Τεχνικές Βελτιστοποίησης

3η εργασία

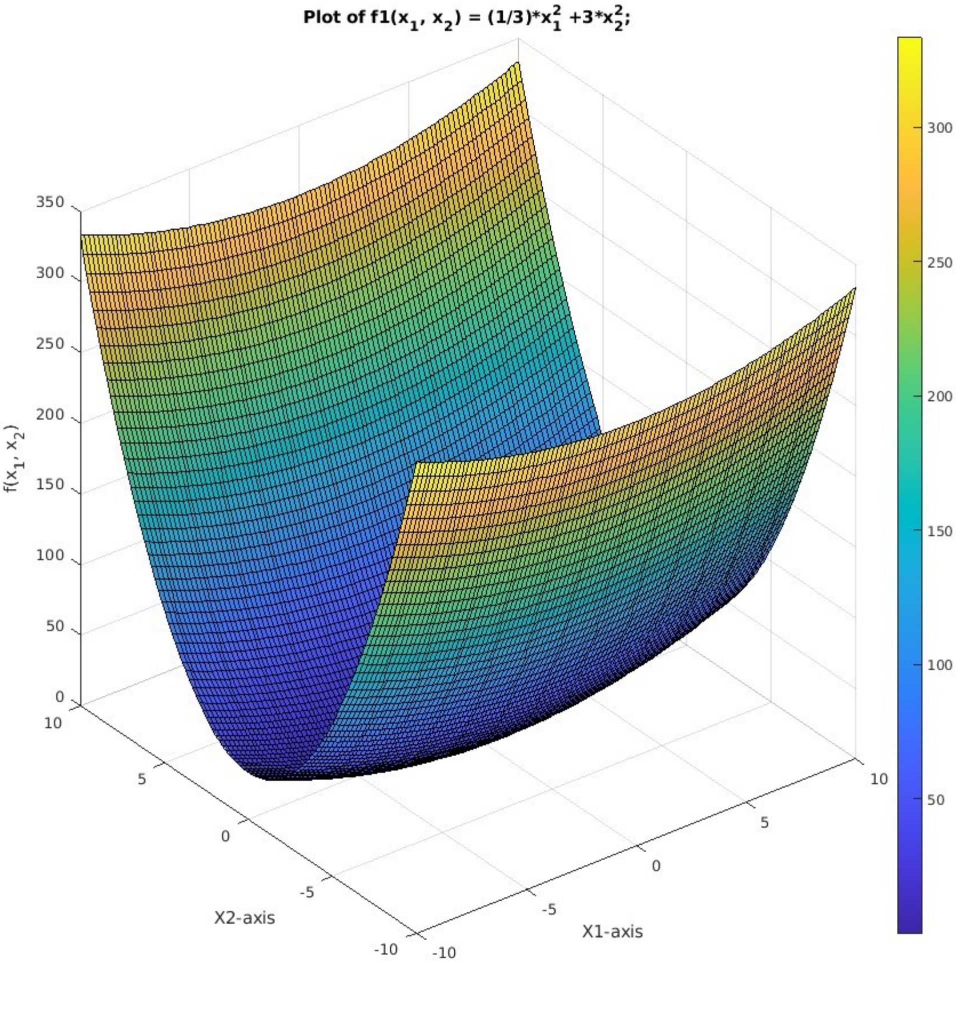
Ονοματεπώνυμο: Παναγιώτης Λεκός
ΑΕΜ: 10347
Email: panaleko@ece.auth.gr

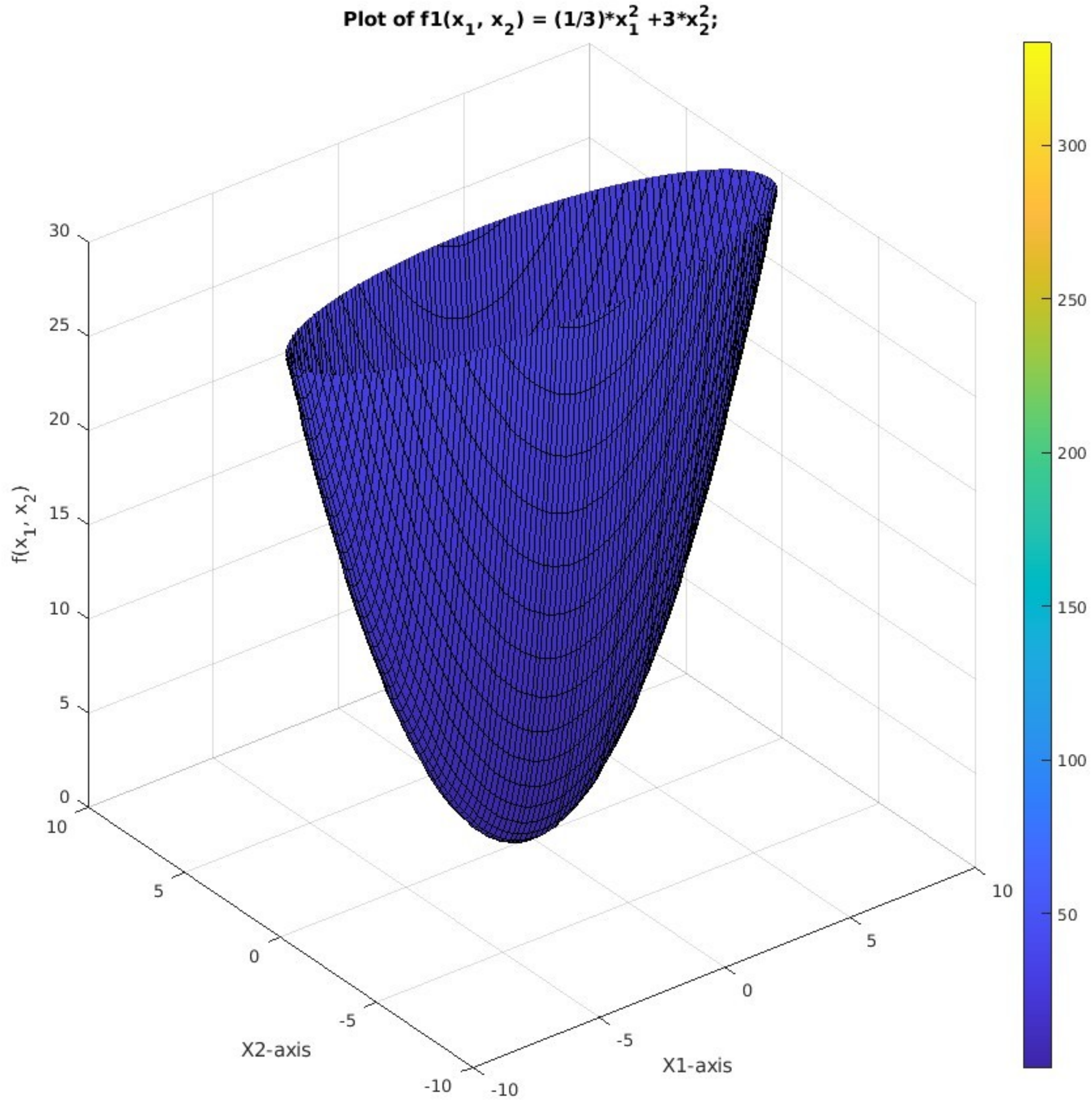
Στην παρούσα εργασία θα μελετήσουμε την μέθοδο της μέγιστης καθόδου με προβολή, χρησιμοποιώντας μια αντικειμενική συνάρτηση 2 μεταβλητών, και περιορισμούς στις τιμές των ορισμάτων της. Θα υλοποιήσουμε στο MATLAB τον αλγόριθμο της μεθόδου, και θα τον εκτελέσουμε για διαφορετικές τιμές των παραμέτρων της και για διαφορετικά σημεία εκκίνησης. Επίσης, όπως και στις προηγούμενες εργασίες, θα μελετήσουμε και θα σχολιάσουμε το υπολογιστικό κόστος που έχει η εκτέλεση του αλγορίθμου κάθε φορά.

Η αντικειμενική συνάρτηση που θα χρησιμοποιήσουμε είναι η

$$f: R^2 \to R, \quad f(x_1, x_2) = \frac{1}{3} * x_1^2 + 3 * x_2^2$$

Είναι εμφανές πως το πραγματικό ελάχιστο της συνάρτησης είναι το 0 στο σημείο (0,0). Υλοποιούμε την f στο αρχείο f.m και την οπτικοποιούμε στο αρχείο plot_f.m. Τα figures που παράχθηκαν φαίνονται παρακάτω. Στο πρώτο γράφημα, η κλίμακα του άξονα z προσαρμόζεται αυτόματα, με αποτέλεσμα να μην είναι πολύ ευδιάκριτη η μορφή της συνάρτησης. Στο δεύτερο γράφημα, περιορίσαμε το εύρος του z (z-limit) για να φανεί καλύτερα η καμπυλότητά της.

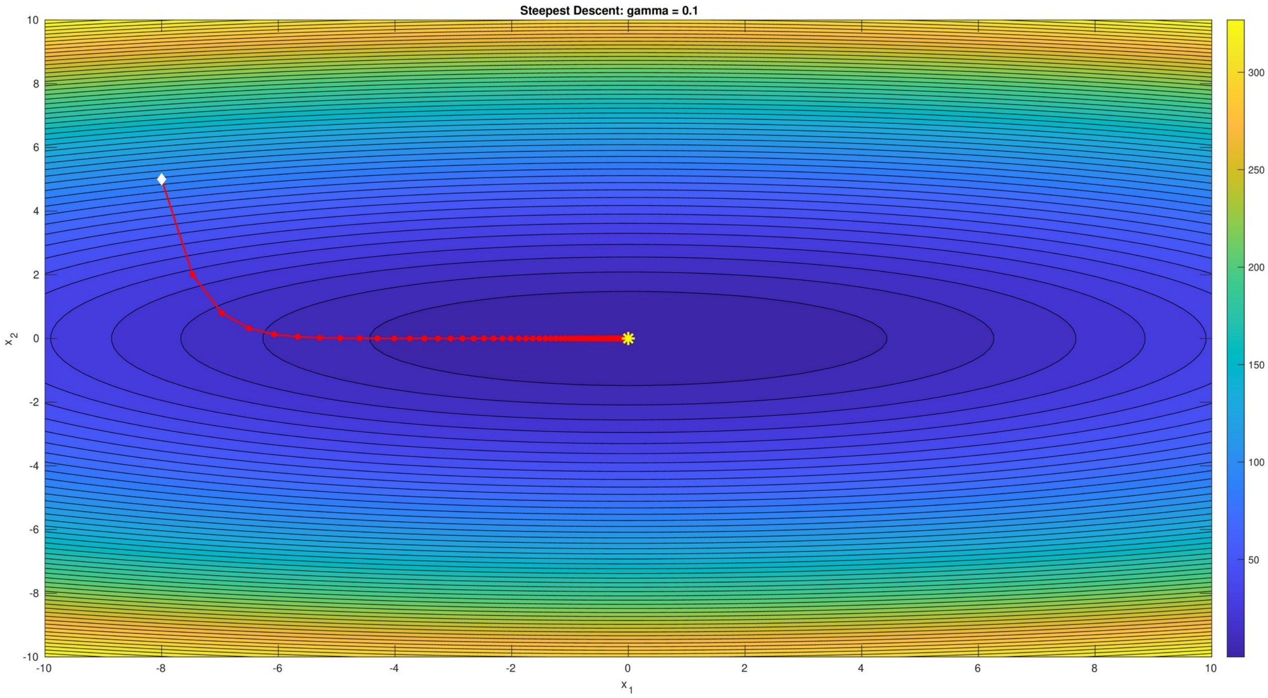




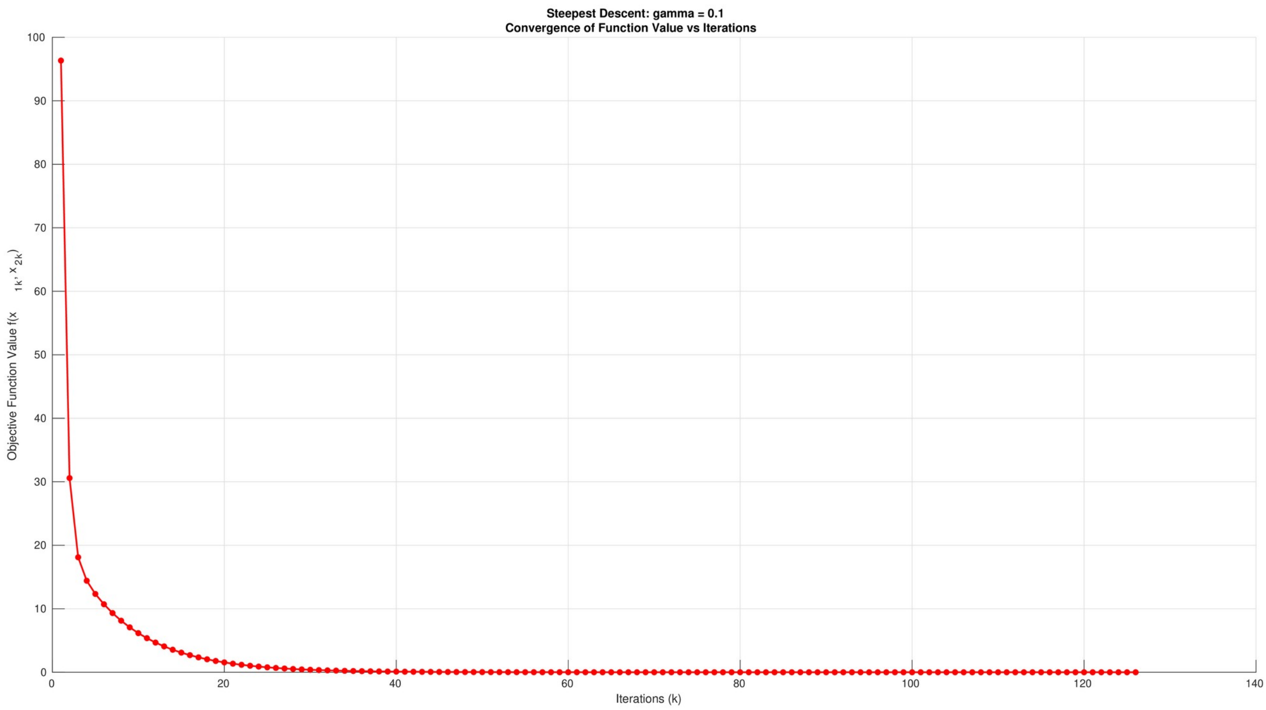
<u>Μέθοδος μέγιστης καθόδου (χωρίς προβολή)</u>

Αρχικά κάνουμε κάποιες δοκιμές να ελαχιστοποιήσουμε την f με την μέθοδο μέγιστης καθόδου χωρίς προβολή όπως υλοποιήθηκε στην προηγούμενη εργασία (αρχείο steepest_descent_a.m). Στη μέθοδο αυτή έχουμε σταθερό βήμα γ_κ για το οποίο δοκιμάζουμε τις τιμές 0.1, 0.3, 3, και 5, και το σημείο εκκίνησης (-8,5) που επιλέχθηκε τυχαία. Εκτελώντας τον αλγόριθμο για τις παραμέτρους αυτές (Thema_1.m) εξάγουμε τα παρακάτω γραφήματα που απεικονίζουν την πορεία του.

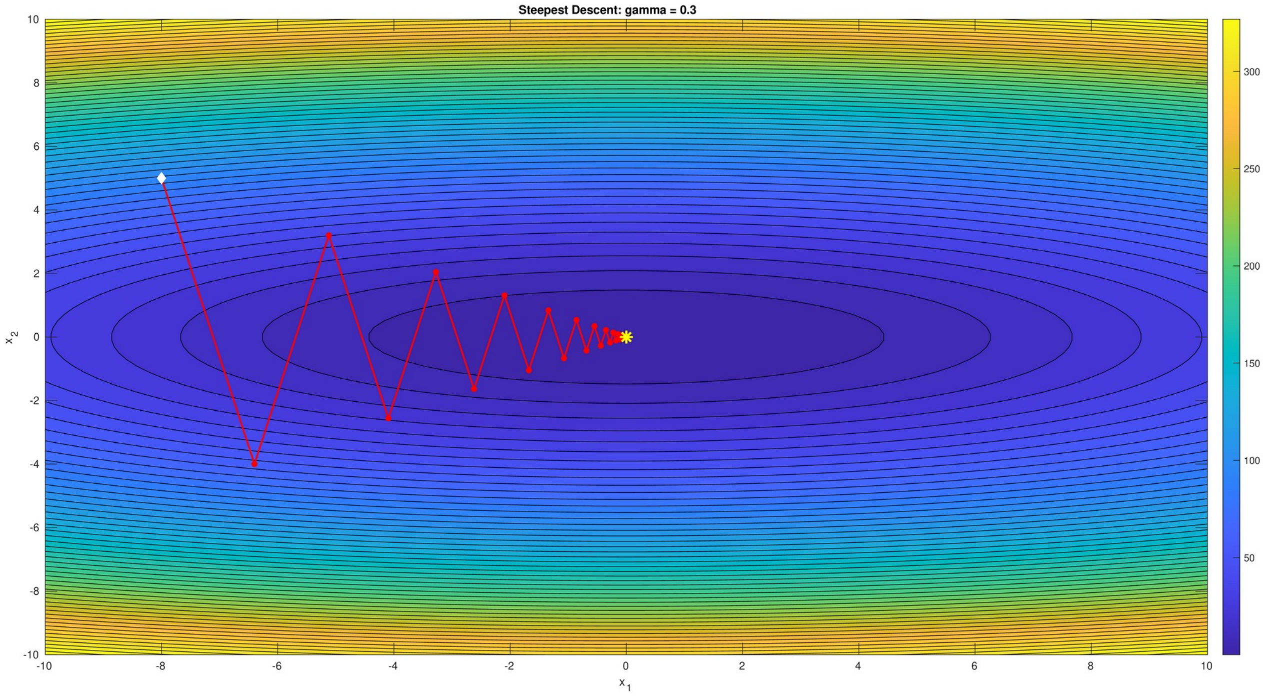
γ_κ	Επαναλήψεις / Υπολογισμοί ∇f	Ελάχιστο
0.1	126	$6.89*10^{-7}$
0.3	48	$7.48*10^{-8}$
3	-	-
5	-	-



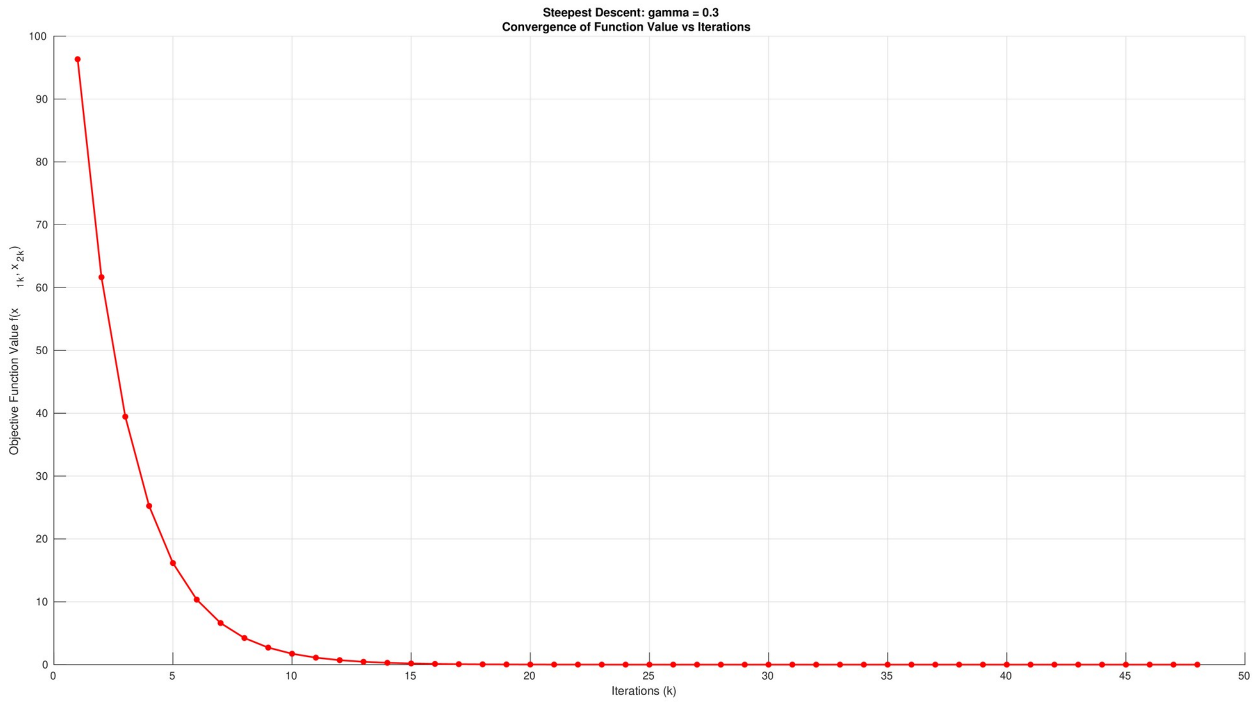
Steepest Descent: gamma = 0.1
x_2
x_1



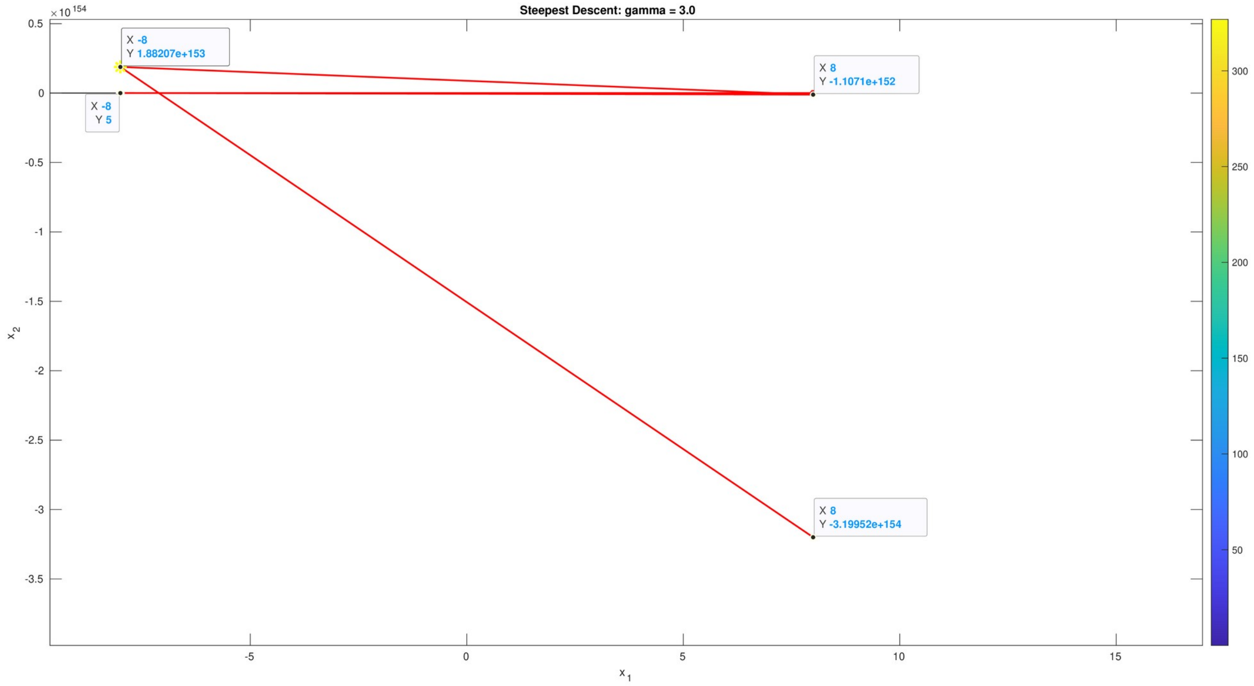
Steepest Descent: gamma = 0.1
Convergence of Function Value vs Iterations
Objective Function Value f(x_1k, x_2k)
Iterations (k)



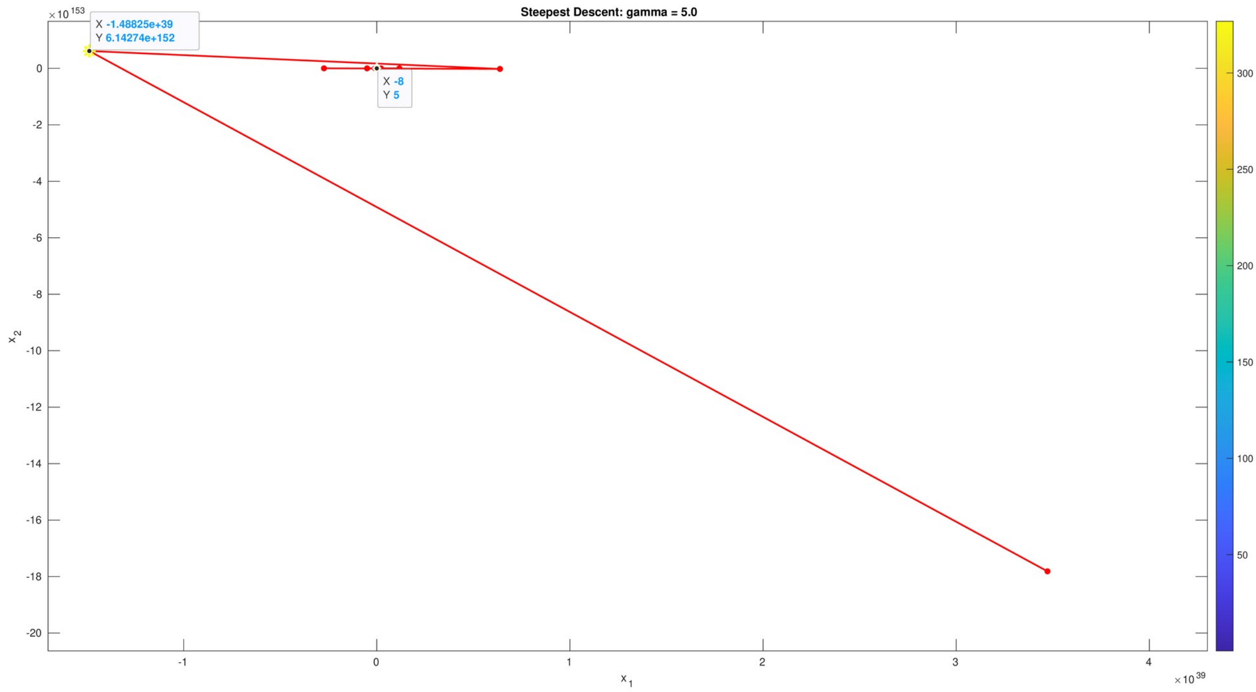
Steepest Descent: gamma = 0.3
x_2
x_1
10
8
6
4
2
0
-2
-4
-6
-8
-10
300
250
200
150
100
50



Steepest Descent: gamma = 0.3
Convergence of Function Value vs Iterations
Objective Function Value $f(x_{1k}, x_{2k})$
Iterations (k)
0
5
10
15
20
25
30
35
40
45
50
60
70
80
90
100



Steepest Descent: gamma = 3.0
×10^154
X -8
Y 1.88207e+153
X 8
Y -1.1071e+152
X -8
Y 5
X 8
Y -3.19952e+154
x 1
x 2



Steepest Descent: gamma = 5.0
×10^153
X -1.48825e+39
Y 6.14274e+152
X -8
Y 5
x 1
x 2
×10^39

Παρατηρήσεις

- $\gamma_\kappa = 0.1$: Ο αλγόριθμος συγκλίνει στο 0 σχετικά αργά, μετά από 126 επαναλήψεις.
- $\gamma_\kappa = 0.1$: Ο αλγόριθμος συγκλίνει στο 0 μετά από 48 επαναλήψεις και υπάρχει μια ταλάντωση κατά μήκος του άξονα x_2.
- $\gamma_\kappa = 3$: Δεν καταφέρνει να συγκλίνει γιατί απειρίζεται.
- $\gamma_\kappa = 5$: Δεν καταφέρνει να συγκλίνει γιατί απειρίζεται.

Στη μέθοδο μέγιστης καθόδου κάθε σημείο αναζήτησης υπολογίζεται σύμφωνα με τον τύπο

$$x_{(k+1)} = x_k - \gamma_\kappa * \nabla f(x_k)$$

ο οποίος εάν υπολογίσουμε και αντικαταστήσουμε την αναλυτική $\nabla f(x_k) = \begin{bmatrix} \frac{2}{3} * x_1 \\ 6 * x_2 \end{bmatrix}$, και αναπτύξουμε το x σε $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ καταλήγουμε στη σχέση:

$$\begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} = \begin{bmatrix} x_{(1,k)} * (1 - \frac{2}{3} * \gamma_\kappa) \\ x_{(2,k)} * (1 - 6 * \gamma_\kappa) \end{bmatrix}$$

- $\gamma_\kappa = 0.1$: $\begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} = \begin{bmatrix} 0.933 * x_{(1,k)} \\ 0.4 * x_{(2,k)} \end{bmatrix}$

  Παρατηρώντας τον τύπο καταλαβαίνουμε ότι το σημείο θα συγκλίνει με το πέρασμα των επαναλήψεων. Όμως η τιμή του x_1 συγκλίνει με αρκετά αργό ρυθμό λόγω του πολλαπλασιαστή 0.933. Φαίνεται και από το γράφημα ότι συγκλίνει γρηγορότερα κατά x_2 και πιο αργά για x_1.

- $\gamma_\kappa = 0.3$: $\begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} = \begin{bmatrix} 0.8 * x_{(1,k)} \\ -0.8 * x_{(2,k)} \end{bmatrix}$

  Σε αυτήν την περίπτωση θα συγκλίνει λίγο γρηγορότερα από πριν, όπως μπορούμε να συμπεράνουμε από τον πολλαπλασιαστή 0.8, και μάλιστα η ταλάντωση που παρατηρήσαμε προηγουμένως οφείλεται στο αρνητικό πρόσημο στον πολλαπλασιαστή του $x_{(2,k)}$.

- $\gamma_\kappa = 3$: $\begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} = \begin{bmatrix} -1 * x_{(1,k)} \\ 17 * x_{(2,k)} \end{bmatrix}$

  Εδώ το x_1 θα ταλαντώνεται για πάντα χωρίς να συγκλίνει και το x_2 θα αυξάνεται σε κάθε επανάληψη, οπότε ο αλγόριθμος δεν θα συγκλίνει.

- $\gamma_\kappa = 5$: $\begin{bmatrix} x_{(1,k+1)} \\ x_{(2,k+1)} \end{bmatrix} = \begin{bmatrix} -2.33 * x_{(1,k)} \\ -29 * x_{(2,k)} \end{bmatrix}$

  Πάλι εδώ τα x_1 , x_2 θα ταλαντώνονται και θα αυξάνονται σε κάθε επανάληψη οπότε θα απειριστεί.

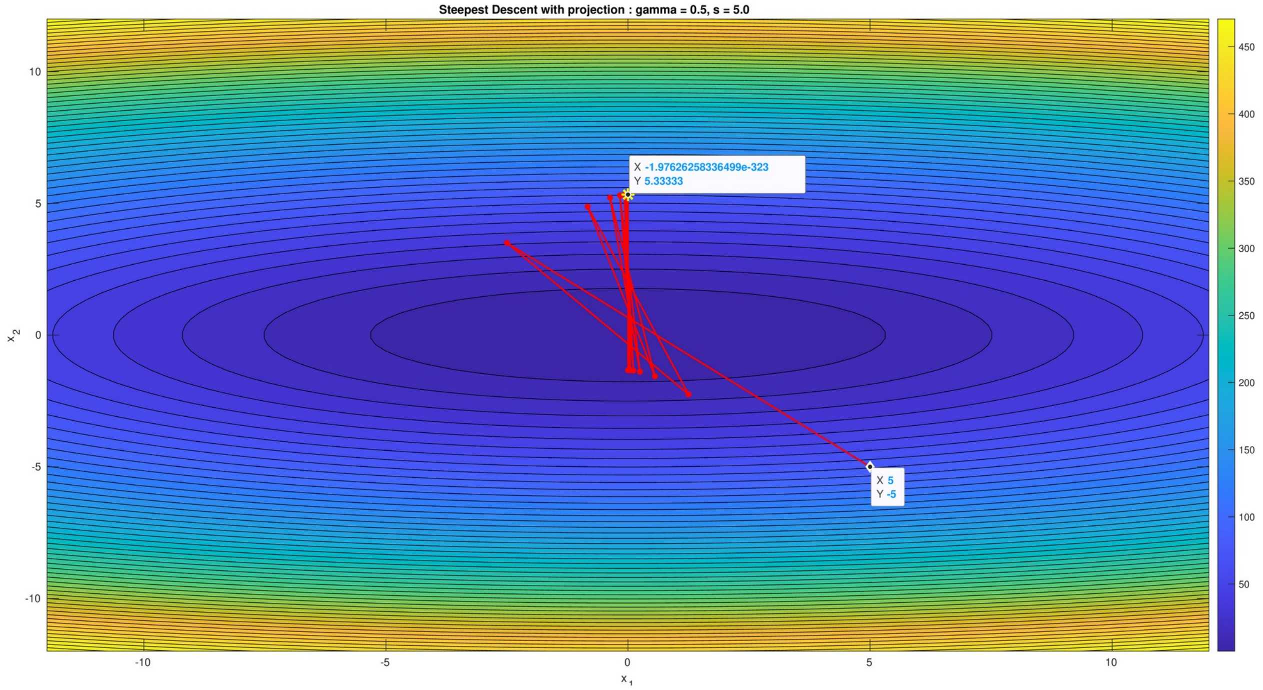
Τώρα εισάγουμε τους περιορισμούς:

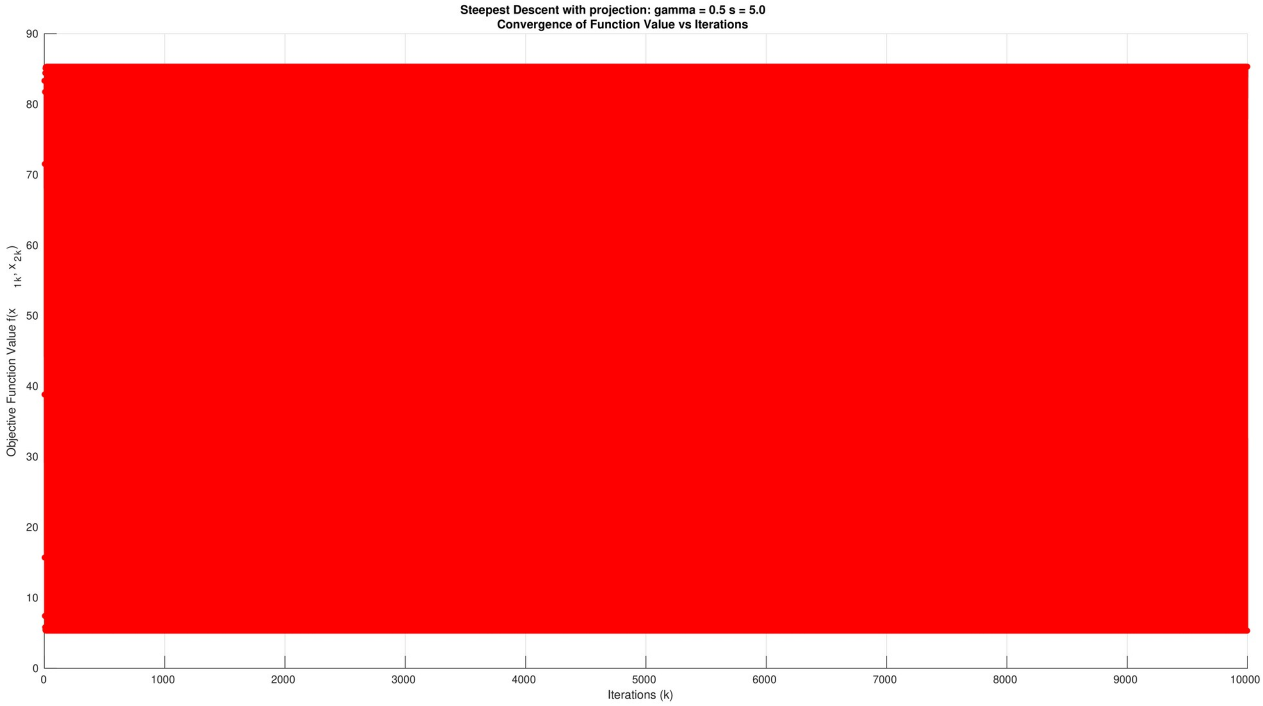
$$-10 \le x_1 \le 5 \quad \text{και} \quad -8 \le x_2 \le 12$$

με τους οποίους θα μελετήσουμε την μέθοδο μέγιστης καθόδου με προβολή. Η υλοποίηση του αλγορίθμου είναι στο αρχείο steepest_descent_proj.m. Για την υλοποίηση της χρειάστηκε επίσης η συνάρτηση proj_x.m η οποία μας δίνει την προβολή ενός σημείου στο παραπάνω διάστημα των περιορισμών.

Μέθοδος μέγιστης καθόδου με προβολή
Εκτελούμε τη μέθοδο με διάφορα σετ παραμέτρων και παρατηρούμε τα αποτελέσματα.

- S_k =5 , γ_k = 0.5 , ακρίβεια ε = 0.01 , σημείο εκκίνησης: (5,-5) (Thema_2.m)

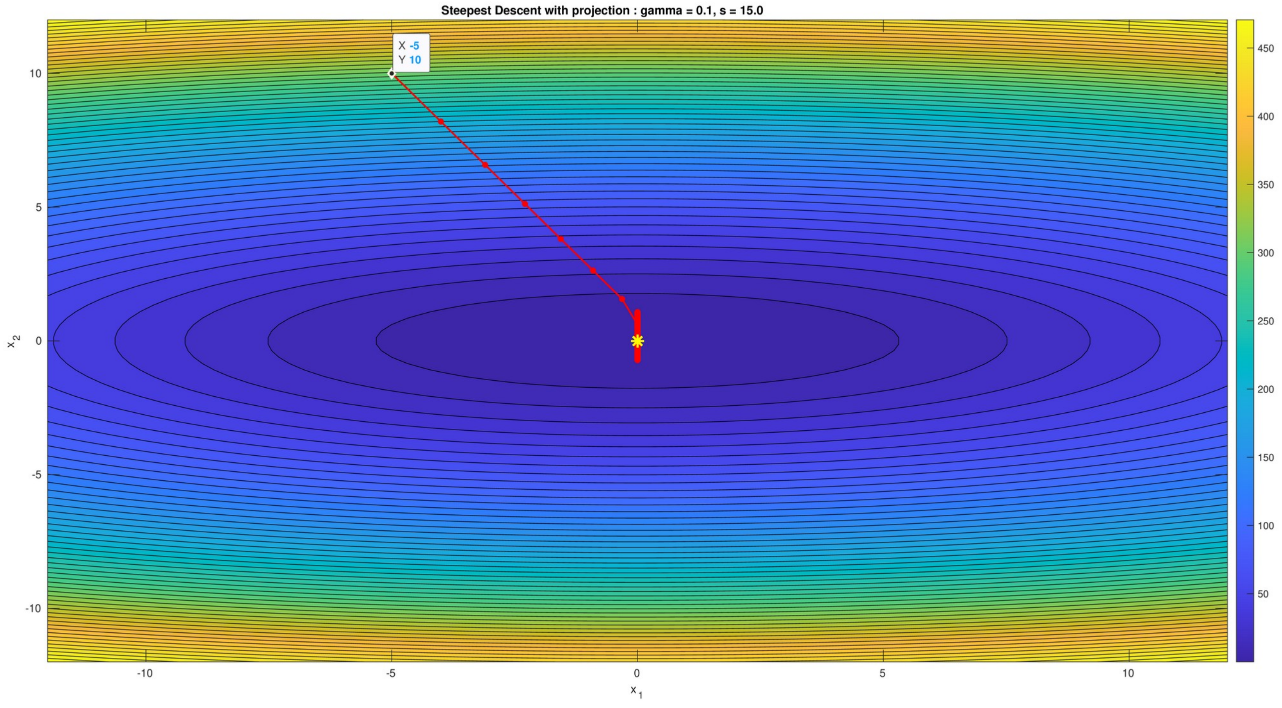


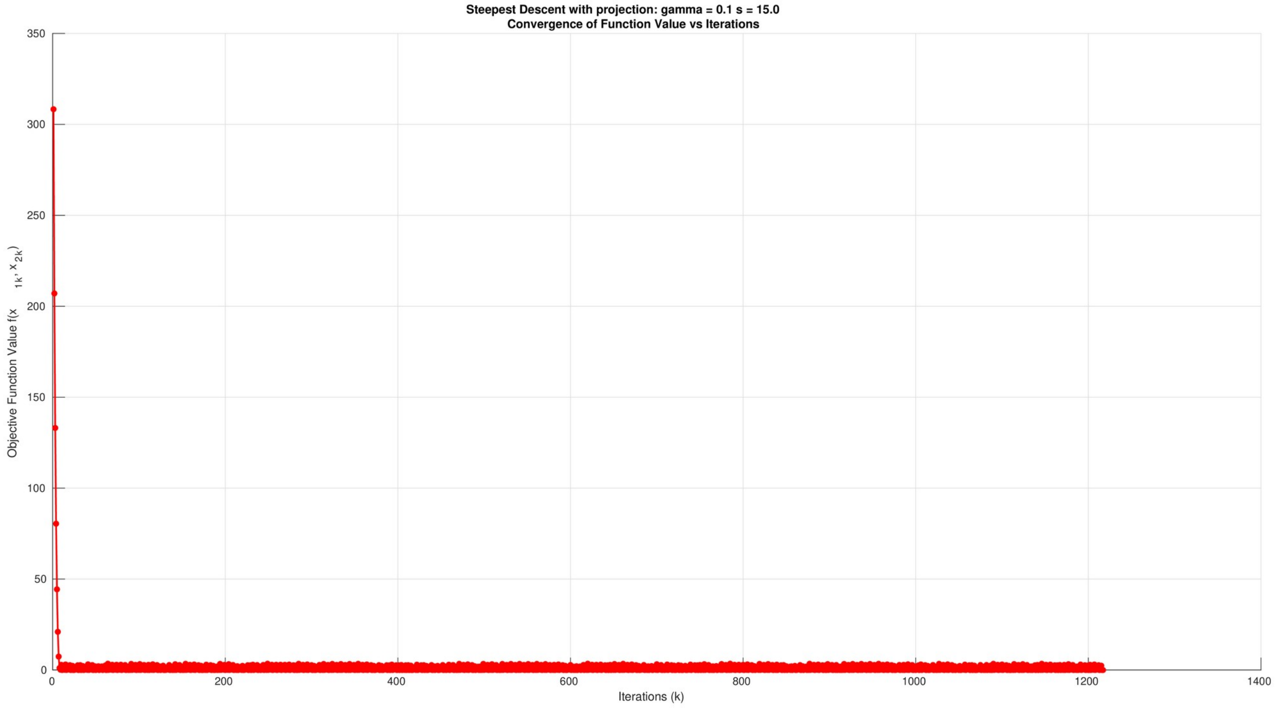


Παρατηρήσεις

Ο αλγόριθμος εκτελεί τον μέγιστο αριθμό επαναλήψεων (10000). Υπάρχει έντονη ταλάντωση στον άξονα x_2 και καταλήγει στο σημείο (0, 5.333) όπου f(0, 5.333) = 85.33 αδυνατώντας να βρει το ελάχιστο. Η ταλάντωση αυτή συμβαίνει λόγω του μεγάλου βήματος s_k = 5. Εδώ ,σε αντίθεση με τη συμπεριφορά της μεθόδου χωρίς προβολή με μεγάλο βήμα, ο αλγόριθμος δεν απειρίζεται. Αυτό συμβαίνει διότι η προβολή επαναφέρει στο σύνολο των περιορισμών τις μεγάλες τιμές που μπορεί να πάρουν τα x_k.

- S_k =15 , γ_k = 0.1 , ακρίβεια ε = 0.01 , σημείο εκκίνησης: (-5,10) (Thema_3.m)

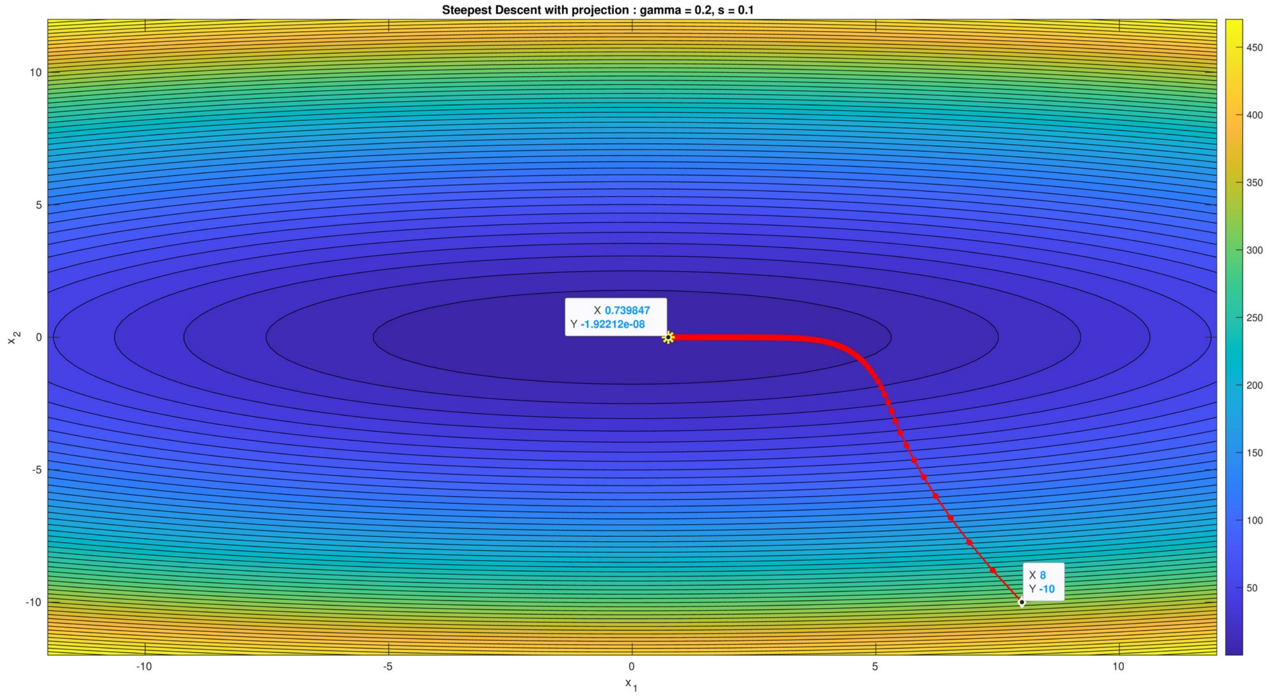


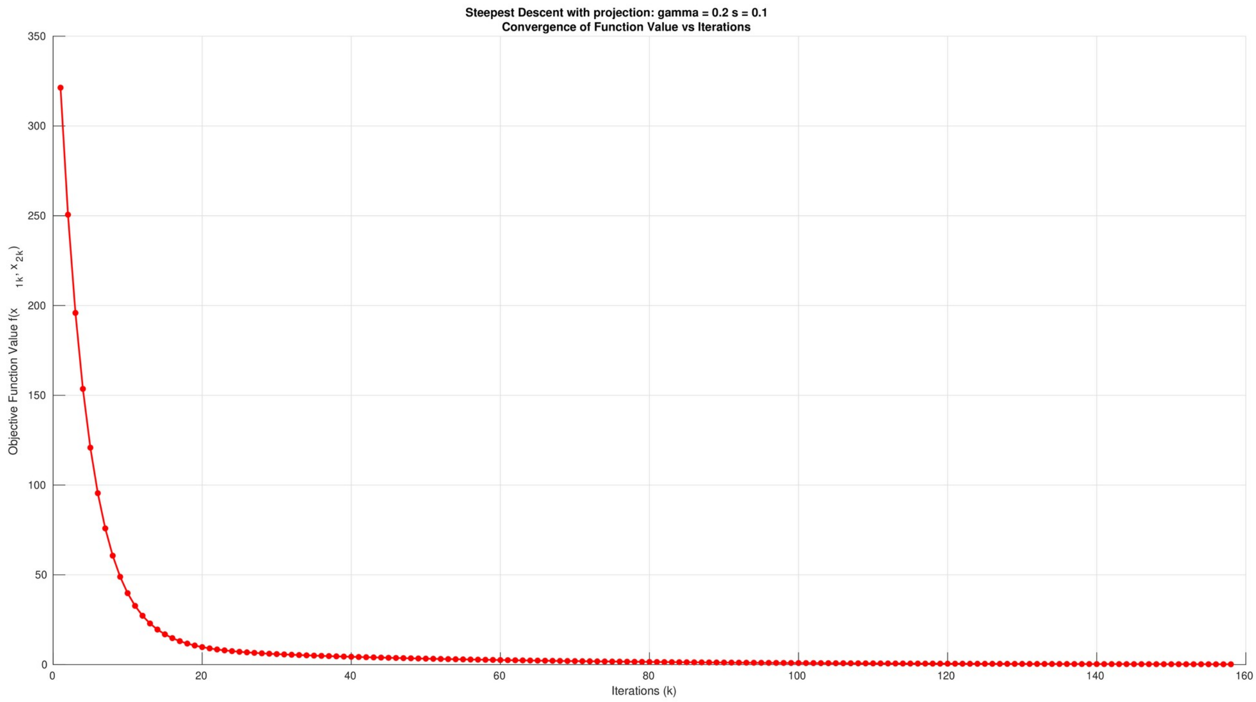


Παρατηρήσεις

Ο αλγόριθμος εκτελεί 1216 επαναλήψεις και καταλήγει στο σημείο (0, 0.0073) όπου f(0, 0.0073) = 0.00016. Η προσέγγιση του ελαχίστου είναι ικανοποιητική, όμως ο αριθμός των επαναλήψεων είναι πολύ μεγάλος. Αυτό συμβαίνει γιατί ο αλγόριθμος ταλαντώνεται ξανά στον άξονα x_2, πράγμα που οφείλεται στο ακόμη μεγαλύτερο βήμα s_k =15. Ένας απλός τρόπος να αποφύγουμε αυτό το πρόβλημα είναι να μικρύνουμε το s_k ώστε τα νέα σημεία να μην υπερβαίνουν συνεχώς τα όρια.

- S_k =0.1 , γ_k = 0.2 , ακρίβεια ε = 0.01 , σημείο εκκίνησης: (8,-10) (Thema_4.m)





<u>Παρατηρήσεις</u>

Ο αλγόριθμος εκτελεί 157 επαναλήψεις και καταλήγει στο σημείο (0.74 ,0) όπου f(0.74 ,0) = 0.18, λύση που δεν είναι αρκετά ικανοποιητική. Σε αυτή την περίπτωση δεν παρατηρούμε ταλάντωση, και αυτό οφείλεται στη μείωση του βήματος σε s_k=0.1, τα x_{k+1} που είναι εκτός ορίων θα είναι πολύ λιγότερα. Επίσης σε αυτήν την περίπτωση το σημείο εκκίνησης είναι δεν είναι εφικτό σημείο. Το αποτέλεσμα του αλγορίθμου μπορεί να βελτιωθεί μπορούμε να αυξήσουμε την ακρίβεια μειώνοντας το ε.