ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

3^{η} εργασία

«Μέθοδος Μέγιστης Καθόδου με Προβολή»

Τεχνικές Βελτιστοποίησης

στο MATLAB

Δεκέμβριος 2022

2022-2023 | ΑΕΜ: 10265

ΣΤΑΥΡΟΣ ΚΙΗΓΜΑΣ

Φοιτητής ΤΗΜΜΥ ΑΠΘ

ΕΙΣΑΓΩΓΗ

Στην εργασία αυτή το ζητούμενο ήταν η μελέτη της ελαχιστοποίησης μιας δοσμένης συνάρτησης πολλών μεταβλητών $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ακολουθώντας την μέθοδο της *Μέγιστης Καθόδου* αρχικά χωρίς περιορισμούς και στην συνέχεια με περιορισμούς με στόχο την σύγκριση αυτών των δύο περιπτώσεων. Ο αλγόριθμος βασίζεται στην ιδέα της επαναληπτικής καθόδου, σύμφωνα με την οποία ξεκινώντας από κάποιο αρχικό σημείο $x_0 \in \mathbb{R}^n$ παράγουμε διαδοχικά τα διανύσματα $x_1, x_2, \ldots$ έτσι ώστε $f(x_{k+1}) < f(x_k),\ k = 1,2,3, \ldots$

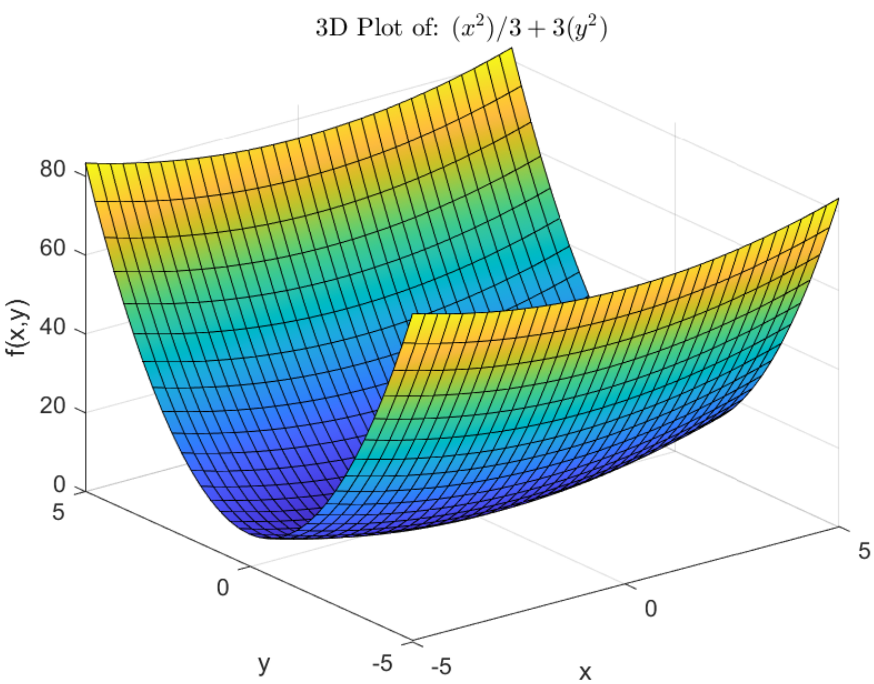
Η αντικειμενική συνάρτηση που ζητείται να χρησιμοποιήσουμε είναι η:

$$f: \mathbb{R}^2 \rightarrow \mathbb{R}, \qquad f(x,y) = \frac{1}{3}x^2 + 3y^2$$

Η υλοποίηση των μεθόδων είναι βασισμένη στο κεφάλαιο 6.2 του βιβλίου *Τεχνικές Βελτιστοποίησης.*

ΘΕΜΑ 0

Σε αυτό το θέμα σχεδιάζουμε την f, έτσι ώστε να έχουμε μια οπτική επαφή με το πρόβλημά μας και με την μορφή της αντικειμενικής συνάρτησης.



Διάγραμμα 1 - Τρισδιάστατη απεικόνιση της αντικειμενικής συνάρτησης

Χρησιμοποιώντας την συνάρτηση `fsurf` (εναλλακτικά την `fmesh`) παρατηρούμε πως υπάρχει ολικό ελάχιστο στο $\{0, 0\}$. Εκεί πρέπει να συγκλίνουν οι μέθοδοι.

ΘΕΜΑ 1

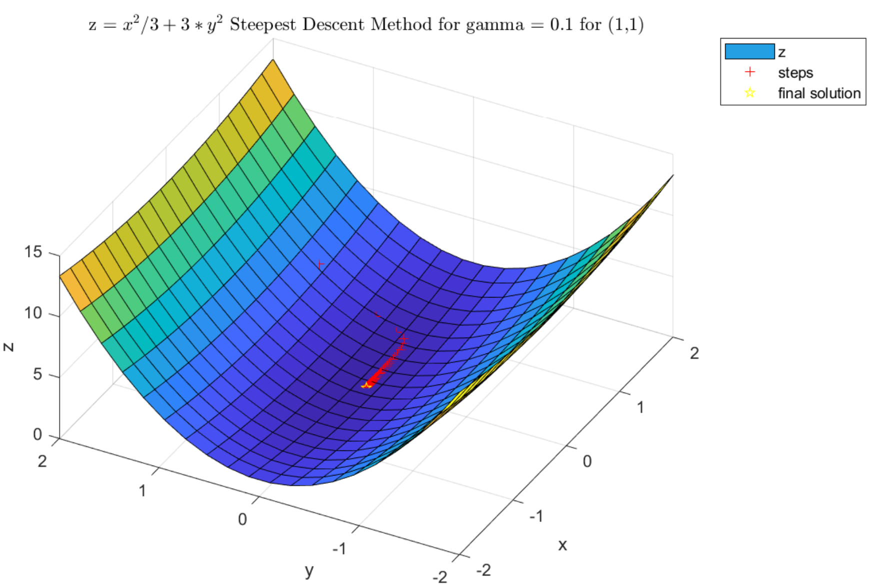
Στο πρώτο θέμα πραγματοποιήθηκε η υλοποίηση της μεθόδου της Μέγιστης Καθόδου χωρίς περιορισμούς. Ζητήθηκε να ελαχιστοποιήσουμε την αντικειμενική συνάρτηση f χρησιμοποιώντας ως αρχικό σημείο (x_0, y_0) κάποιο της επιλογής μας εκτός του $(0,0)$. Προτιμήθηκε το $(1,1)$. Αναλύθηκαν περιπτώσεις διαφορετικών σταθερών βημάτων γ_k. Πιο συγκριμένα:

- ***Περίπτωση i***: Το βήμα επιλέγεται ως 0.1.
- ***Περίπτωση ii***: Το βήμα επιλέγεται ως 0.3.
- ***Περίπτωση iii***: Το βήμα επιλέγεται ως 3.
- ***Περίπτωση iv***: Το βήμα επιλέγεται ως 5.

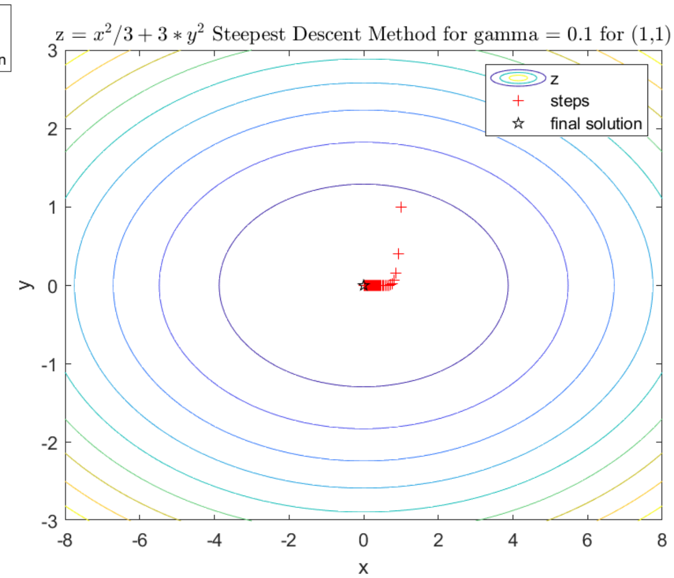
Σχετικά με το ε, επιλέχθηκε ως 0.001, σύμφωνα με την εκφώνηση.

- ## Περίπτωση i: $\gamma_k = 0.1$

Ο αλγόριθμος συγκλίνει στο $(0,0)$ μετά από 96 επαναλήψεις.



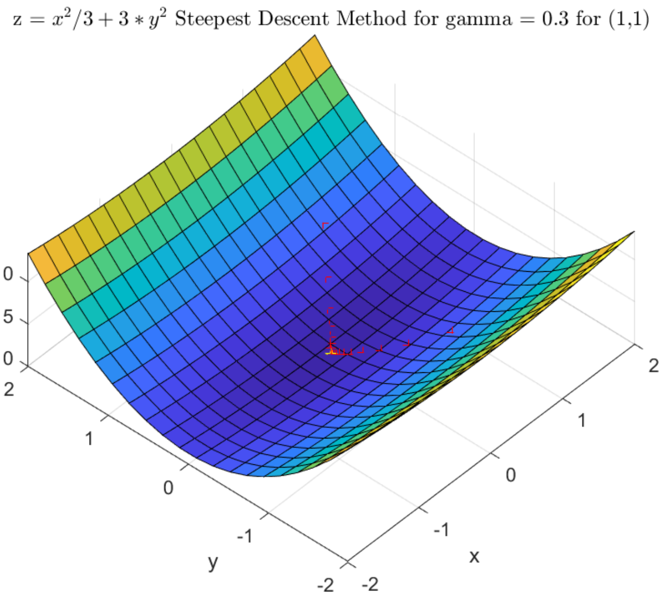
Διάγραμμα 2



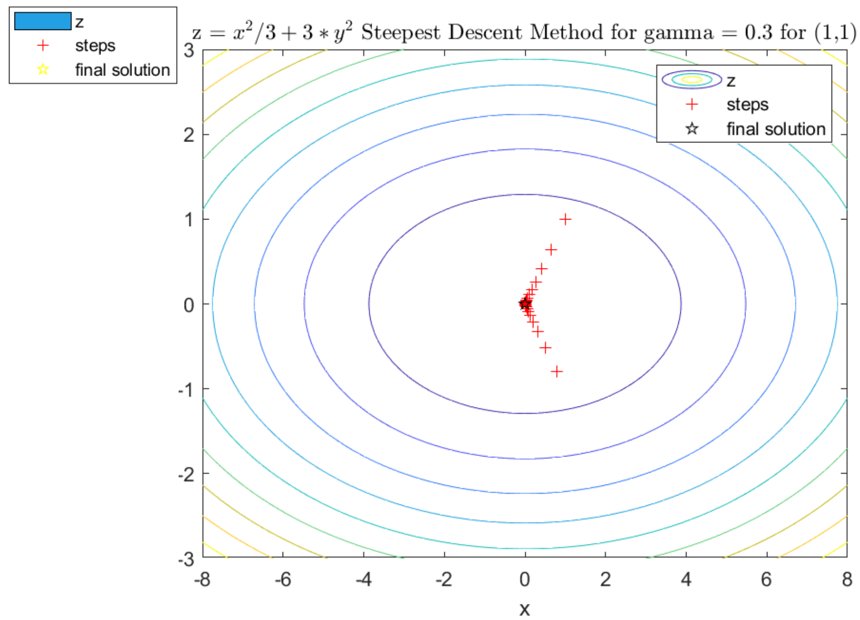
Διάγραμμα 3

- **Περίπτωση ii: $\gamma_k = 0.3$**

Όπως και στην προηγούμενη περίπτωση, ο αλγόριθμος συγκλίνει στο $(0,0)$, όμως τώρα απαιτεί 41 επαναλήψεις.



Διάγραμμα 4



Διάγραμμα 5

- **Περίπτωση iii: $\gamma_k = 3$**

Εδώ εξαιτίας του μεγάλου βήματος, το y απειρίζεται, αλλάζοντας μάλιστα πρόσημο μετά από κάθε επανάληψη. Αντίστοιχα, το x εναλλάσσεται μεταξύ 1 και -1. Το πρόγραμμα σταμάτησε στις 253 επαναλήψεις. Ο αλγόριθμος δεν συγκλίνει.

- **Περίπτωση iv: $\gamma_k = 5$**

Για άλλη μια φορά λόγω του μεγάλου βήματος ο αλγόριθμος δεν συγκλίνει. Το y σε κάθε επανάληψη αλλάζει πρόσημο φτάνοντας στο άπειρο. Το x δε, κατά τον ίδιο τρόπο πλησιάζει πολύ μεγάλες τιμές. Το πρόγραμμα σταμάτησε στις 251 επαναλήψεις.

- **Μαθηματική απόδειξη**

Ζητείται να αναλύσουμε με μαθηματική αυστηρότητα τον λόγο που για τα μικρά γ_k $(0.1\ \kappa\alpha\iota\ 0.3)$ ο αλγόριθμος συγκλίνει, ενώ για τα μεγάλα $(3\ \kappa\alpha\iota\ 5)$ δεν συμβαίνει το ίδιο.

Γενικά ισχύει η σχέση: $x_{k+1} = x_k - \gamma_k \nabla f(x_k)\ (1)$

Γνωρίζοντας τον τύπο της f από τα δεδομένα της εργασίας, υπολογίζω το: $\nabla f(x) = \begin{bmatrix} {2x_1}/{3} \\ 6x_2 \end{bmatrix}$

Τότε από την (1) προκύπτει: $x_{k+1} = x_k - \gamma_k \begin{bmatrix} {2x_{k,1}}/{3} \\ 6x_{k,2} \end{bmatrix}$

Δηλαδή: $\begin{cases} x_{1,k+1} = x_{1,k} - \frac{2}{3}\gamma_k x_{1,k} \\ x_{2,k+1} = x_{2,k} - 6\gamma_k x_{2,k} \end{cases}\ (2)$

$i)\ \Gamma\iota\alpha\ \gamma_k = 0.1\ \kappa\alpha\iota\ \sigma\eta\mu\varepsilon\acute{\iota}o\ \tau o\ (1{,}1){:}\ (2) \Rightarrow \begin{matrix} x_{1,k+1} = \frac{14}{15}x_{1,k} \\ x_{2,k+1} = 0.4x_{2,k} \end{matrix}$

Θα έχουμε σύγκλιση, αλλά για το x_1 θα είναι πιο αργή από ότι για την συντεταγμένη x_2.

Πράγματι, παρατηρώντας τα παραπάνω αποτελέσματα συνέκλινε μετά από 96 επαναλήψεις.

$ii)\ \Gamma\iota\alpha\ \gamma_k = 0.3\ \kappa\alpha\iota\ \sigma\eta\mu\varepsilon\acute{\iota}o\ \tau o\ (1{,}1){:}\ (2) \Rightarrow \begin{matrix} x_{1,k+1} = \frac{4}{5}x_{1,k} \\ x_{2,k+1} = -0.8x_{2,k} \end{matrix}$

Τα x_1 και x_2 συγκλίνουν στο 0 με τον ίδιο ρυθμό, απλά το x_2 αλλαζει προσημο σε κάθε επανάληψη. Συγκρίνοντας με τα παραπάνω αποτελέσματα, συνέβη ακριβώς αυτό και ο αλγόριθμος τερμάτισε μετά από 41 επαναλήψεις.

$iii)\ \Gamma\iota\alpha\ \gamma_k = 3\ \kappa\alpha\iota\ \sigma\eta\mu\varepsilon\acute{\iota}o\ \tau o\ (1{,}1){:}\ (2) \Rightarrow \begin{matrix} x_{1,k+1} = -x_{1,k} \\ x_{2,k+1} = -17x_{2,k} \end{matrix}$

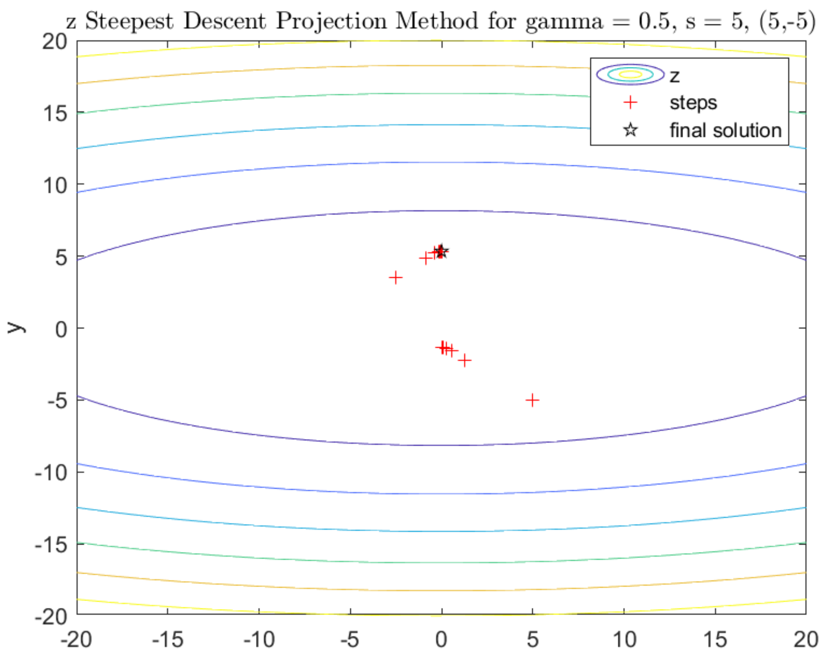
Εδώ περιμένουμε το x_1 να ελλασεται μεταξυ 1 και -1, ενώ το x_2 να απειριστει μετα από κάποιες επαναλήψεις. Βλέποντας τα αποτελέσματα μας, επιβεβαιώνεται αυτό.

$iv)\ \Gamma\iota\alpha\ \gamma_k = 5\ \kappa\alpha\iota\ \sigma\eta\mu\varepsilon\acute{\iota}o\ \tau o\ (1{,}1){:}\ (2) \Rightarrow \begin{matrix} x_{1,k+1} = -\frac{7}{3}x_{1,k} \\ x_{2,k+1} = -29x_{2,k} \end{matrix}$

Σε αυτήν την περίπτωση και οι δυο συντεταγμένες αλλάζουν πρόσημο σε κάθε επανάληψη και εν τέλει καταλήγουν στο άπειρο. Τα πειραματικά αποτελέσματα επαληθεύονται και εδώ.

ΘΕΜΑ 2

Στο δεύτερο θέμα πραγματοποιήθηκε η υλοποίηση της μεθόδου της *Μέγιστης Καθόδου με προβολή*. Πιο συγκεκριμένα με $s_k = 5$, $\gamma_k = 0.5$, σημείο εκκίνησης τo $(5, -5)$ και ακρίβεια $\varepsilon = 0.01$. Αυτή η διαφοροποίηση στην μέθοδο πραγματοποιήθηκε, επειδή πλέον στο πρόβλημά μας εισήχθησαν οι εξής περιορισμοί: $-10 \leq x \leq 5$ και $-8 \leq y \leq 12$.



Διάγραμμα 6

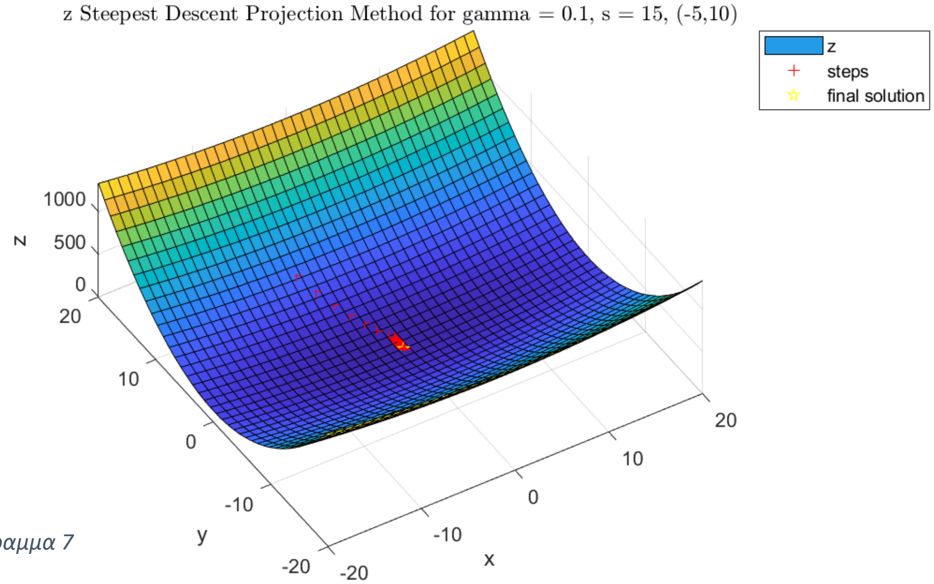
Εδώ, παρατηρούμε εξ αρχής πως το βήμα s_k έχει ίδια τιμή με το βήμα γ_k στην περίπτωση *iv* του *θέματος 1*. Επομένως περιμένουμε να υπάρχει παρόμοια συμπεριφορά του αλγόριθμου και να μην συγκλίνει.

Πράγματι, ο αλγόριθμος δεν συγκλίνει στην επιθυμητή τιμή και παγιδεύεται σε έναν ατέρμονα βρόχο. Μια προτεινόμενη λύση εδώ θα ήταν να μειώνεται το βήμα γ_k, όταν μετά από κάποια επανάληψη δεν παρατηρείται σημαντική διαφορά στις τιμές της αντικειμενικής συνάρτησης. Μικρότερα βήματα θα οδηγούσαν σε καλύτερες επιλογές του επόμενου σημείου.

Μια άλλη λύση είναι να μειωθεί το βήμα s_k.

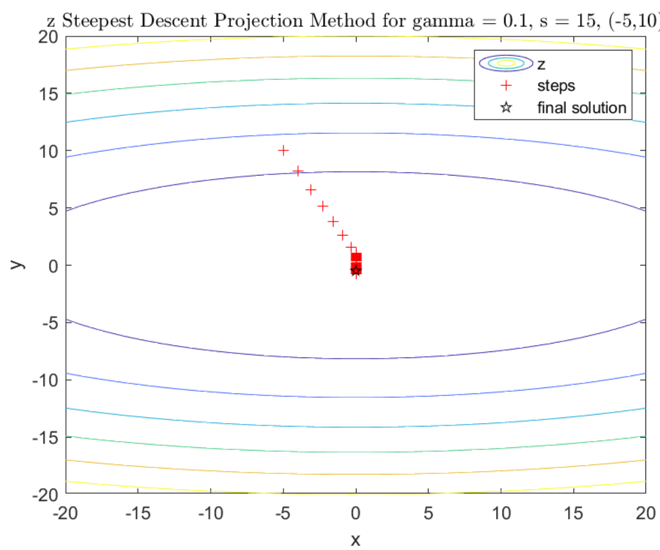
ΘΕΜΑ 3

Στο τρίτο θέμα πραγματοποιήθηκε η υλοποίηση της μεθόδου της *Μέγιστης Καθόδου με προβολή* με $s_k = 15,\ \gamma_k = 0.1$, σημείο εκκίνησης $\tau o\ (-5, 10)$ και ακρίβεια $\varepsilon = 0.01$. Ξανά ισχύουν οι εξής περιορισμοί: $-10 \leq x \leq 5$ και $-8 \leq y \leq 12$.



Διάγραμμα 7

Ο αλγόριθμος συγκλίνει στην επιθυμητή τιμή μετά από 238 επαναλήψεις. Ο αριθμός αυτός είναι τόσο μεγάλος εξαιτίας του μικρού ε που επιλέξαμε.

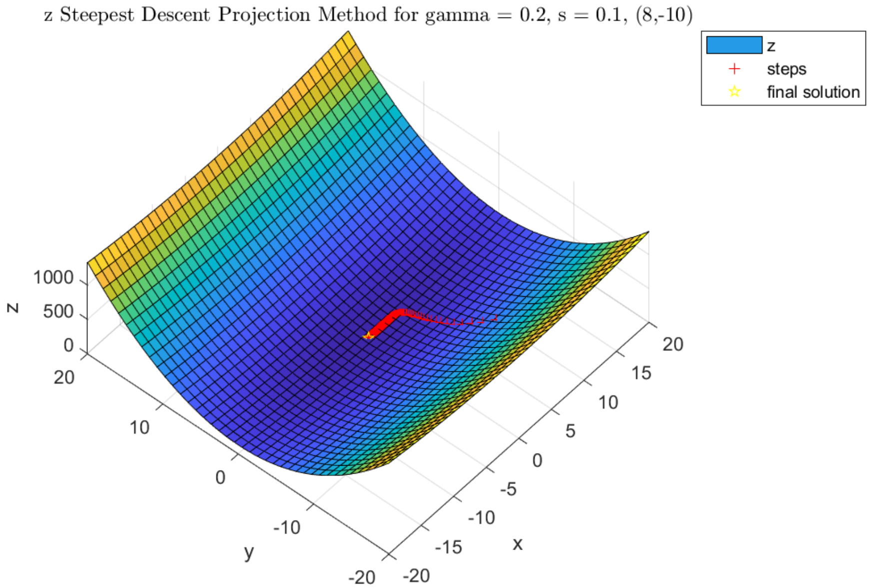


Διάγραμμα 8

Εδώ, περιμέναμε όπως και πριν ο αλγόριθμος να μην συγκλίνει εξαιτίας του μεγάλου s_k. Μάλιστα, τώρα είναι μεγαλύτερο. Απεναντίας, συγκλίνει στο ελάχιστο.

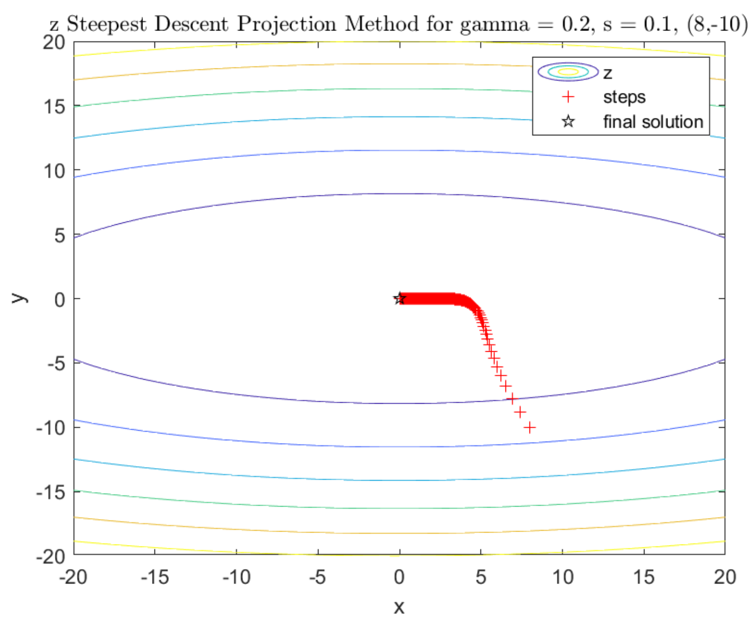
ΘΕΜΑ 4

Στο τέταρτο θέμα πραγματοποιήθηκε η υλοποίηση της μεθόδου της *Μέγιστης Καθόδου με προβολή* με $s_k = 0.1$, $\gamma_k = 0.2$, σημείο εκκίνησης *το* $(8, -10)$ και ακρίβεια $\varepsilon = 0.01$. Ξανά, ισχύουν οι εξής περιορισμοί: $-10 \leq x \leq 5$ και $-8 \leq y \leq 12$.



Διάγραμμα 9

Ο αλγόριθμος συγκλίνει στην επιθυμητή τιμή μετά από 449 επαναλήψεις. Ο αριθμός αυτός είναι τόσο μεγάλος ξανά εξαιτίας του μικρού ε που επιλέξαμε.



Διάγραμμα 10

Εκ των προτέρων περιμέναμε αργή σύγκλιση εξαιτίας του μικρού βήματος $s_k = 0.1$.

ΑΡΧΕΙΑ ΕΡΓΑΣΙΑΣ

Στο συμπιεσμένο φάκελο που ανέβηκε, εκτός από την αναφορά αυτή υπάρχουν τέσσερις φάκελοι συνδεδεμένοι με τα τέσσερα θέματα που ζητήθηκαν.

Στον φάκελο `0. Function Plot` υπάρχει η συνάρτηση functionPlot η οποία αναπαριστά γραφικά την αντικειμενική συνάρτηση f.

Στον φάκελο `1. Steepest Descent Method` περιλαμβάνεται η υλοποίηση της ομώνυμης μεθόδου. Τα αρχεία `steepestDescentConst`, `steepestDescentPlots` περιλαμβάνουν την υλοποίηση της μεθόδου της Μέγιστης Καθόδου χωρίς περιορισμούς και την διαδικασία δημιουργίας των γραφημάτων αντίστοιχα, σύμφωνα με τις αντίστοιχες διαδικασίες του βιβλίου. Επιπλέον, χρειάζεται να υπολογίζουμε το *gradient* της συνάρτησης και έπειτα το διάνυσμα σε κάθε σημείο που βρισκόμαστε και αυτό γίνεται ορίζοντας την `gradientVector`.

Στον φάκελο `2. Steepest Descent Method with Projection` περιλαμβάνεται ο παραπάνω αλγόριθμος με μια τροποποίηση εξαιτίας της ύπαρξης περιορισμών. Συγκεκριμένα κάνει χρήση της προβολής του διανύσματος σε κάθε επανάληψη. Στα `steepestDescentConstProjection`, `steepestDescentProjectionPlots` γίνονται η υλοποίηση της μεθόδου και η παραγωγή γραφημάτων αντίστοιχα.