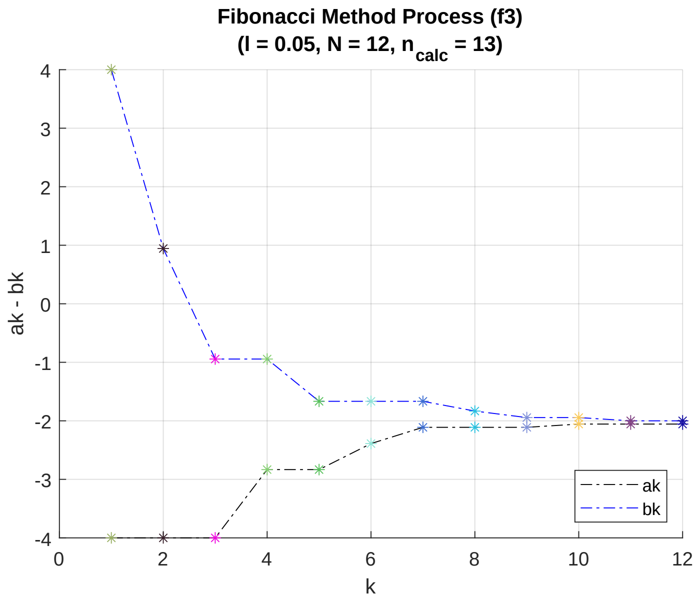

Figure 2.31: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

2.4 Μέθοδος Διχοτόμου - Με χρήση παραγώγων

2.4.1 Σύντομη περιγραφή της μεθόδου

Στη μέθοδο Διχοτόμου, με χρήση παραγώγων, ορίζουμε το x_{1_k} ως εξής:

$$x_{1_k} = \frac{a_k + b_k}{2} \tag{2.5}$$

όπου k η επανάληψη στην οποία βρισκόμαστε κάθε στιγμή. Στη συνέχεια ελέγχουμε στην παράγωγο της συνάρτησης f, που μας έχει δοθεί, f', την τιμή $f'(x_{1_k})$ και μετακινούμε τα αρχικά όρια $[a_0, b_0]$ προς το σημείο ελαχιστοποίησης της συνάρτησης με τον παρακάτω αλγόριθμο:

Μόλις ο αλγόριθμος βγει από το βρόχο, γνωρίζουμε ότι έχει βρεθεί προσεγγιστικά το ελάχιστο της συνάρτησης και είναι ίσο με $min = \frac{a_n+b_n}{2}$, όπου n είναι ο συνολικός αριθμός των επαναλήψεων του βρόχου.

2.4.2 Γραφήματα

Παρακάτω παραθέτουμε τα γραφήματα και τους πίνακες που εξήχθησαν από το πρόγραμμα στο MATLAB για τη συγκεκριμένη μέθοδο, τα οποία χρησιμοποιούνται για την εξαγωγή των συμπερασμάτων. Θεωρούμε ότι με l θέτουμε την ακρίβεια.

Algorithm 4 Set a_k, b_k in BisectionDer method

$k \leftarrow 0$
while $(b_k - a_k) > l$ **do**
  if $f'(x_{1_k}) > 0$ **then**
    $b_{k+1} \leftarrow x_{2_k}$
  else if $f(x_{1_k}) < 0$ **then**
    $a_{k+1} \leftarrow x_{1_k}$
  else
    break
  end if
  $x_{1_{k+1}} \leftarrow \frac{a_{k+1}+b_{k+1}}{2}$
  $k \leftarrow k + 1$
end while

Σύγκλιση στο ελάχιστο

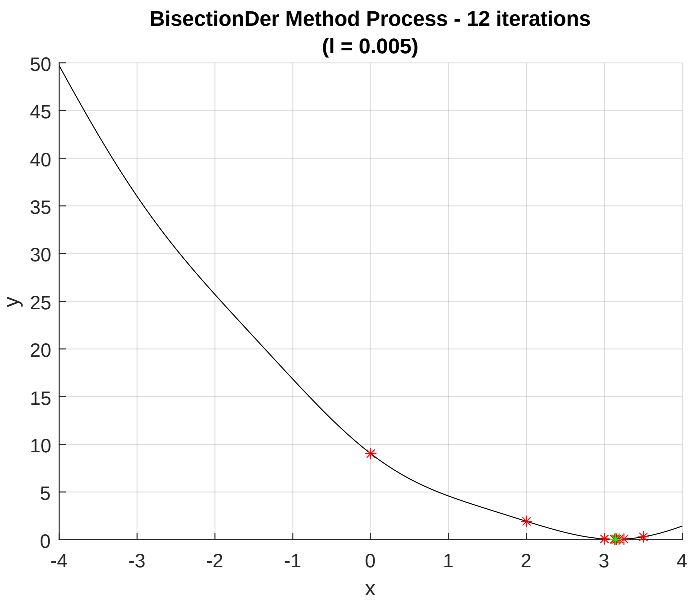

Figure 2.32: [F1] Convergence to minimum

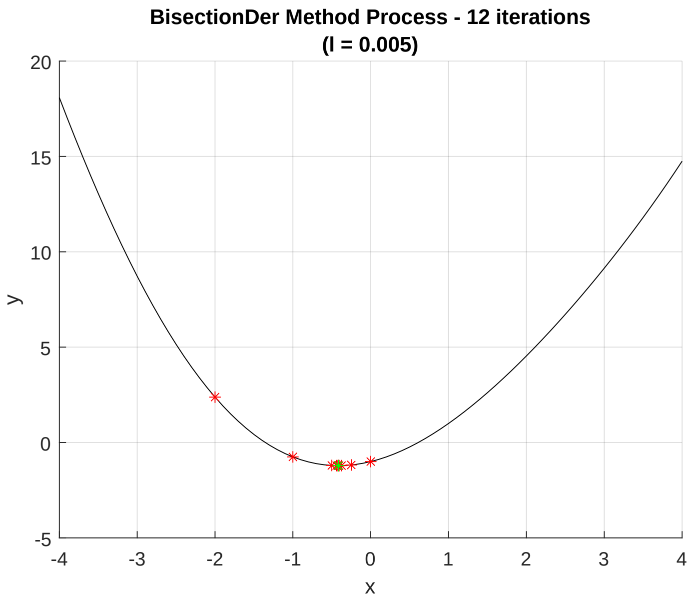

Figure 2.33: [F2] Convergence to minimum

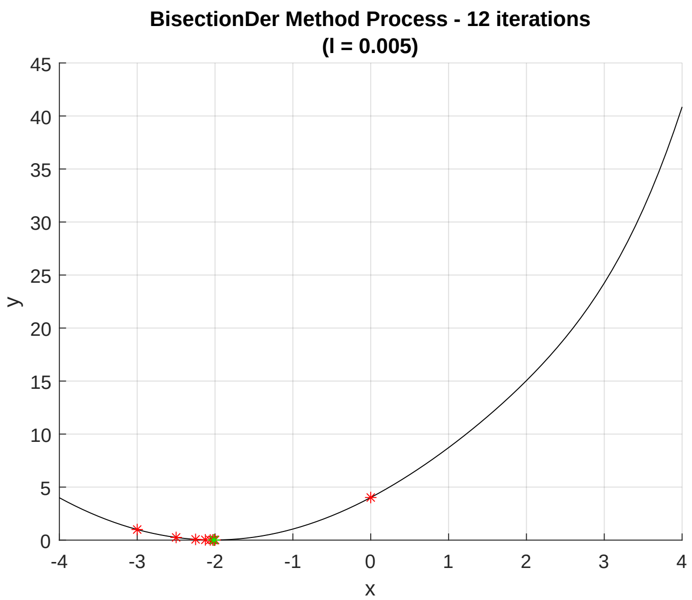

Figure 2.34: [F3] Convergence to minimum

Μεταβολή του l

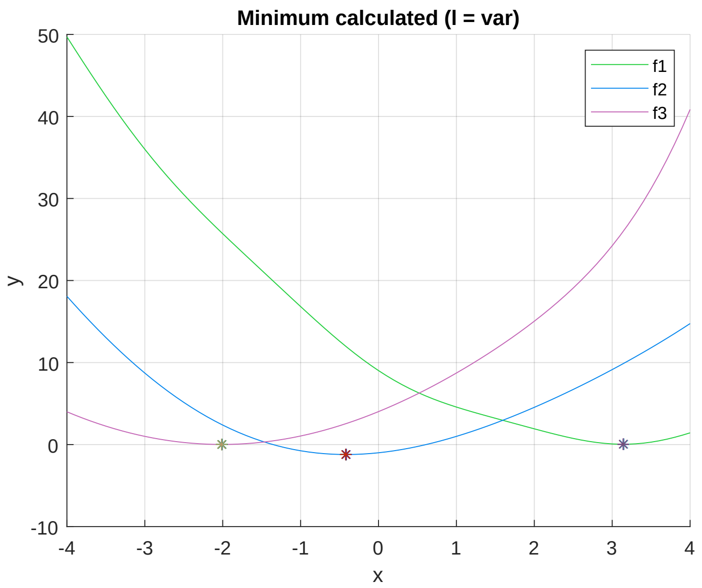

Figure 2.35: [F1, F2, F3] Variation of l

l	e	f1		f2		f3	
		x_min	**N**	**x_min**	**N**	**x_min**	**N**
0.005	0.001	3.14258	12	-0.416020	12	-2.01367	12
0.01	0.001	3.14453	11	-0.417970	11	-2.01172	11
0.015	0.001	3.14453	11	-0.417970	11	-2.01172	11
0.02	0.001	3.14844	10	-0.414060	10	-2.00781	10
0.025	0.001	3.14844	10	-0.414060	10	-2.00781	10
0.03	0.001	3.14844	10	-0.414060	10	-2.00781	10
0.035	0.001	3.14063	9	-0.421880	9	-2.01563	9
0.04	0.001	3.14063	9	-0.421880	9	-2.01563	9

Table 2.5: Variation of l

Σύγκλιση του διαστήματος $[a_k, b_k]$ στο ελάχιστο

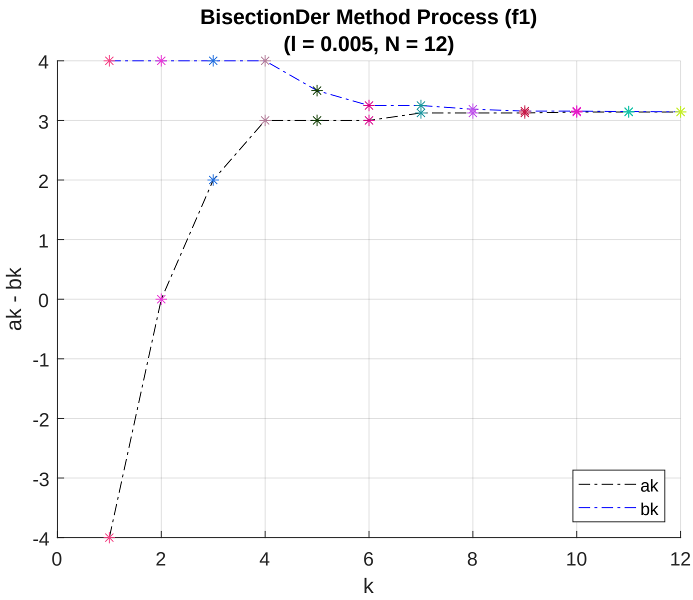

Figure 2.36: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

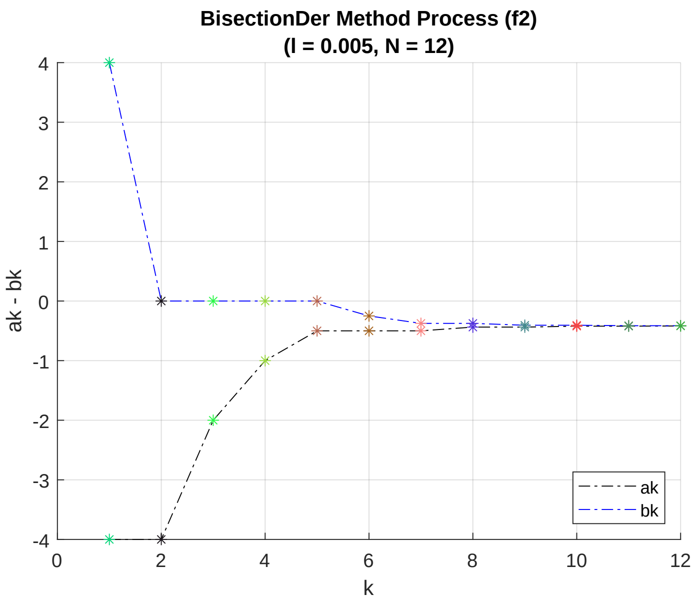

Figure 2.37: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

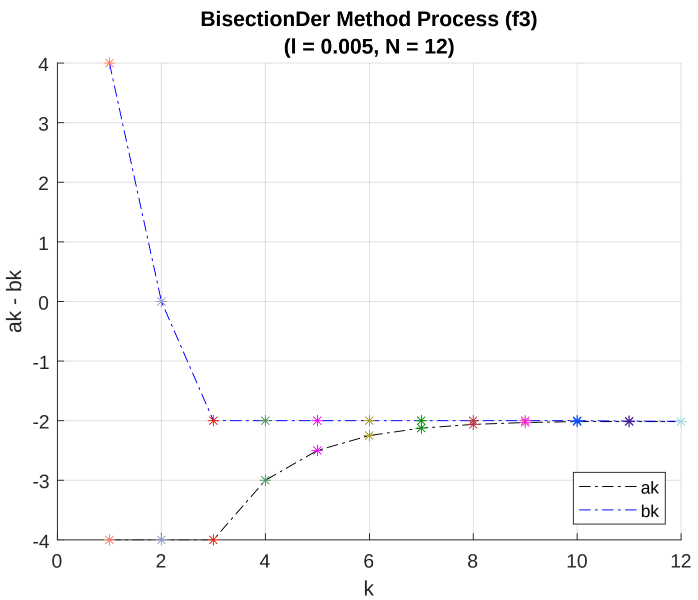

Figure 2.38: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.005$)

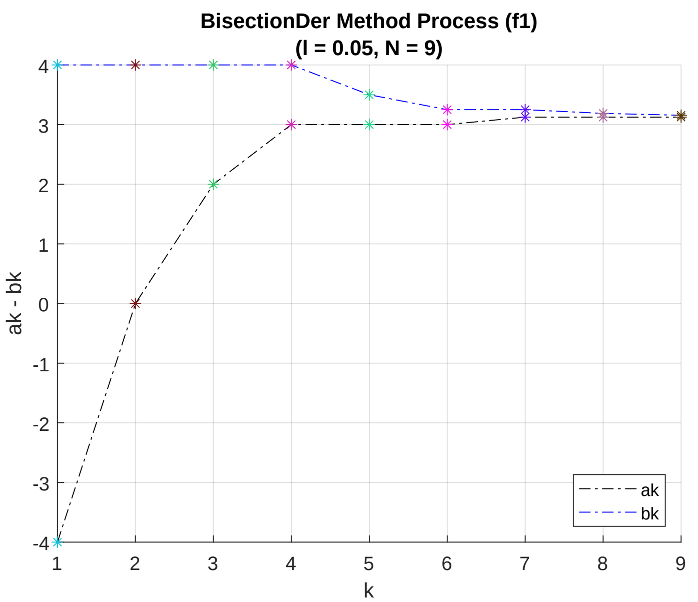

Figure 2.39: [F1] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

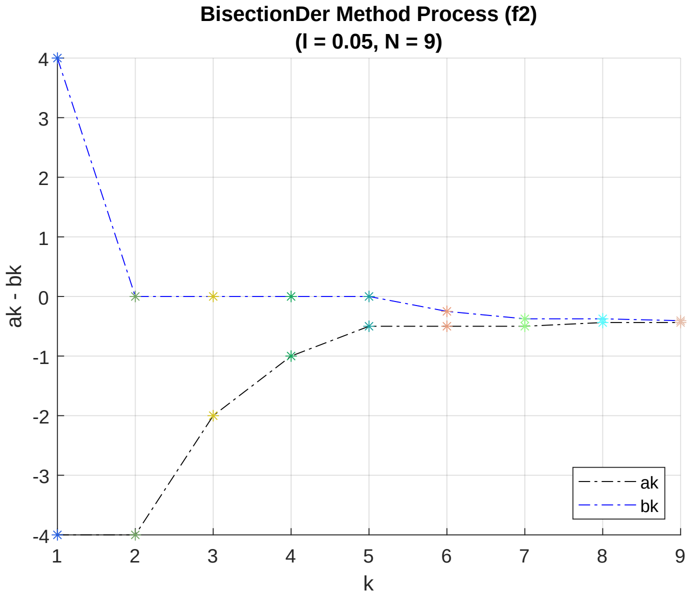

Figure 2.40: [F2] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

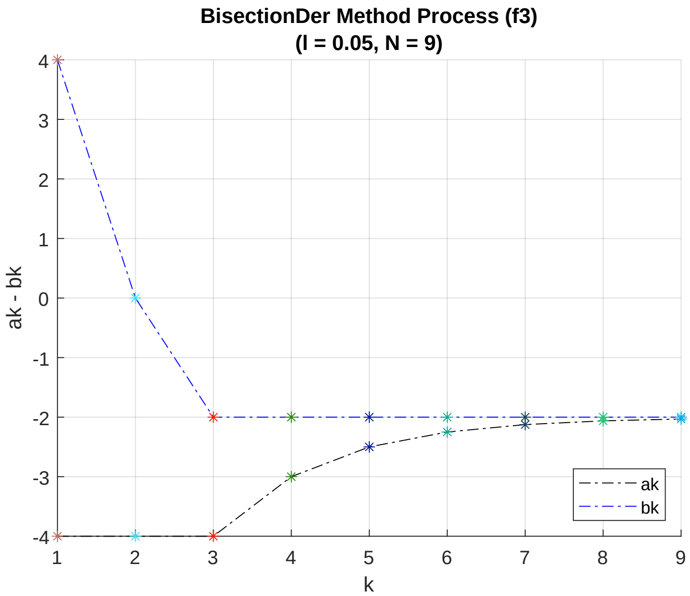

Figure 2.41: [F3] Convergence of $[a_k, b_k]$ to minimum ($l = 0.05$)

2.5 Γενικά Συμπεράσματα

Από τους πίνακες 2.1, 2.3, 2.4, 2.5 μπορούμε να παρατηρήσουμε ότι όσο μειώνουμε το l ο αλγόριθμος απαιτεί περισσότερες επαναλήψεις k για να υπολογίσει το ελάχιστο, ωστόσο επιτυγχάνεται σαφώς μεγαλύτερη ακρίβεια προσέγγισης του ελαχίστου της συνάρτησης.

Από τα διαγράμματα, μάλιστα, 2.4, 2.15, 2.25, 2.35, στα οποίο εμφανίζονται όλα τα ελάχιστα της κάθε συνάρτησης για τα διαφορετικά l που επιλέξαμε, μπορούμε να επιβεβαιώσουμε ότι για τόσο μικρές διαφορές της ακρίβειας, το ελάχιστο της συνάρτησης υπολογίζεται εξίσου καλά, καθώς τα σημεία πάνω στο διάγραμμα σχεδόν συμπέφτουν.

Φυσικά, είναι αισθητή η διαφορά της σύγκλισης των $[a_k, b_k]$ στο ελάχιστο της συνάρτησης (έχοντας λάβει το maximum και minimum l που δοκιμάσαμε παραπάνω), κρίνοντας από τη απόσταση που έχουν οι τιμές των $[a_k, b_k]$ για $k = N - 2, N - 1, N$, όπου N ο συνολικός αριθμός των επαναλήψεων του αλγορίθμου. Οι αποστάσεις αυτές διαφένονται στα διαγράμματα 2.6 - 2.11 (Μέθοδος Διχοτόμων χωρίς χρήση παραγώγων), 2.16 - 2.21 (Μέθοδος Χρυσού Τομέα), 2.26 - 2.31 (Μέθοδος Fibonacci) και 2.36 - 2.41 (Μέθοδος Διχοτόμου με χρήση παραγώγων).

Αντίστροφα, στην μέθοδο της Διχοτόμου χωρίς χρήση παραγώγων, από τον πίνακα 2.2 παρατηρούμε ότι όσο μεγαλώνουμε το e ο αλγόριθμος απαιτεί περισσότερες επαναλήψεις για να υπολογίσει το ελάχιστο, καθώς το διάστημα $[a_k, b_k]$ πλησιάζει πιο αργά στην ακρίβεια l. Βέβαια, βλέπουμε ότι, όπως και με τη μεταβολή του l σε όλες τις μεθόδους που μελετήσαμε, έτσι και η μείωση του e στη συγκεκριμένη μέθοδο οδηγεί σε πιο ακριβή προσέγγιση της τιμής του ελαχίστου. Τα ελάχιστα που προκύπτουν από τη μεταβολή του e προβάλλονται στο 2.5.

Κεφάλαιο 3

Σύγκριση Μεθόδων

l	e	Bisection NoDer	Golden Section	Fibonacci	Bisection Der
0.0001	0.00001	18	25	25	18
0.005	0.0005	12	17	17	12
0.01	0.001	11	15	15	11
0.05	0.005	9	12	12	9
0.1	0.01	8	11	10	8

Table 3.1: Συνολικές επαναλήψεις για εύρεση ελαχίστου

l	e	Bisection NoDer	Golden Section	Fibonacci
0.0001	0.00001	36	25	26
0.005	0.0005	24	17	18
0.01	0.001	22	15	16
0.05	0.005	18	12	13
0.1	0.01	16	11	11

Table 3.2: Συνολικοί υπολογισμοί αντικειμενικής συνάρτησης $f(i), i = 1, 2, 3$

Δεδομένης της αυστηρά σχεδόν-κυρτής συνάρτησης φ στο διάστημα $[a_0, b_0]$, όλες οι μέθοδοι που μελετήθηκαν στην συγκεκριμένη εργασία υπολογίζουν το ελάχιστο με ακρίβεια l, σε πεπερασμένο αριθμό επαναλήψεων.

Μελετώντας όλες τις μεθόδους μαζί, εξήχθησαν οι πίνακες 3.1 και 3.2. Στον πρώτο πίνακα, για διαφορετικές τιμές του l (και του e στη μέθοδο Διχοτόμου χωρίς παραγώγους), καταγράφηκαν οι συνολικές επαναλήψεις του αλγορίθμου για την εύρεση του ελαχίστου στις 3 συναρτήσεις, $f1, f2, f3$. Στον δεύτερο πίνακα εμφανίζονται οι συνολικοί υπολογισμοί των αντικειμενικών συναρτήσεων $f1, f2, f3$ για τις μεθόδους που δεν χρησιμοποιούν παραγώγους, θεωρώντας ότι το πρόγραμμα μας κρατάει στη μνήμη της τιμές της συνάρτησης $fi, i = 1, 2, 3$ που έχει ήδη υπολογίσει.

Μπορούμε εύκολα να παρατηρήσουμε ότι η μέθοδος Διχοτόμου (με και χωρίς τη χρήση παραγώγων) απαιτούν τον ίδιο αριθμό επαναλήψεων για να προσεγγίσουν με την ίδια ακρίβεια το ελάχιστο των συναρτήσεων.

Αντίστοιχα μπορούμε να δούμε ότι μεταξύ των μεθόδων Χρυσού Τομέα και Fibonacci υπαρχεί σχετικά κοινός αριθμός επαναλήψεων για διαφορετικά l. Αυτό συμβαίνει καθώς, όπως αναφέραμε στο προηγούμενο κεφάλαιο, το μόνο που αλλάζει μεταξύ των 2 μεθόδων είναι ο παράγοντας γ, ο οποίος πλέον αντικαθίσταται από τον λόγο $\frac{fibonacci(n-k)}{fibonacci(n-k+1)}$. Ισχύει όμως ότι ο λόγος $\frac{1}{fibonacci(n)}$ συγκλίνει στο $\gamma = 0.618$, που είναι και η βέλτιστη τη τιμή που λαμβάνει το γ στη μέθοδο του Χρυσού Τομέα, επομένως οδηγεί στη μεγάλη ομοιομορφία που παρουσιάζεται μεταξύ των 2 μεθόδων.

Επίσης, συμπεραίνουμε ότι η Μέθοδος Χρυσού τομέα απαιτεί τους λιγότερους υπολογισμούς της αντικειμενικής συνάρτησης (θεωρώντας ότι το πρόγραμμα μας κρατάει στη μνήμη της τιμές της συνάρτησης $fi, i = 1, 2, 3$ που έχει ήδη υπολογίσει), διαφέροντας κατά 1 από τους αντίστοιχους υπολογισμούς στη μέθοδο Fibonacci. Αντιθέτως, οι αντίστοιχοι υπολογισμοί στη Μέθοδο Διχοτόμου χωρίς τη χρήση παραγώγων είναι αρκετά μεγαλύτερη, καθώς χρειάζεται να καλέσει 2 φορές τη συνάρτηση για διαφορετικές τιμές του x σε κάθε *loop* για να υπολογίσει τις 2 τιμές $f(x1), f(x2)$. Οι συνολικοί υπολογισμοί τις f μπορεί στα παραδείγματά μας να επηρεάζουν ελάχιστα την ταχύτητα του αλγορίθμου, ωστόσο, αν η συνάρτηση περιλαμβάνει απαιτητικές υπολογιστικά πράξεις, ο αριθμός κλήσεων των συναρτήσεων για διαφορετικές τιμές του x μπορεί να αποτελέσει καίριο παράγοντα στην ταχύτητα.

Στην περίπτωση της μεθόδου Διχοτόμου με τη χρήση παραγώγων, πρακτικά καλούμε 1 φορά σε κάθε *loop* την παράγωγο της συνάρτησης $f'(x1)$.

Τέλος, αξίζει να τονίσουμε ότι ένα μειονέκτημα της μεθόδου Fibonacci είναι το ότι περιλαμβάνει τον υπολογισμό του αριθμού Fibonacci για διάφορα k, καθώς και τον υπολογισμό του n μέσω της σχέσης 2.4, όπου $n - 1$ ο συνολικός αριθμός των επαναλήψεων (πριν την εκτέλεση του αλγορίθμου).