Τεχνικές Βελτιστοποίησης - 2η Εργαστηριακή Άσκηση

Αριστείδης Δασκαλόπουλος (AEM: 10640)

27 Νοεμβρίου 2024

Γενική Περιγραφή Προβλήματος

Στόχος: *Εύρεση ελαχίστου δοσμένης συνάρτησης $f(x, y)$ χωρίς περιορισμούς με χρήση παραγώγων.*

Πιο συγκεκριμένα, θα υλοποιήσουμε τις μεθόδους:

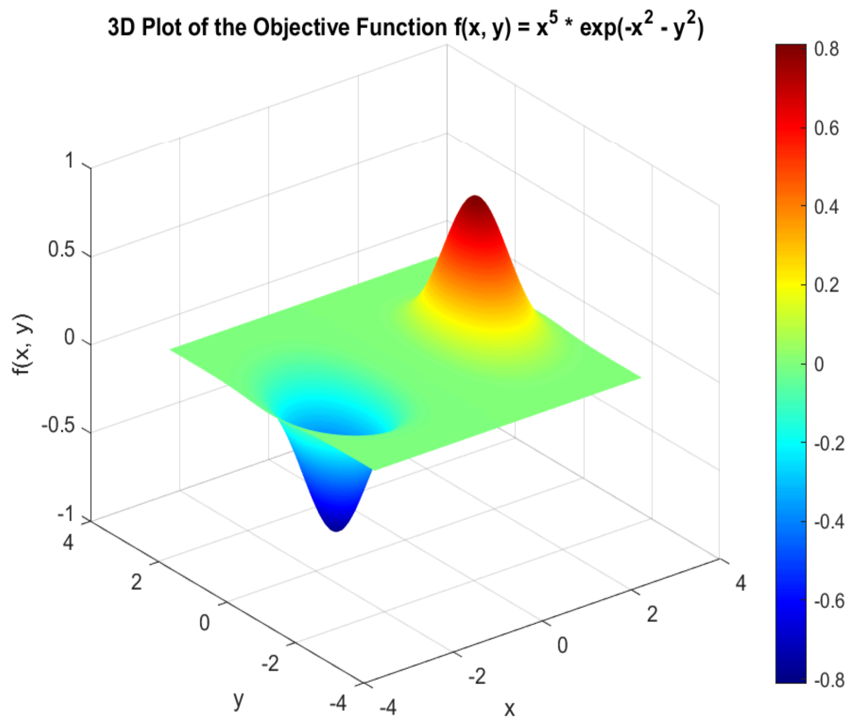
- Μέγιστης Καθόδου (Steepest Descent)
- Newton
- Levenberg-Marquardt

που βασίζονται στην ιδέα της επαναληπτικής καθόδου, κατά την οποία ξεκινώντας από κάποιο σημείο x_0 παράγουμε διαδοχικά διανύσματα x_1, x_2,... τέτοια, ώστε $f(x_{k+1}) < f(x_k) \quad k = 1, 2, \ldots$. Αυτό σημαίνει πως σε κάθε επανάληψη θα πρέπει να βλέπουμε την τιμή της $f(x_k)$ να φθίνει και ιδανικά - αν η μέθοδος που χρησιμοποιούμε για την εύρεση ελαχίστου μπορεί να εφαρμοστεί στην f - να συγκλίνει στο ελάχιστο.

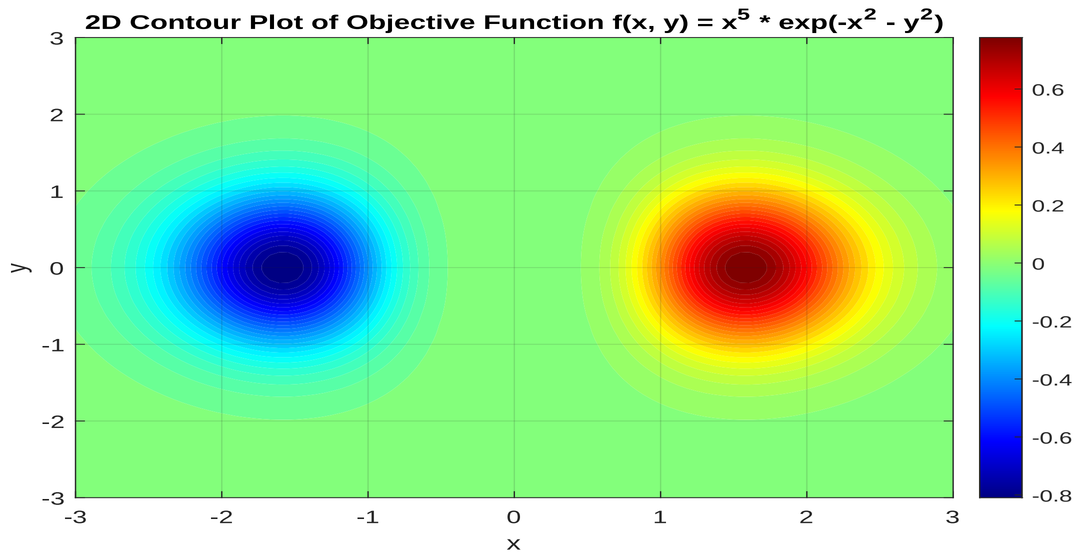
Θέμα 1

Η συνάρτηση που στην συνέχεια της αναφοράς θα μελετήσουμε είναι:

$$f(x, y) = x^5 \cdot e^{-x^2-y^2}$$



Σχήμα 1: 3D$-f(x, y)$



Σχήμα 2: 2D contour $-f(x, y)$

Ο κώδικας για την σχεδίαση των γραφικών παραστάσεων υπάρχει στο αρχείο `matlabCode/mainPlot.m`

Για την συνάρτηση αυτή έχουμε πως το ελάχιστο που ψάχνουμε είναι στο σημείο: $(-\sqrt{\frac{5}{2}}, 0) \simeq (-1.58114, 0)$, στο οποίο η f παίρνει τιμή: $-0.811174...$. Επίσης η f έχει μέγιστο στο $(\sqrt{\frac{5}{2}}, 0) \simeq (1.58114, 0)$ με τιμή $0.811174...$.

Παίρνοντας το gradient για την συνάρτησή μας έχουμε:

$$\nabla f(x,y) = \begin{bmatrix} 5 \cdot x^4 \cdot e^{-x^2-y^2} - 2 \cdot x^6 \cdot e^{-x^2-y^2} \\ -2 \cdot x^5 \cdot y \cdot e^{-x^2-y^2} \end{bmatrix}$$

και λύνοντας το $\nabla f = 0$ έχουμε αποτέλεσμα τα σημεία $(0, a), (1.58114, 0), (-1.58114, 0)$.

Τόσο από την ανάλυση της f όσο και από τα δύο σχήματα με τη γραφική παράστασή της, καταλήγουμε στο συμπέρασμα ότι τα δύο σημεία $(1.58114, 0), (-1.58114, 0)$ είναι τα σημεία όπου υπάρχει το μέγιστο και το ελάχιστο αντίστοιχα, ενώ τα $(0, a)$ αποτελούν σαγματικά σημεία (saddle points) - μιας και σε αυτά ο εσσιανός πίνακας $\nabla^2 f(0, a), \forall a \in \Re$ δεν είναι ούτε θετικά (για να χαρακτηριστούν τοπικά ελάχιστα), ούτε αρνητικά ορισμένος (για να χαρακτηριστούν τοπικά μέγιστα). Η παραπάνω μαθηματική ανάλυση θα μας βοηθήσει να κατανοήσουμε καλύτερα τα αποτελέσματα των αλγορίθμων για διαφορετικά αρχικά σημεία εκκίνησης.

Στην συνέχεια της αναφοράς για καθεμία από τις μεθόδους θα αναλύσουμε περιεκτικά τον τρόπο λειτουργίας της, θα παρουσιάσουμε/εξηγήσουμε τα αποτελέσματα των ερωτημάτων που ζητούνται για διάφορους κανόνες επιλογής βήματος επανάληψης γ_k και τέλος θα αναφέρουμε τα πλεονεκτήματα και μειονεκτήματα που εμφανίζει ο κάθε αλγόριθμος - μέθοδος.

Θέμα 2

Παρουσίαση Μεθόδου Μέγιστης Καθόδου (Steepest Descent)

Σε αυτήν την μέθοδο για την επιλογή του επόμενου σημείου παίρνουμε $x_{k+1} = x_k + \gamma_k \cdot d_k$, με $d_k = -\nabla f(x_k)$. Η επιλογή μας αυτή βασίζεται στο γεγονός ότι το εσωτερικό γινόμενο $\nabla f^T(x_k) \cdot d_k$ γίνεται ελάχιστο όταν $d_k = -\frac{\nabla f(x_k)}{|\nabla f(x_k)|}$. Με κατάλληλη επιλογή του γ_k σε κάθε επανάληψη μπορούμε να υπολογίζουμε το επόμενο σημείο x_{k+1}.

Για την υλοποίηση της μεθόδου ακολουθήσαμε τον αλγόριθμο 5.2.1, σελ.121 του βιβλίου. Οι διαφοροποίησή μας από αυτόν τον αλγόριθμο είναι στον κανόνα με τον οποίο επιλέγεται το γ_k, μιας και στην δική μας υλοποίηση έχουμε τρείς διαφορετικούς τρόπους με τους οποίους κάνουμε την επιλογή, τους οποίους παρουσιάζουμε παρακάτω μαζί με τα αποτελέσματα της σύγκλησης (ή μη) του αλγορίθμου.

Α - Βήμα γ_k σταθερό

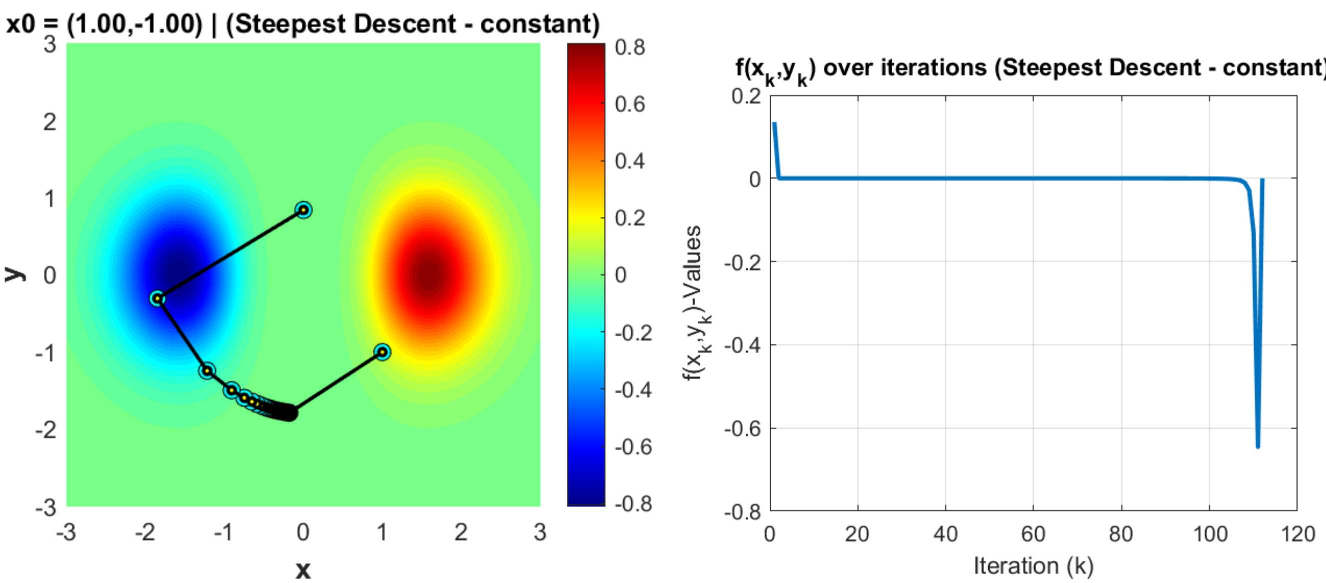
Για την επιλογή του σταθερού βήματος γ_k παρουσιάζεται η εξής δυσκολία: Αν έχουμε πολύ μικρό βήμα τότε η μέθοδός μας θα αργήσει πολύ να φτάσει στο ελάχιστο, με αποτέλεσμα να απαιτηθούν πολλές επαναλήψεις. Αντιθέτως, αν επιλέξουμε μεγάλο γ υπάρχει κίνδυνος να οδηγηθούμε σε αστάθεια μιας και θα είναι δύσκολο για την μέθοδο να προσεγγίσει με την απαιτούμενη ακρίβεια το ελάχιστο. Θέτοντας:

```
epsilon = 1e-4;  % This is the e value to check if grad(F) is near zero.
maxIter = 10000; % If the total number of iterations is more than maxIter, the program stops.
```

παίρνουμε τα παρακάτω αποτελέσματα:

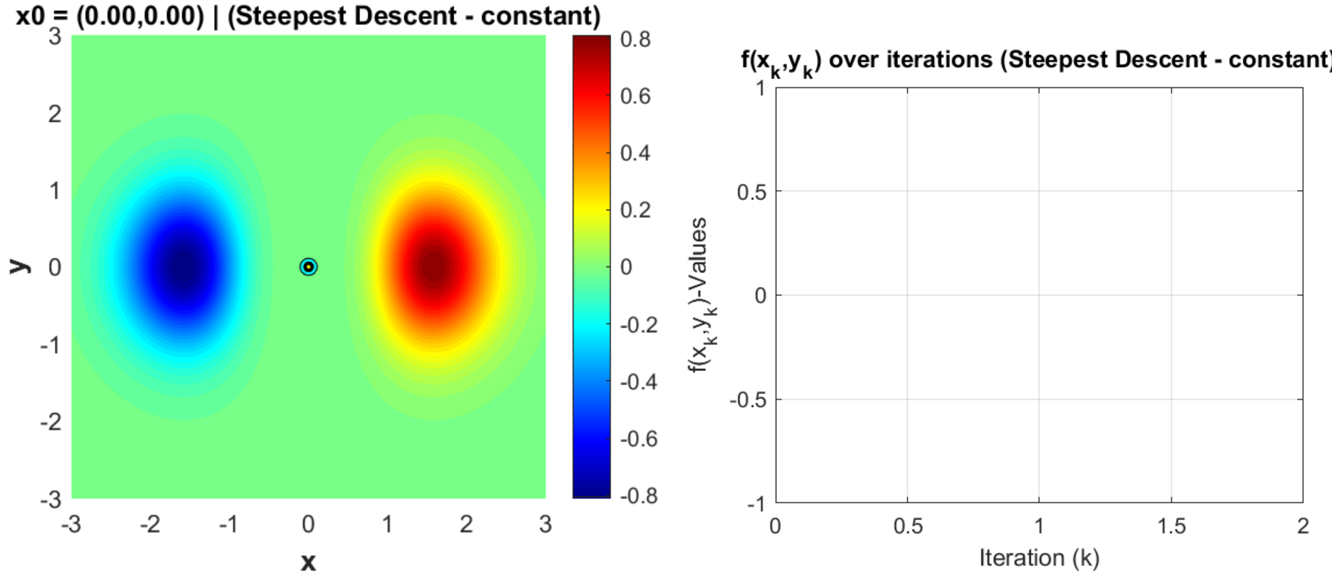
- Για $\gamma = 2.9$:

Βάζοντας μια μεγάλη τιμή στο γ περιμένουμε να υπάρχει αστάθεια στον αλγόριθμο μιας και θα είναι δύσκολο να πετύχουμε ακριβώς το σημείο που ψάχνουμε. Αυτό προκύπτει και από τις γραφικές παραστάσεις παρακάτω:

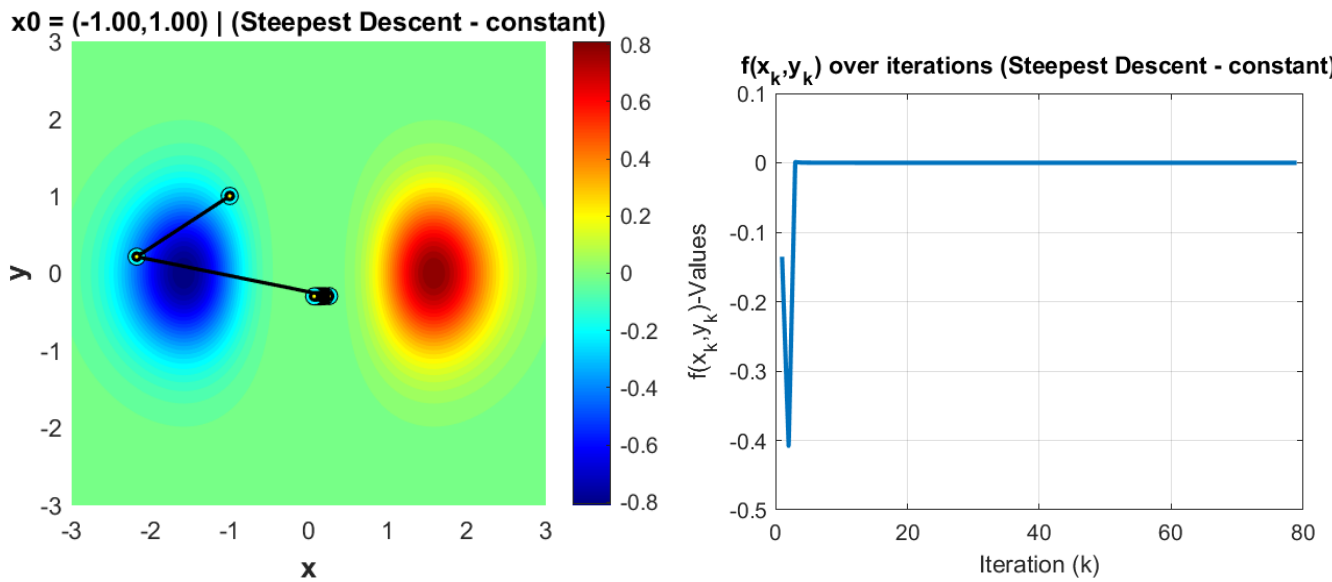


Θα μπορούσαμε να τερματίζουμε τον αλγόριθμο όταν $f(x_k) < f(x_{k+1})$, όμως επιλέξαμε να το αφήσουμε να τρέξει για να δείξουμε ακριβώς αυτήν την αστάθεια, που από τα διαγράμματα είναι εμφανής. Επίσης με κανονικοποίηση της d_k θα είχαμε μια καλύτερη αναπαράσταση για μεγαλύτερες τιμές του γ ομώς αυτό δεν θα συμβάδιζε απόλυτα με τον ζητούμενο αλγόριθμο του βιβλίου.

Ομοίως για τα υπόλοιπα αρχικά σημεία:

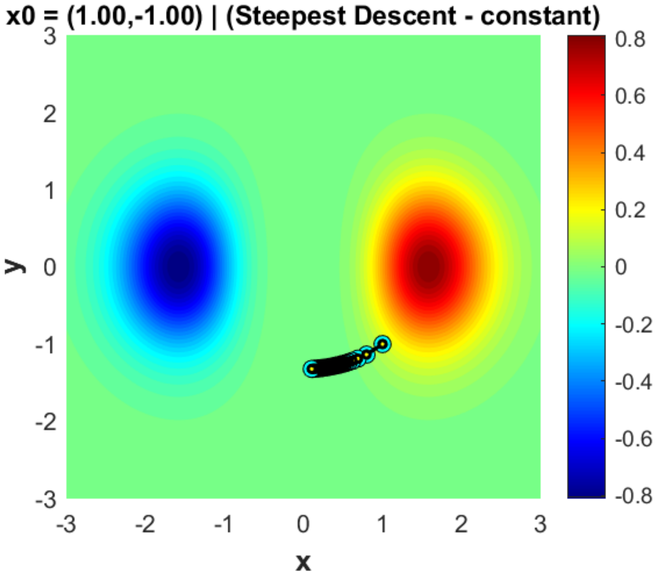


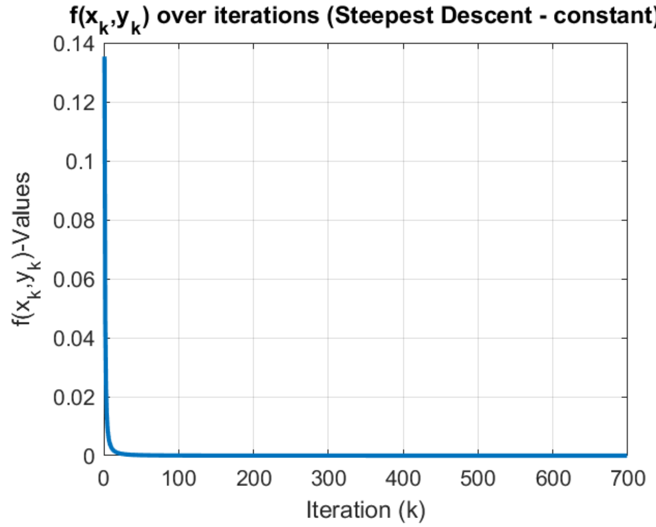
Πάντα όταν αρχίζουμε από το $(0,0)$ μένουμε σε αυτό, μιας και όλα τα σημεία της μορφής $(0,a)$ έχουν μηδενική παράγωγο, επομένως με κανέναν αλγόριθμο και με κανένα κανόνα επιλογής γ δεν θα μπορέσουμε να κινηθούμε από το σημείο $(0,0)$. Για τις υπόλοιπες μεθόδους παρόλο που συμπεριλαμβάνουμε και κάνουμε plot πάντα στον κώδικα και την περίπτωση αρχικού σημείου $(0,0)$, για λόγους συνοχής και μη επανάληψης δεν θα ξαναπαρουσιάσουμε την περίπτωση $(0,0)$.



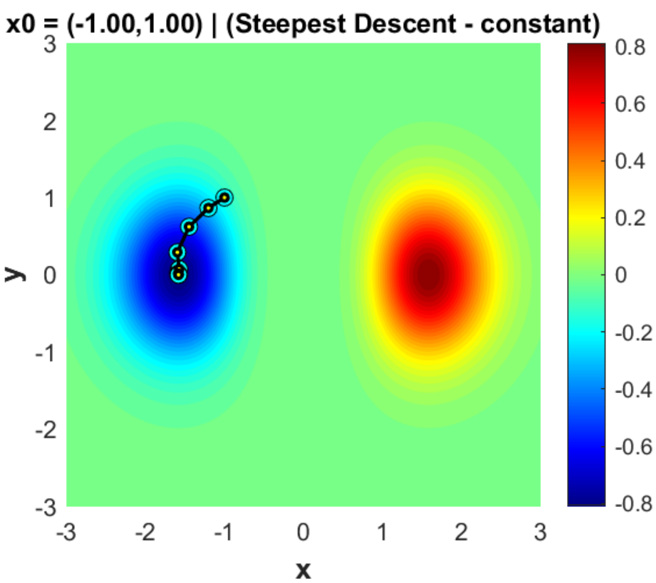
- **Για** $\gamma = 0.5$**:**

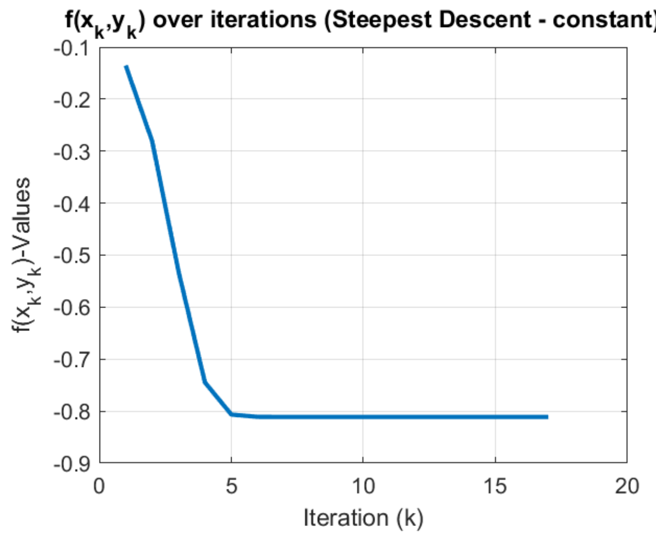
Αντίστοιχα βλέπουμε πως για μικρό γ, ναι μεν δεν υπάρχει περίπτωση αστάθειας του αλγορίθμου, όμως υπάρχει περίπτωση να εγκλωβιστούμε σε σημείο που να μην είναι το ελάχιστο που ψάχνουμε. Τα συμπεράσματα αυτά προκύπτουν και μέσα από τις παραστάσεις:





Πάνω βλέπουμε πως για πολύ μικρό γ δεν έχουμε την δυνατότητα να "δούμε" πέρα από την γραμμή που σχηματίζουν τα σαγματικά σημεία $(0, a)$, με αποτέλεσμα ο αλγόριθμος να εγκλωβίζεται σε κάποιο από αυτά.





Στην πάνω περίπτωση παρατηρούμε ότι όταν είμαστε σχετικά κοντά στο ελάχιστο σημείο που ψάχνουμε και έχουμε ένα μικρό βήμα, μπορούμε (όχι απαραίτητα με τον ελάχιστο αριθμό βημάτων) να καταλήξουμε στο ζητούμενο. Αυτό συμβαίνει επειδή κάθε φορά το διάνυσμα κατεύθυνσης θα δείχνει προς το ελάχιστο και στην διαδρομή που τα x_k θα ακολουθήσουν δεν υπάρχει κάποιο σημείο $(0, a)$ ώστε να εγκλωβιστεί η λύση του αλγορίθμου.

Η επιλογή της παραμέτρου γ, λοιπόν, επηρεάζει τόσο τον αριθμό των επαναλήψεων όσο και τη σύγκλιση του αλγορίθμου στο ελάχιστο. Έχουμε τα εξής:

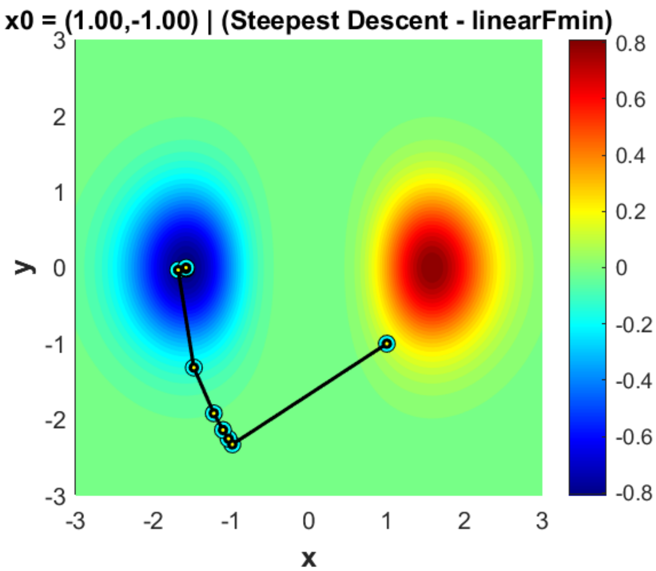
1. Για μικρές τιμές του γ:
   - Όταν επιλέγουμε ως σημείο εκκίνησης το $[-1, 1]$, ο αλγόριθμος συγκλίνει στο ελάχιστο της συνάρτησης. Ωστόσο, με την αύξηση του γ, το βήμα γίνεται ολοένα και μεγαλύτερο, γεγονός που οδηγεί σε σημαντική απόκλιση από το ολικό ελάχιστο και τελικά εγκλωβισμό σε κάποιο άλλο σημείο μηδενικής παραγώγου.
2. Για σημείο εκκίνησης $[1, -1]$:
   - Με μικρή τιμή γ, ο αλγόριθμος συγκλίνει σε ένα σαγματικό σημείο, καθώς η αρχική θέση βρίσκεται μακριά από το ολικό ελάχιστο της συνάρτησης f, όπως αναλύσαμε και παραπάνω.
   - Μεγαλύτερη τιμή γ οδηγεί σε "προσωρινή" προσέγγιση του ολικού ελαχίστου μετά από ορισμένο αριθμό επαναλήψεων. Παρ' όλα αυτά,λόγω της αυξημένης τιμής του βήματος προκαλείται τελικά απόκλιση/αστάθεια.

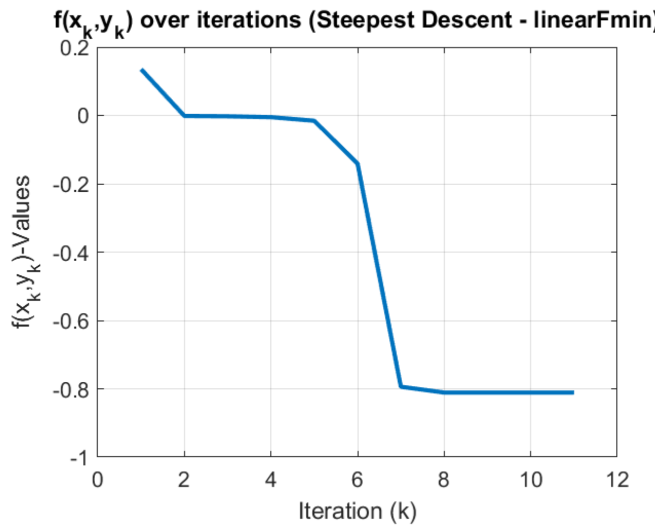
3. Για σημείο εκκίνησης $[0, 0]$: Καθώς το σημείο αυτό έχει μηδενική παράγωγο, ο αλγόριθμος τερματίζει αμέσως από την πρώτη επανάληψη.

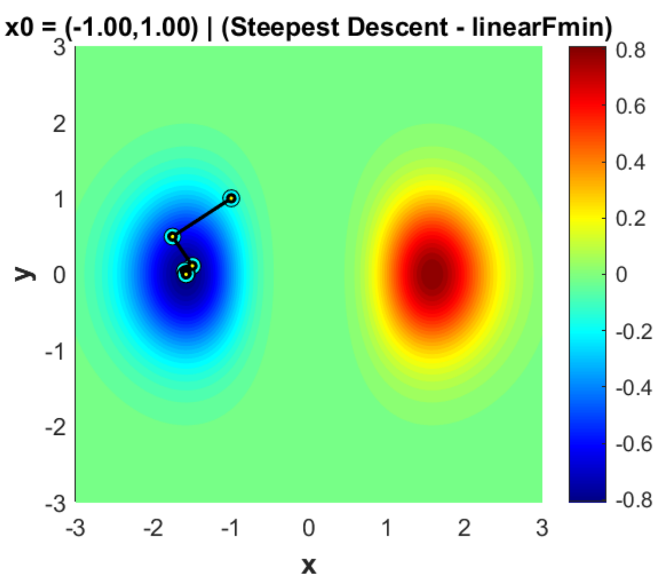
Η παραπάνω ανάλυση υπογραμμίζει τη σημασία της σωστής επιλογής της παραμέτρου γ για τη σύγκλιση του αλγορίθμου και την αποφυγή εγκλωβισμού σε ανεπιθύμητα σημεία. Επομένως, εμφανής είναι η ανάγκη δυναμικής επιλογής του γ σε κάθε επανάληψη, σύμφωνα με κάποιον κανόνα.

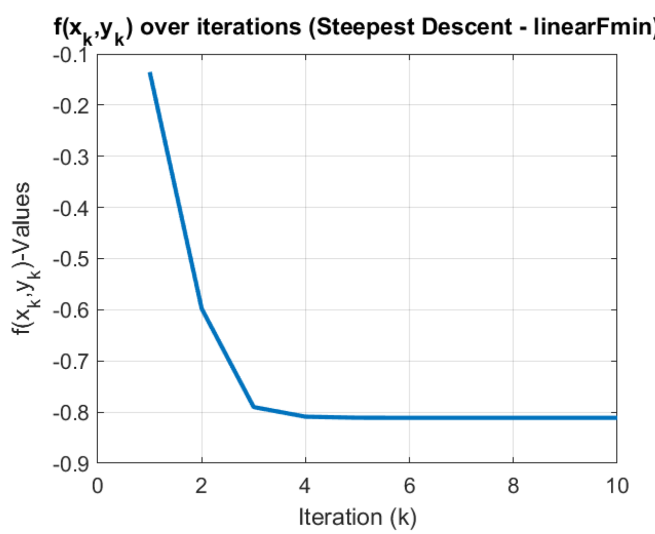
B - Βήμα γ_k τέτοιο ώστε να ελαχιστοποιεί την $f(x_k + \gamma_k \cdot d_k)$

Σε κάθε επανάληψη - αφότου υπολογίσουμε την τιμή του d_k - ψάχνουμε να βρούμε το γ_k για το οποίο ελαχιστοποιείται η $g(\gamma_k) = f(x_k + \gamma_k \cdot d_k)$.Μεθόδους ελαχιστοποίησης έχουμε ήδη δει σε προηγούμενη εργασία, οπότε εδώ το ελάχιστο το βρίσκουμε με την συνάρτηση `fibonacciMethod.m` (Κάνοντας plot τις συναρτήσεις που καλούμαστε να ελαχιστοποιήσουμε ως προς γ είναι εμφανές ότι καλύπτουν τις προϋποθέσεις για την εφαρμογή της μεθόδου Fibonacci). Παρακάτω φαίνονται τα αποτελέσματα του αλγορίθμου με σημεία εκκίνησης τα $(1, -1)$ στην πρώτη γραμμή και $(-1, 1)$ στην δεύτερη. (Η περίπτωση $(0, 0)$ όπως έχουμε ήδη εξηγήσει τερματίζει τον αλγόριθμο από την πρώτη επανάληψη και δεν μετακινεί καθόλου το σημείο από την αρχική του θέση).







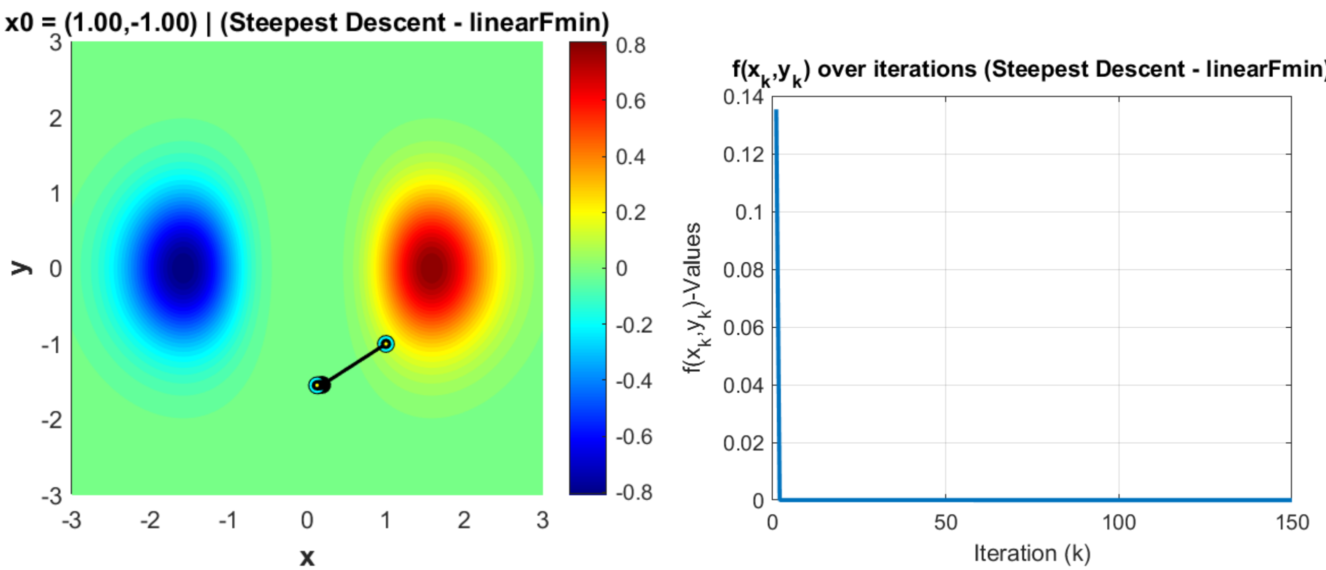


Εδώ όπως είναι φανερό, και με τα δύο αρχικά σημεία ο αλγόριθμος καταλήγει να συγκλίνει στο ολικό ελάχιστο (το συμπέρασμα αυτό υποστηρίζουν και τα δύο διαγράμματα κάθε γραμμής - το δεύτερο διάγραμμα στο οποίο έχουμε αρχικό σημείο πιο μακριά από το ακρότατο βλέπουμε ότι θέλει περισσότερες επαναλήψεις για να φτάσει στο ελάχιστο). Αυτό συμβαίνει επειδή πάντα επιλέγουμε το γ_k εκείνο που μας οδηγεί όλο και πιο κοντά στο ελάχιστο, μιας και στην ευθεία που ορίζει το διάνυσμα d_k παίρνουμε το γ_k που μας δίνει την ελάχιστη τιμή της f - όποτε όπως είναι λογικό στο σημείο x_{k+1} η f θα έχει σίγουρα μικρότερη τιμή.

Στην παραπάνω ανάλυση, το βέλτιστο γ_k το ψάχνουμε στο διάστημα $(0, 10)$. Αλλάζοντας το άνω άκρο αυτού του διαστήματος θα είμαστε σε θέση να βρίσκουμε το γ_k για το οποίο η f παίρνει, στην ευθεία που ψάχνουμε, ακόμα πιο μικρές τιμές βελτιώνοντας έτσι τον αλγόριθμο - καθώς θα πλησιάζουμε στο βέλτιστο πιο γρήγορα (θα επιτρέπουμε μεγαλύτερα βήματα). Έχουμε ήδη βάλει μία αρκούντος μεγάλη τιμή ώστε να έχουμε τα επιθυμητά/βέλτιστα αποτελέσματα (και να έχουμε σύγκληση).

Πρέπει επίσης να προσέξουμε ότι στην περίπτωση που αυτό το άνω ακρό πάρει πολύ μικρή τιμή (πχ 2), δεν θα μπορούμε να φτάσουμε ποτέ στο ολικό ελάχιστο μιας και θα υπάρχει κίνδυνος εγκλωβισμού σε κάποιο τοπικό ακρότατο (ή σε κάποιο σαγματικό σημείο), κυρίως στην περίπτωση όπου το αρχικό μας σημείο είναι το $(1, -1)$ (βρίσκεται δηλαδή στο δεξή ημιεπίπεδο μακριά από το ολικό ελάχιστο).

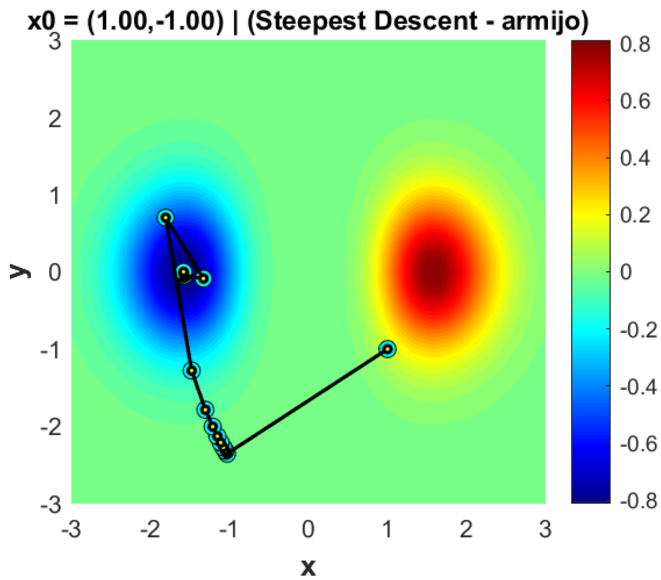
Για να επαληθεύσουμε αυτό τρέχουμε ξανά τον αλγόριθμο για εύρος $\gamma \in (0, 2)$ (η αλλαγή του εύρους γίνεται μέσα στις παραμέτρους της συνάρτησης `fibonacciMethod` στο αρχείο `src/findGamma.m`).

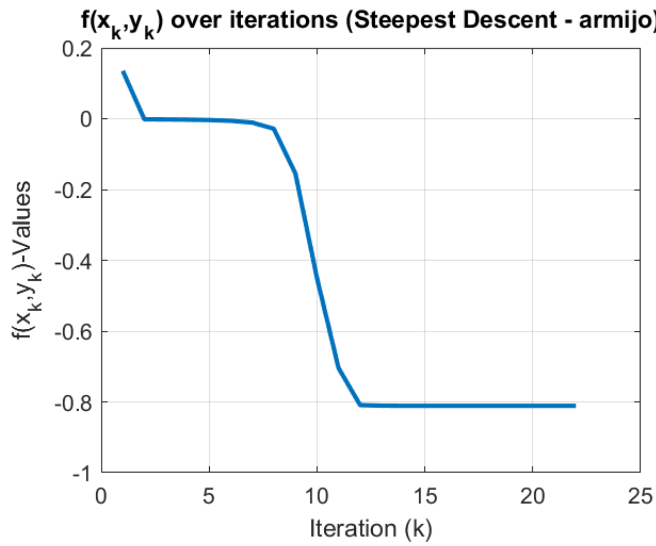


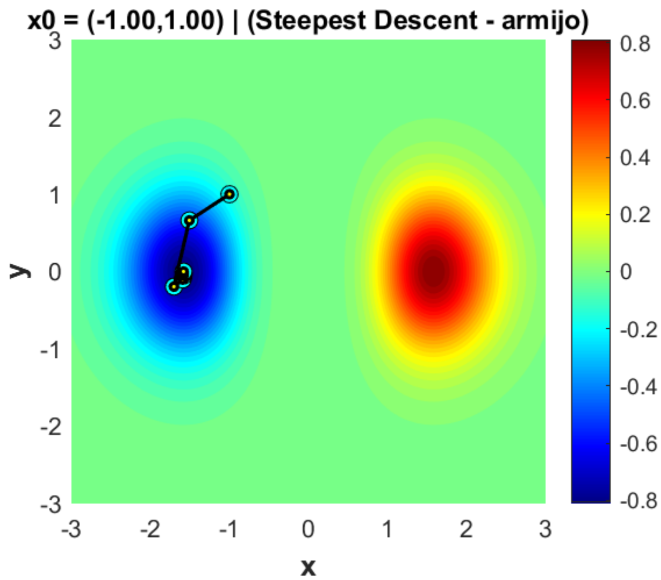
Γ - Βήμα γ_k βάσει του κανόνα Armijo

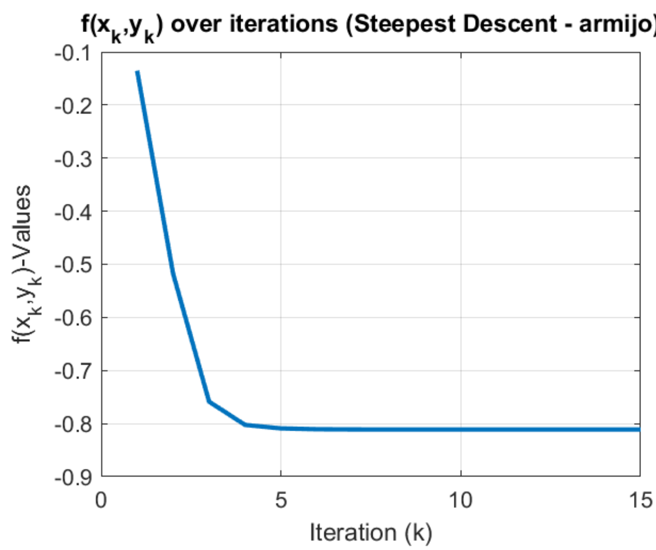
Ο κανόνας Armijo παρουσιάζεται στην σελίδα 140 του βιβλίου. Συνοπτικά, θέτοντας κατάλληλες τιμές α και β ψάχνουμε τον μικρότερο μη-αρνητικό ακέραιο m_k ώστε να ικανοποιείται η συνθήκη 5.2.39. Τότε επιλέγουμε $\gamma_k = s{\cdot}\beta^{m_k}$, όπου s είναι μια πρώτη προσέγγιση (ένα αρχικό βήμα) γ. Το τελικό γ_k είναι τέτοιο, ώστε να ικανοποιείται το κριτήριο 4 και να μπορούμε με μεγαλύτερη ασφάλεια να πούμε πως το αποτέλεσμα συγκλίνει στην τελική τιμή. Φυσικά για κάθε επανάληψη (με τα ίδια α, β και s) θα πρέπει να υπολογίζουμε το καινούριο γ_k.

Τρέχοντας και πάλι τον αλγόριθμο παίρνουμε τα εξής (για $\alpha = 0.01$, $\beta = 0.25$ και $s = 5$):

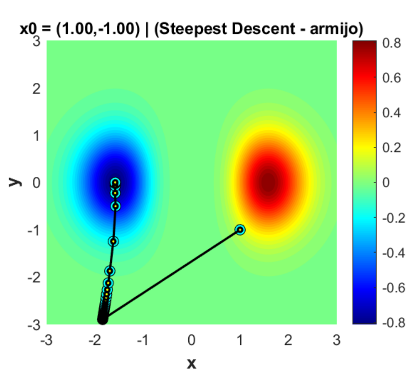


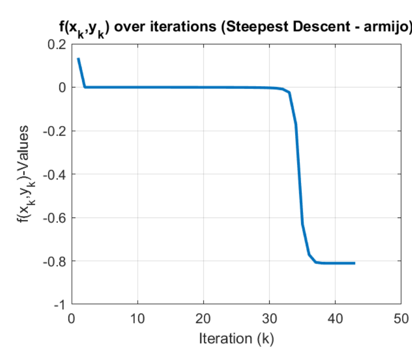


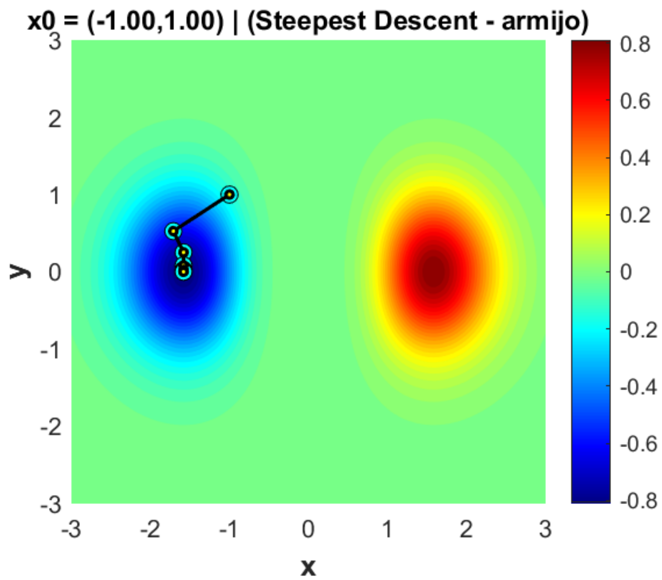


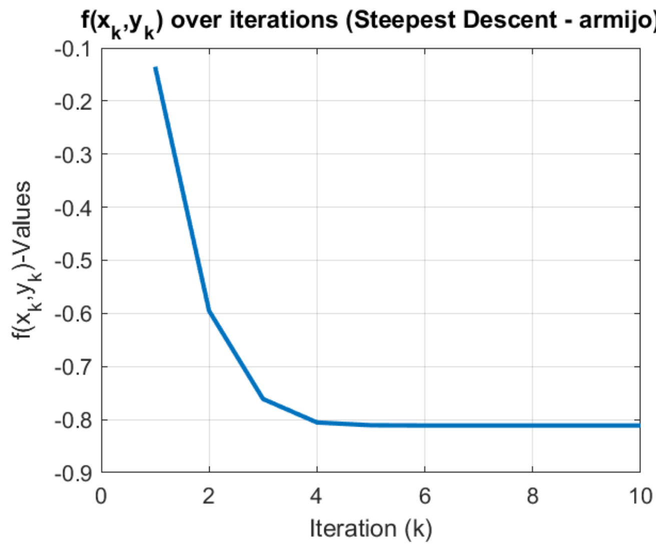


Η επιλογή των α, β και s προφανώς επηρεάζει το πόσο γρήγορα θα συγκλίνει ο αλγόριθμος. Αυξάνοντας για παράδειγμα το s επιτρέπουμε στον αλγόριθμο να βλέπει πιο μακριά και να μπορεί να τοποθετεί τα νέα σημεία σε μεγαλύτερη απόσταση (αυξάνουμε το μέγιστο επιτρεπτό βήμα - με αποτέλεσμα αλλαγή στο πλήθος των επαναλήψεων που απαιτεί ο αλγόριθμος, ανάλογα με το που βρέθηκε το νέο σημείο και τις τιμές των α και β). Παρακάτω θέτουμε $s = 7$ και ξανατρέχουμε τον αλγόριθμο:









Μετά την αλλαγή βλέπουμε πως για το σημείο που είναι μακριά από το ελάχιστο απαιτούνται πλέον περισσότερες επαναλήψεις (συγκλίνοντας και πάλι στην σωστή λύση), ενώ για το σημείο κοντά στο ελάχιστο καταλήγουμε στο σωστό αποτέλεσμα μέσα σε λιγότερες επαναλήψεις (τόσες όσες απαιτούνταν και με τον προηγούμενο κανόνα επιλογής του γ_k).

Σχόλια

Από όλη την ανάλυση της μεθόδου καταλαβαίνουμε πως τόσο το αρχικό σημείο εκκίνησης του αλγορίθμου όσο και ο κανόνας επιλογής του γ_k παίζουν καθοριστικό ρόλο στην σύγκληση της μεθόδου. Συνοπτικά, συμπεραίνουμε πως για σταθερό γ_k δεν είμαστε πάντα σε θέση να κάνουμε τον αλγόριθμο να συγκλίνει (μπορούμε να φτάσουμε κοντά στην λύση αλλά και πάλι υπάρχει αστάθεια). Πρέπει να λαμβάνουμε υπόψιν τόσο την απόσταση του αρχικού σημείου από το ελάχιστο όσο και το αν υπάρχουν ενδιάμεσα άλλα σημεία όπου $\nabla f = 0$ εκτός του ζητούμενου (όπως έχουμε ήδη αναλύσει).

Σχετικά με τους άλλους κανόνες επιλογής του γ_k, βλέπουμε και πάλι εξάρτηση του αποτελέσματος από τις τιμές με τις οποίες θα αρχικοποιήσουμε το πρόβλημα (τι άνω όριο θα βάλουμε στο γ όταν ψάχνουμε το ελάχιστο σε ένα εύρος και τι τιμές θα δώσουμε στα α, β και s για την Armijo). Μπορούμε όμως να πούμε ότι φαίνεται πως ο δεύτερος κανόνας επιλογής στον οποίο επιλέγουμε την τιμή που ελαχιστοποιεί την f στην ευθεία φαίνεται γενικά συγκλίνει ταχύτερα, ενώ ο Amrijo - παρόλο που με μεγαλύτερη ασφάλεια μπορούμε να πούμε ότι οδηγεί σε λύση - υπάρχει περίπτωση να απαιτηθούν περισσότερα βήματα.

Θέμα 3

Παρουσίαση Μεθόδου Newton

Σε αυτήν την μέθοδο για την επιλογή του επόμενου σημείου παίρνουμε $x_{k+1} = x_k + \gamma_k \cdot d_k$, με $d_k = -(\nabla^2 f(x_k))^{-1} \cdot \nabla f(x_k)$. Για να οδηγούμαστε σωστά στην επόμενη τιμή x_{k+1} και να έχουμε έναν λειτουργικό αλγόριθμο που να συγκλίνει στο ελάχιστο πρέπει ο εσσιανός πίνακας $\nabla^2 f(x_k)$ να είναι θετικά ορισμένος (σε κάθε επανάληψη). Τότε με κατάλληλη επιλογή του γ_k σε κάθε επανάληψη, μπορούμε να υπολογίζουμε το επόμενο σημείο x_{k+1}.

Για την υλοποίηση της μεθόδου ακολουθήσαμε τον αλγόριθμο 5.2.2, σελ.126 του βιβλίου. Οι διαφοροποίησή μας από αυτόν τον αλγόριθμο είναι στον κανόνα με τον οποίο επιλέγεται το γ_k, μιας και στην δική μας υλοποίηση έχουμε τρείς διαφορετικούς τρόπους με τους οποίους κάνουμε την επιλογή, τους οποίους παρουσιάσαμε ήδη στην ανάλυση της προηγούμενης μεθόδου.

Για την προϋπόθεση ο εσσιανός πίνακας να είναι θετικά ορισμένος σε καθένα από τα σημεία x_k, έχουμε βάλει έλεγχο ιδιοτιμών. Συγκεκριμένα ελέγχουμε αν όλες οι ιδιοτιμές του πίνακα είναι θετικές και αν όχι *τερματίζουμε* τον αλγόριθμο επιστρέφοντας όσα σημεία x_k κατάφερε να παράξει έως τότε.

Όπως θα δούμε σε καμία εκ των περιπτώσεων που μας ζητούνται (για κανένα από τα αρχικά σημεία που μας δίνονται) δεν έχουμε θετικά ορισμένο εσσιανό.

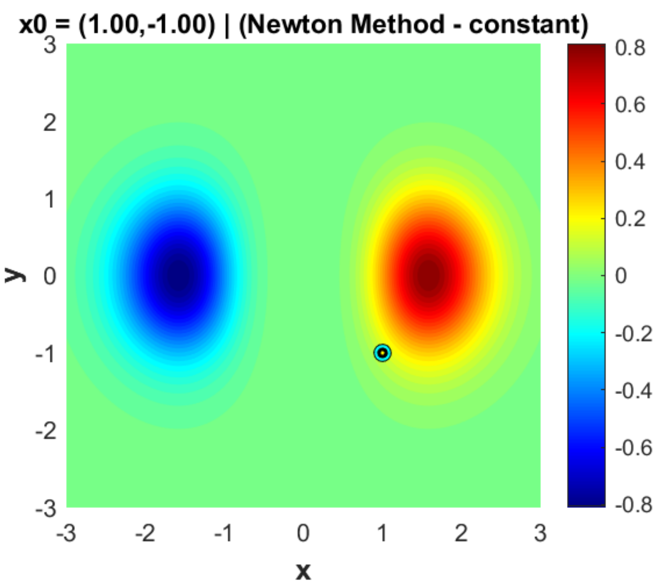
A - Βήμα γ_k σταθερό

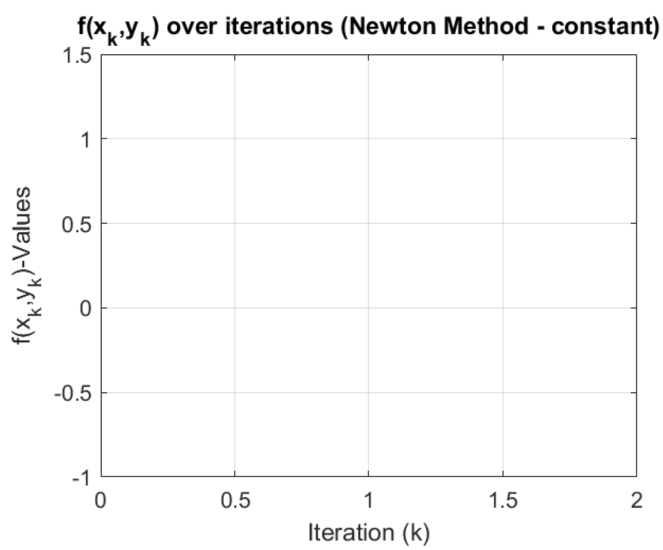
Θέτοντας:

```
epsilon = 1e-4;  % This is the e value to check if grad(F) is near zero.
maxIter = 10000; % If the total number of iterations is more than maxIter, the program stops.
```

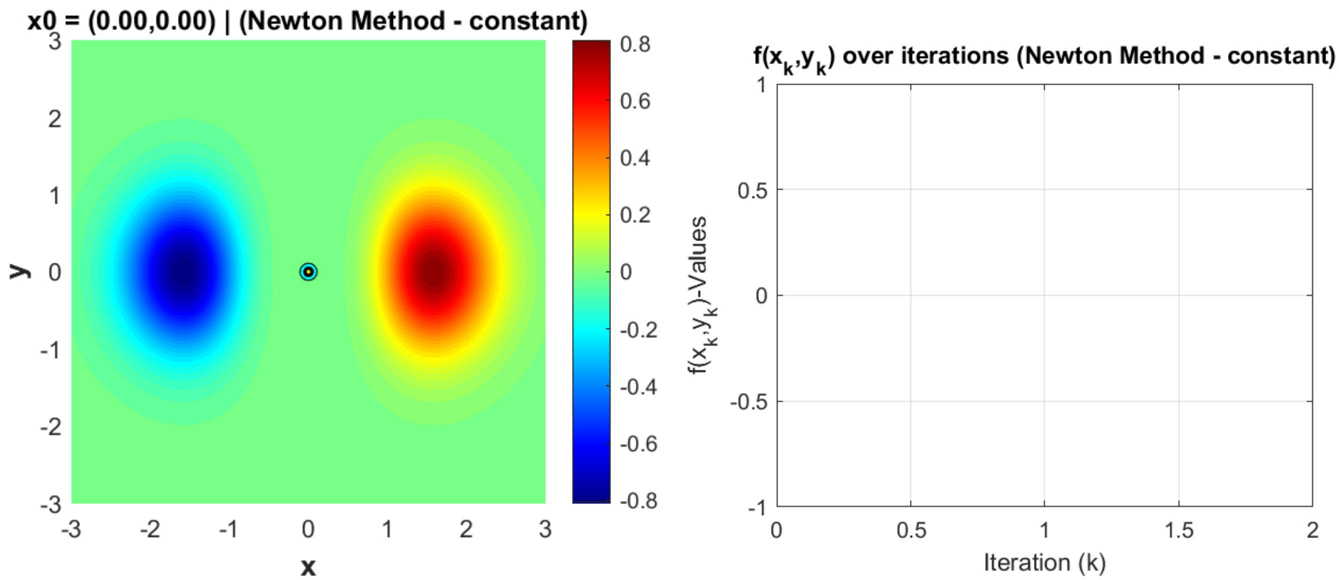
παίρνουμε τα παρακάτω αποτελέσματα (εδώ τρέχουμε τον κώδικα μόνο για $\gamma = 0.5$, μίας και όπως αναφέραμε ήδη για κανένα από τα ζητούμενα αρχικά σημεία δεν τρέχει λειτουργικά ο αλγόριθμος):

- Για $\gamma = 0.5$:

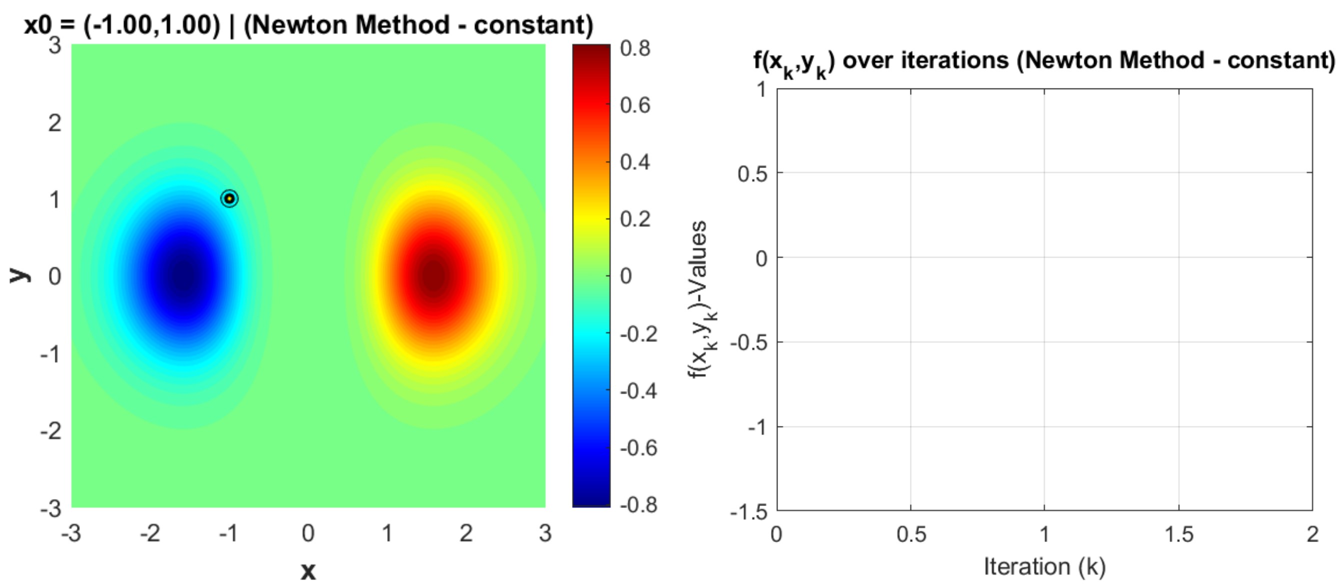




Ομοίως για τα υπόλοιπα αρχικά σημεία:



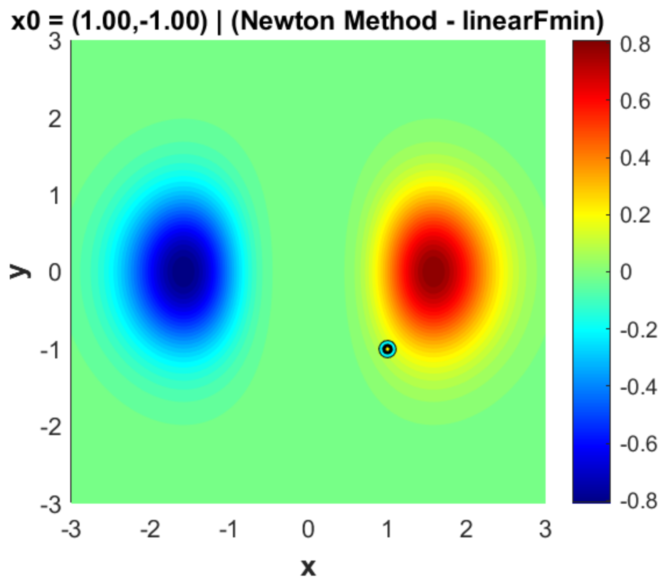
Πάντα όταν αρχίζουμε από το $(0, 0)$ μένουμε σε αυτό, για όλες τις μεθόδους που παρουσιάζουμε.

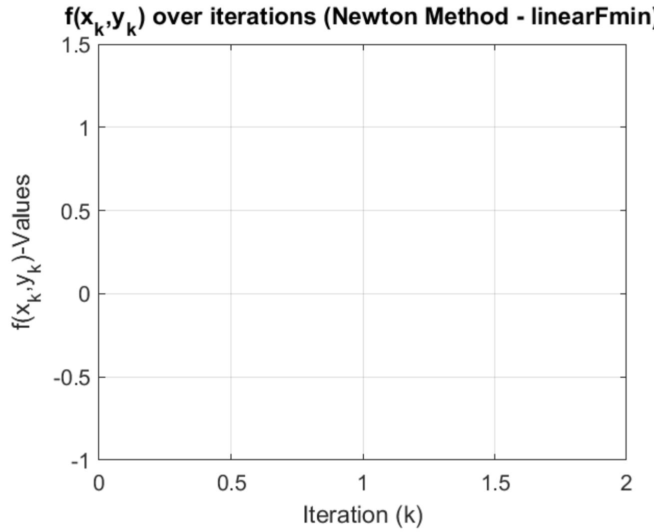


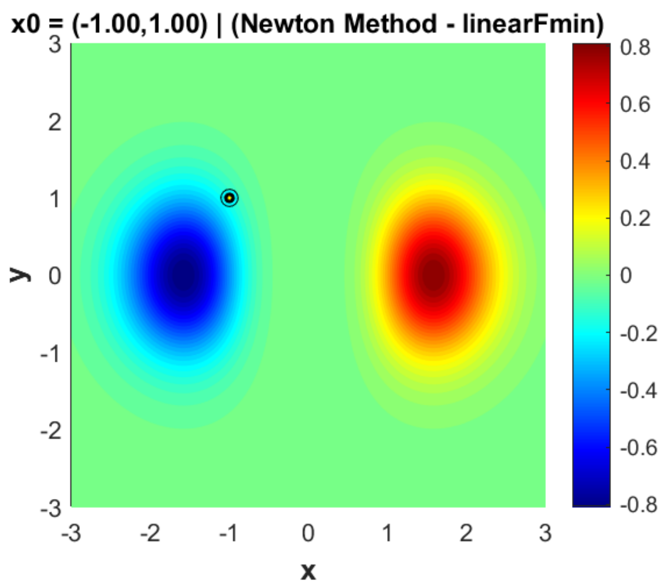
B - Βήμα γ_k τέτοιο ώστε να ελαχιστοποιεί την $f(x_k + \gamma_k \cdot d_k)$

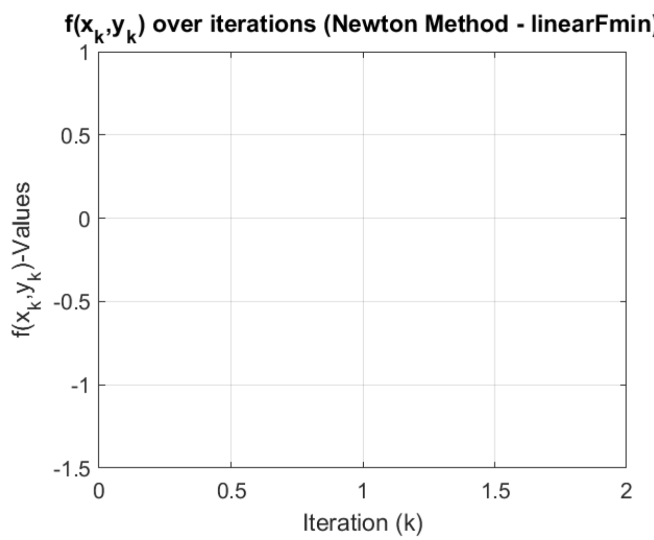
Σε κάθε επανάληψη - αφότου υπολογίσουμε την τιμή του d_k - ψάχνουμε να βρούμε το γ_k για το οποίο ελαχιστοποιείται η $g(\gamma_k) = f(x_k + \gamma_k \cdot d_k)$. Και πάλι όμως, για τον λόγο που αναφέραμε, ο αλγόριθμος δεν τρέχει για καμία επανάληψη.

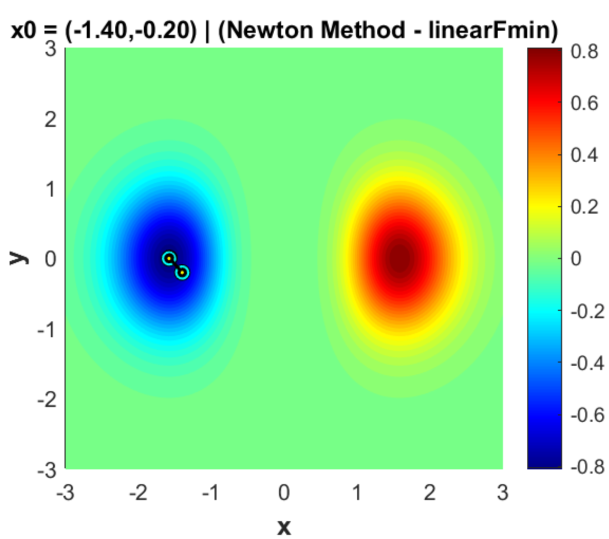
Για να βεβαιωθούμε πως τουλάχιστον η υλοποίηση μπορεί να συγκλίνει με κατάλληλη επιλογή αρχικού σημείου, τρέχουμε τον κώδικα και με σημείο εκκίνησης το $(-1.4, -0.2)$.

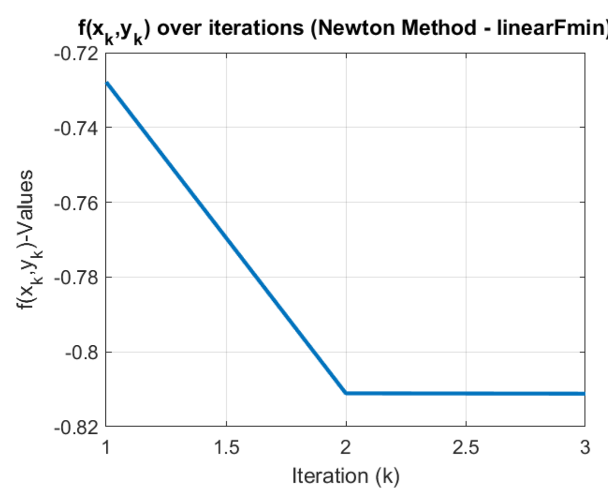

x0 = (1.00,-1.00) | (Newton Method - linearFmin)
y
x


f(x_k,y_k) over iterations (Newton Method - linearFmin)
f(x_k,y_k)-Values
Iteration (k)


x0 = (-1.00,1.00) | (Newton Method - linearFmin)
y
x

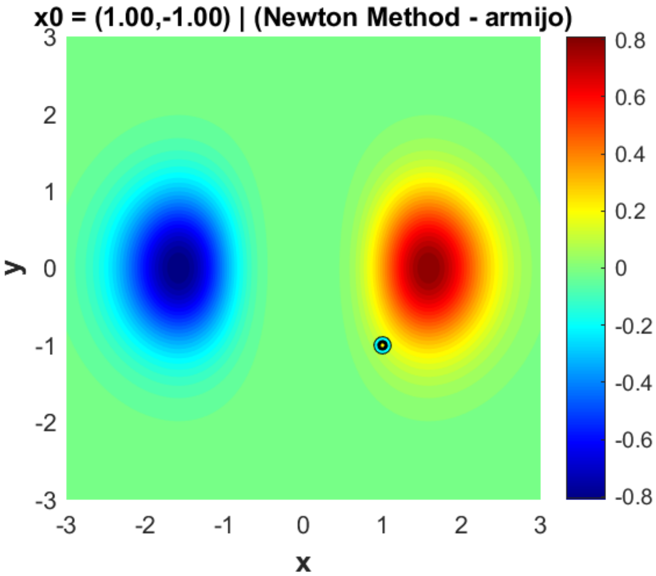

f(x_k,y_k) over iterations (Newton Method - linearFmin)
f(x_k,y_k)-Values
Iteration (k)

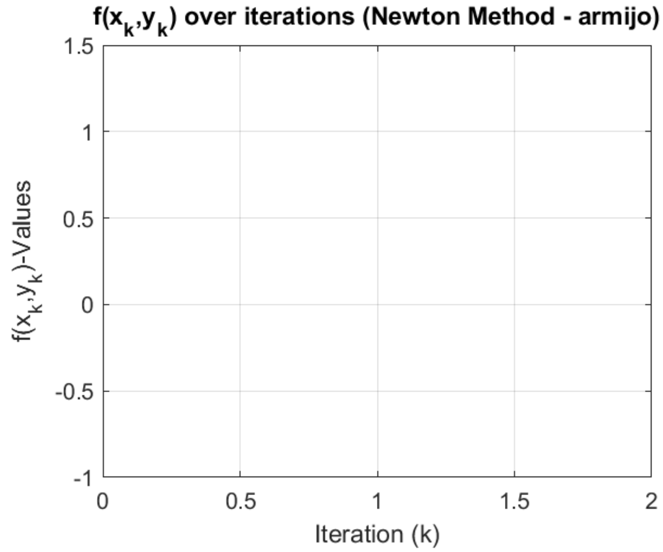

x0 = (-1.40,-0.20) | (Newton Method - linearFmin)
y
x

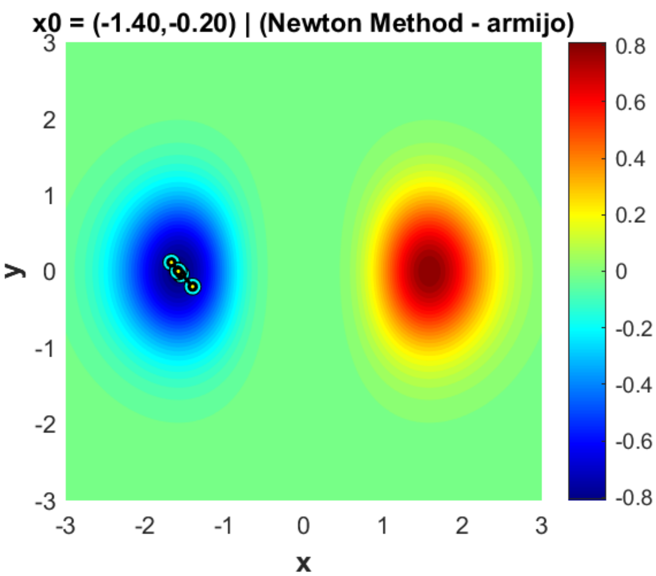

f(x_k,y_k) over iterations (Newton Method - linearFmin)
f(x_k,y_k)-Values
Iteration (k)

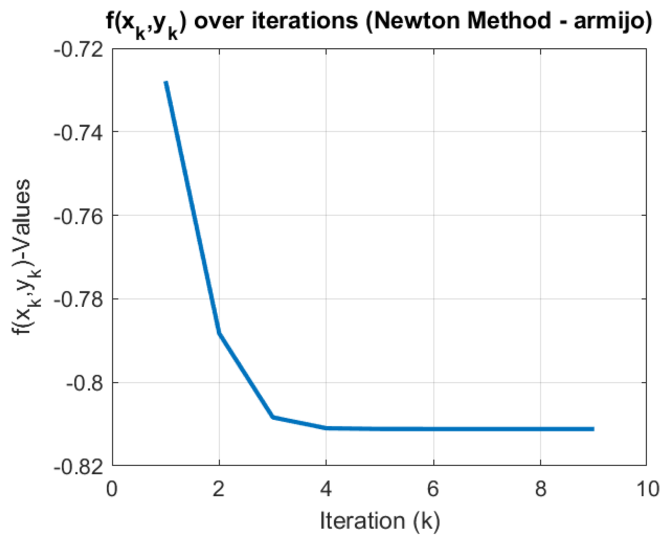
Γ - Βήμα γ_k βάσει του κανόνα Armijo

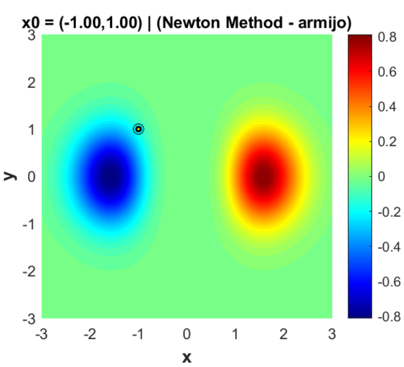
Τρέχοντας και πάλι τον αλγόριθμο (για $\alpha = 0.01$, $\beta = 0.25$ και $s = 5$). Δεν τρέχουμε τον αλγόριθμο για το $(0, 0)$ αλλά για το δικό μας σημείο, ώστε και πάλι να δούμε ότι τουλάχιστον ο κώδικας αλγοριθμικά λειτουργεί όταν ισχύει η συνθήκη σωστής λειτουργίας που αναφέραμε.

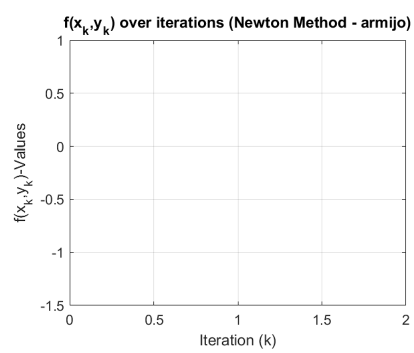












Η επιλογή των α, β και s προφανώς και πάλι επηρεάζει το πόσο γρήγορα θα συγκλίνει ο αλγόριθμος (όταν τρέχει σωστά).

Σχόλια

Από όλη την ανάλυση της μεθόδου καταλαβαίνουμε πως το αρχικό σημείο εκκίνησης του αλγορίθμου παίζει καθοριστικό ρόλο στην σύγκληση της μεθόδου, αλλά και στο αν η μέθοδος μπορεί να εφαρμοστεί (επειδή ο εσσιανός μπορεί να μην ορίζεται στο x_0 θετικά).

Για το σημείο μας, στο οποίο η μέθοδος τρέχει, βλέπουμε και πάλι πως ο Armijo είναι πιο αργός κανόνας για την επιλογή γ_k (με αυτόν η μέθοδος θέλει περισσότερες επαναλήψεις για να φτάσει στο τελικό σημείο/ελάχιστο) από τον δεύτερο κατά σειρά. Όμως, όπως έχουμε ήδη πει, το συμπέρασμα αυτό το βγάλαμε για τις δεδομένες τιμές των α, β και s.

Θέμα 4

Παρουσίαση Μεθόδου Levenberg-Marquardt

Σε αυτήν την μέθοδο για την επιλογή του επόμενου σημείου παίρνουμε $x_{k+1} = x_k + \gamma_k \cdot d_k$, με d_k: την λύση του συστήματος $(\nabla^2 f(x_k) + \mu_k \cdot I) \cdot d_k = -\nabla f(x_k)$. Θέλουμε μ_k τέτοιο, ώστε ο πίνακας $\nabla^2 f(x_k) + \mu_k \cdot I$ να είναι θετικά ορισμένος.

Για την υλοποίηση της μεθόδου ακολουθήσαμε τον αλγόριθμο 5.2.3, σελ.139 του βιβλίου.

A - Βήμα γ_k σταθερό

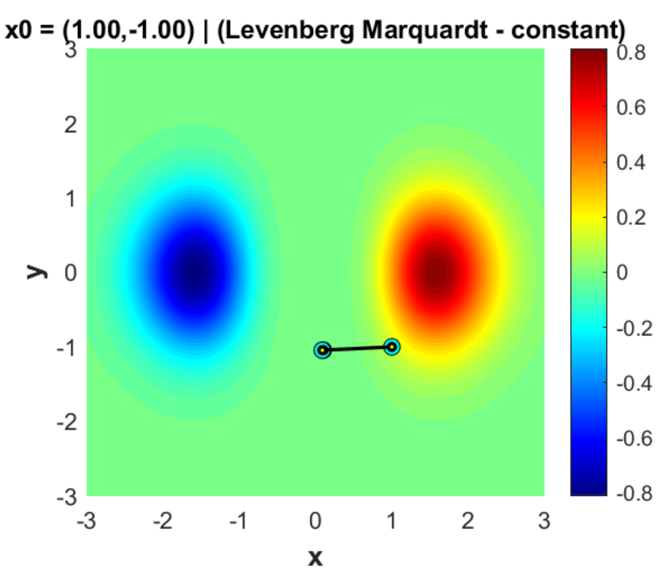
Για την επιλογή του σταθερού βήματος γ_k παρουσιάζεται η εξής δυσκολία: Αν έχουμε πολύ μικρό βήμα τότε η μέθοδός μας θα αργήσει πολύ να φτάσει στο ελάχιστο, με αποτέλεσμα να απαιτηθούν πολλές επαναλήψεις. Αντιθέτως, αν επιλέξουμε μεγάλο γ υπάρχει κίνδυνος να οδηγηθούμε σε αστάθεια μιας και θα είναι δύσκολο για την μέθοδο να προσεγγίσει με την απαιτούμενη ακρίβεια το ελάχιστο. Θέτοντας:

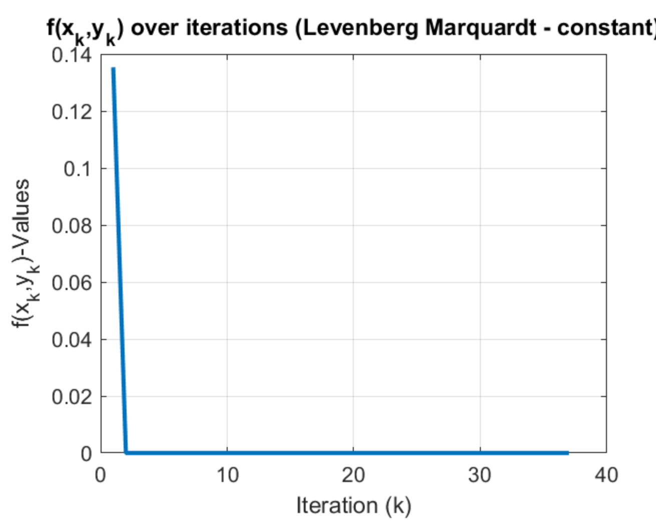
```
epsilon = 1e-4;  % This is the e value to check if grad(F) is near zero.
maxIter = 10000; % If the total number of iterations is more than maxIter, the program stops.
```

παίρνουμε τα παρακάτω αποτελέσματα:

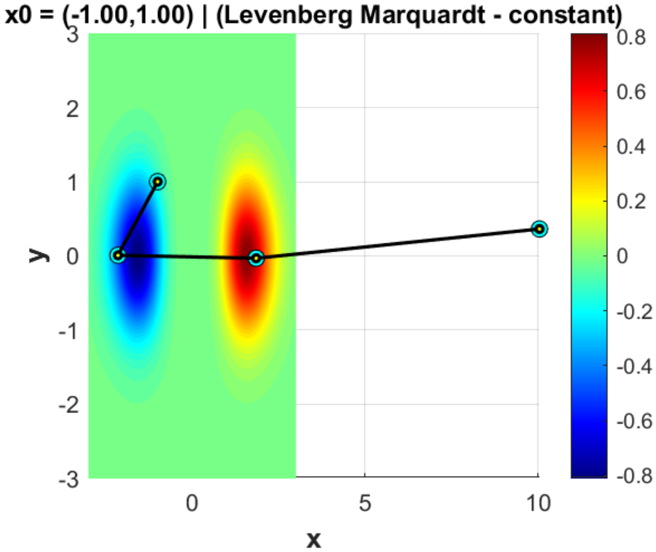
- Για $\gamma = 2.9$:

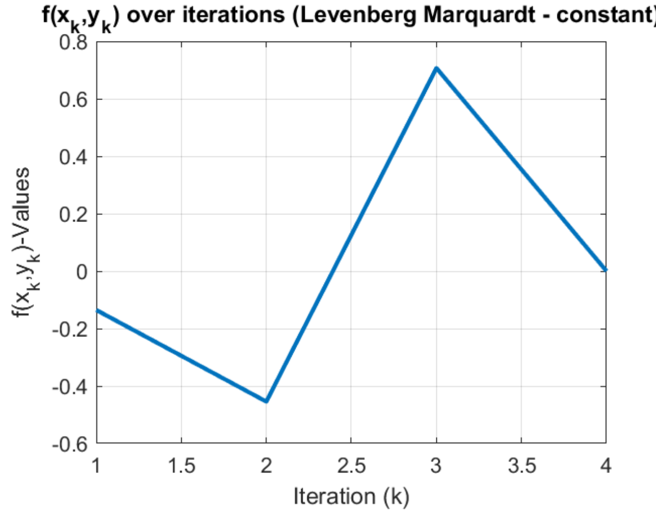
Βάζοντας μια μεγάλη τιμή στο γ περιμένουμε να υπάρχει αστάθεια στον αλγόριθμο μιας και θα είναι δύσκολο να πετύχουμε ακριβώς το σημείο που ψάχνουμε. Αυτό προκύπτει και από τις γραφικές παραστάσεις παρακάτω:





Ομοίως για τα υπόλοιπα αρχικά σημεία:

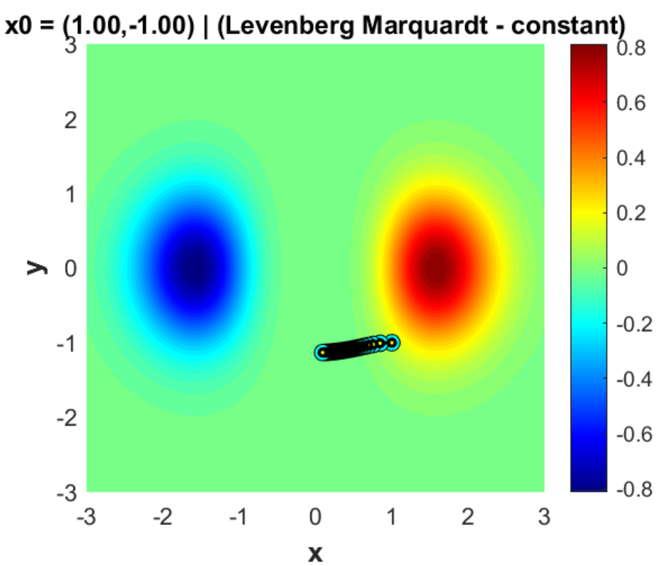


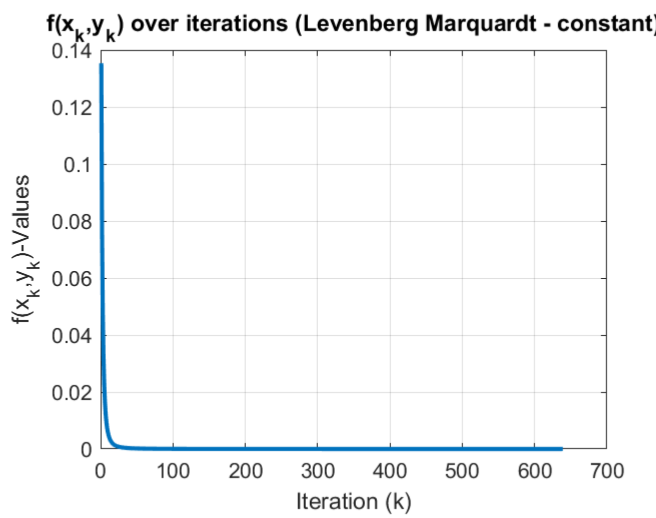


Θα μπορούσαμε να τερματίζουμε τον αλγόριθμο όταν $f(x_k) < f(x_{k+1})$, όμως επιλέξαμε να το αφήσουμε να τρέξει για να δείξουμε ακριβώς αυτήν την αστάθεια, που από τα διαγράμματα είναι εμφανής.

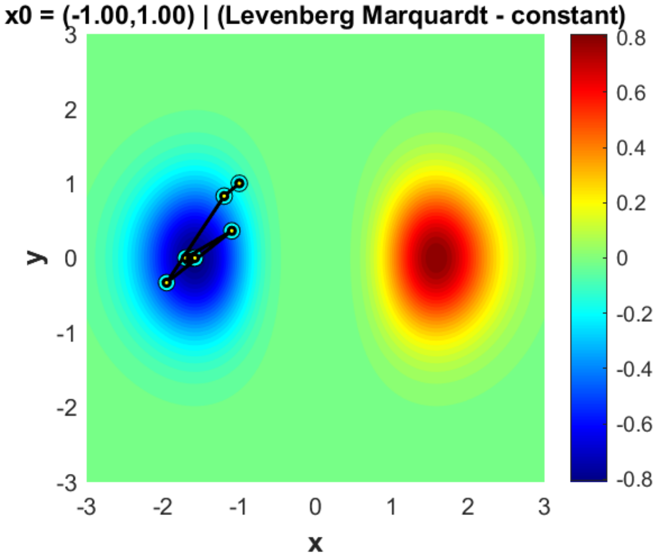
- Για $\gamma = 0.5$**:**

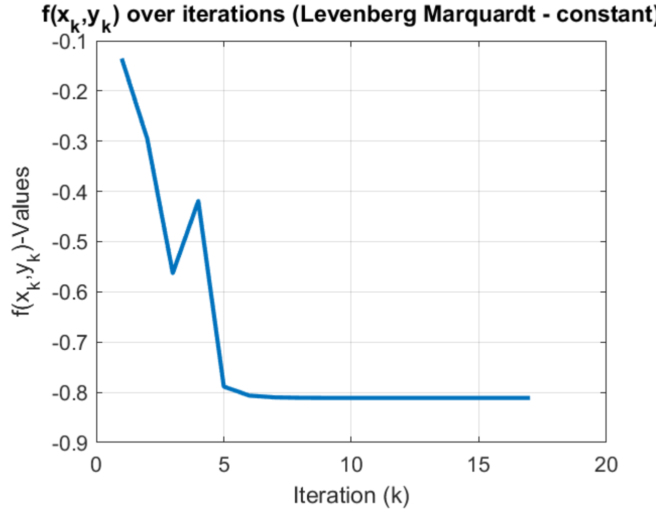
Αντίστοιχα βλέπουμε για μικρό γ:





Πάνω βλέπουμε πως για πολύ μικρό γ δεν έχουμε την δυνατότητα να "δούμε" πέρα από την γραμμή που σχηματίζουν τα σαγματικά σημεία $(0, a)$, με αποτέλεσμα ο αλγόριθμος να εγκλωβίζεται σε κάποιο από αυτά.



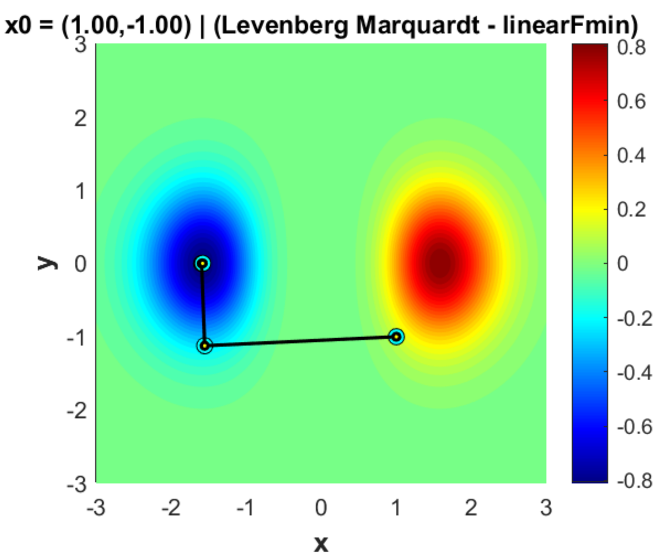


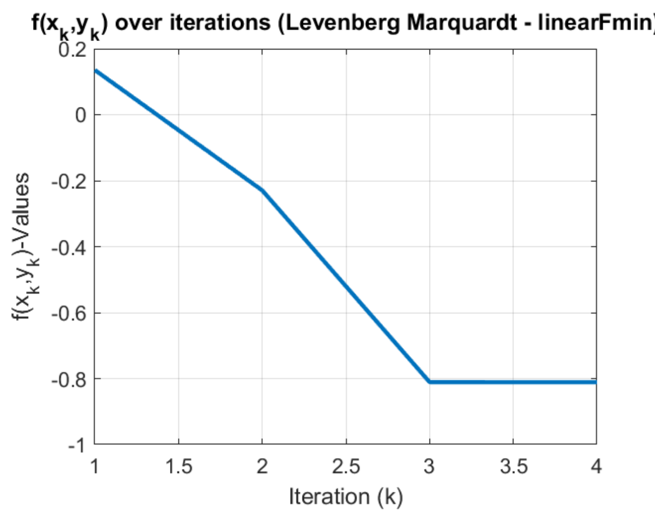
Στην πάνω περίπτωση παρατηρούμε ότι όταν είμαστε σχετικά κοντά στο ελάχιστο σημείο που ψάχνουμε και έχουμε ένα μικρό βήμα, μπορούμε (όχι απαραίτητα με τον ελάχιστο αριθμό βημάτων) να καταλήξουμε στο ζητούμενο. Αυτό συμβαίνει επειδή κάθε φορά το διάνυσμα κατεύθυνσης θα δείχνει προς το ελάχιστο και στην διαδρομή που τα x_k θα ακολουθήσουν δεν υπάρχει κάποιο σημείο $(0, a)$ ώστε να εγκλωβιστεί η λύση του αλγορίθμου.

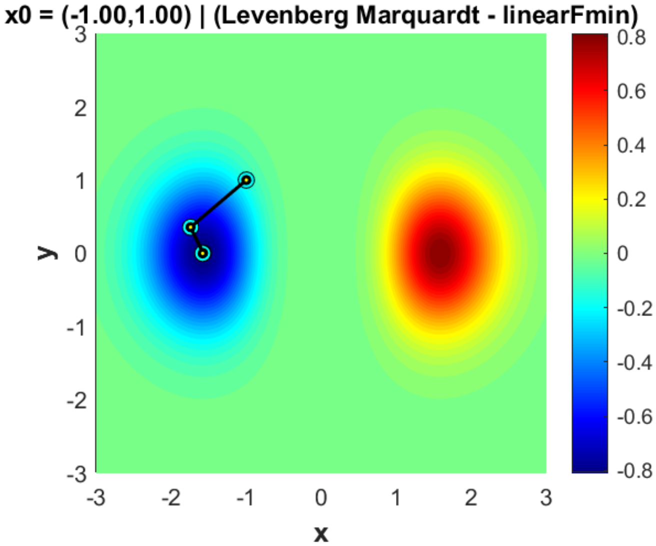
Η επιλογή της παραμέτρου γ, λοιπόν, όπως έχουμε ήδη πει επιρεάζει την λύση.

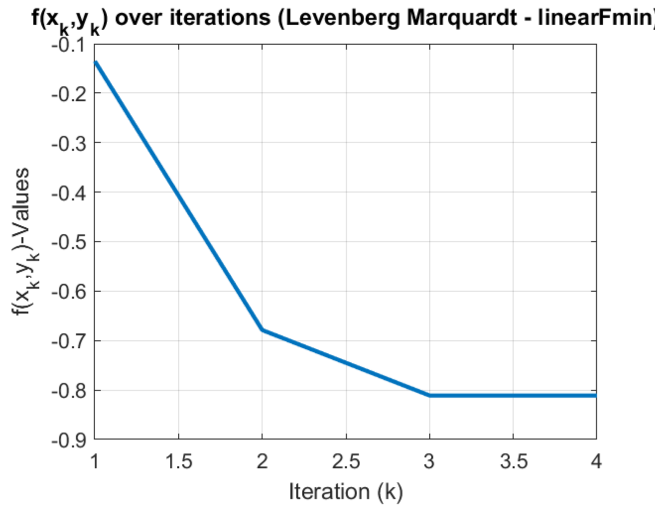
B - Βήμα γ_k τέτοιο ώστε να ελαχιστοποιεί την $f(x_k + \gamma_k \cdot d_k)$

Σε κάθε επανάληψη - αφότου υπολογίσουμε την τιμή του d_k - ψάχνουμε να βρούμε το γ_k για το οποίο ελαχιστοποιείται η $g(\gamma_k) = f(x_k + \gamma_k \cdot d_k)$.Μεθόδους ελαχιστοποίησης έχουμε ήδη δει σε προηγούμενη εργασία, οπότε εδώ το ελάχιστο το βρίσκουμε με την συνάρτηση `fibonacciMethod.m` (Κάνοντας plot τις συναρτήσεις που καλούμαστε να ελαχιστοποιήσουμε ως προς γ είναι εμφανές ότι καλύπτουν τις προϋποθέσεις για την εφαρμογή της μεθόδου Fibonacci). Παρακάτω φαίνονται τα αποτελέσματα του αλγορίθμου με σημεία εκκίνησης τα $(1, -1)$ στην πρώτη γραμμή και $(-1, 1)$ στην δεύτερη. (Η περίπτωση $(0, 0)$ όπως έχουμε ήδη εξηγήσει τερματίζει τον αλγόριθμο από την πρώτη επανάληψη και δεν μετακινεί καθόλου το σημείο από την αρχική του θέση).



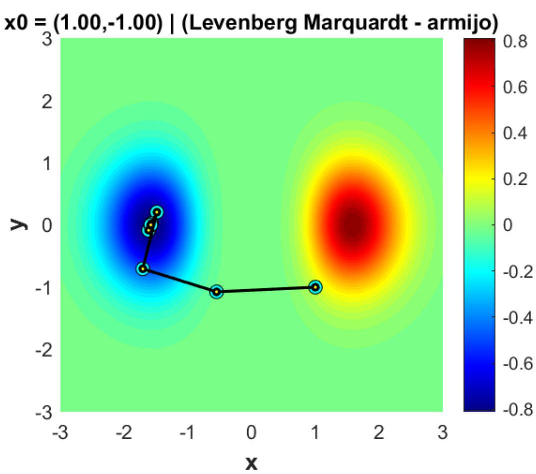


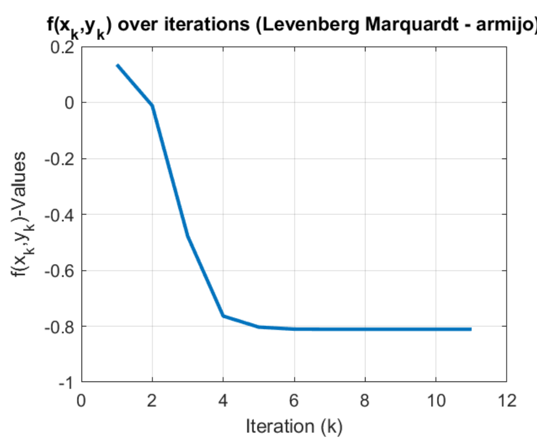


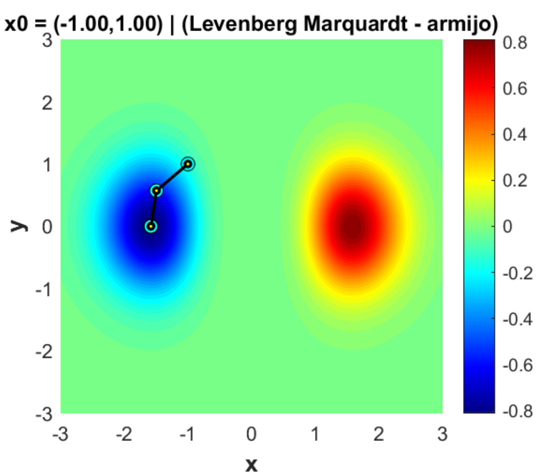


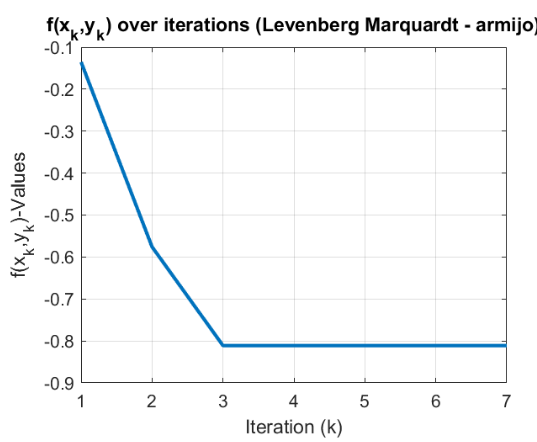
Γ - Βήμα γ_k βάσει του κανόνα Armijo

Τρέχοντας και πάλι τον αλγόριθμο παίρνουμε τα εξής (για $\alpha = 0.01$, $\beta = 0.25$ και $s = 5$):









Η επιλογή των α, β και s έχουμε ήδη ξαναπεί πως επηρεάζει το πόσο γρήγορα θα συγκλίνει ο αλγόριθμος. Αυξάνοντας για παράδειγμα το s επιτρέπουμε στον αλγόριθμο να βλέπει πιο μακριά και να μπορεί να τοποθετεί τα νέα σημεία σε μεγαλύτερη απόσταση (αυξάνουμε το μέγιστο επιτρεπτό βήμα - με αποτέλεσμα αλλαγή στο πλήθος των επαναλήψεων που απαιτεί ο αλγόριθμος, ανάλογα με το που βρέθηκε το νέο σημείο και τις τιμές των α και β).

Σχόλια

Από όλη την ανάλυση της μεθόδου καταλαβαίνουμε πως τόσο το αρχικό σημείο εκκίνησης του αλγορίθμου όσο και ο κανόνας επιλογής του γ_k παίζουν καθοριστικό ρόλο στην σύγκληση της μεθόδου. Συνοπτικά, συμπεραίνουμε πως για σταθερό γ_k δεν είμαστε πάντα σε θέση να κάνουμε τον αλγόριθμο να συγκλίνει (μπορούμε να φτάσουμε κοντά στην λύση αλλά και πάλι υπάρχει αστάθεια). Πρέπει να λαμβάνουμε υπόψιν τόσο την απόσταση του αρχικού σημείου από το ελάχιστο όσο και το αν υπάρχουν ενδιάμεσα άλλα σημεία όπου $\nabla f = 0$ εκτός του ζητούμενου (όπως έχουμε ήδη αναλύσει).

Σχετικά με τους άλλους κανόνες επιλογής του γ_k, βλέπουμε και πάλι εξάρτηση του αποτελέσματος από τις τιμές με τις οποίες θα αρχικοποιήσουμε το πρόβλημα (τι άνω όριο θα βάλουμε στο γ όταν ψάχνουμε το ελάχιστο σε ένα εύρος και τι τιμές θα δώσουμε στα α, β και s για την Armijo). Μπορούμε όμως να πούμε ότι φαίνεται πως ο δεύτερος κανόνας επιλογής στον οποίο επιλέγουμε την τιμή που ελαχιστοποιεί την f στην ευθεία φαίνεται γενικά συγκλίνει ταχύτερα, ενώ ο Amrijo - παρόλο που με μεγαλύτερη ασφάλεια μπορούμε να πούμε ότι οδηγεί σε λύση - υπάρχει περίπτωση να απαιτηθούν περισσότερα βήματα.

Η επιλογή της παραμέτρου μ_k διαδραματίζει καθοριστικό ρόλο στη συμπεριφορά του αλγορίθμου. Συγκεκριμένα για μικρές τιμές του μ_k: Ο αλγόριθμος παρουσιάζει χαρακτηριστικά παρόμοια με αυτά της μεθόδου Newton, καθώς η επίδραση του όρου $\mu_k \cdot I$ είναι περιορισμένη και κυριαρχεί η συμβολή της πληροφορίας από τη δεύτερη παράγωγο (Hessian). Για μεγάλες τιμές του μ_k: Ο όρος $\mu_k \cdot I$ γίνεται κυρίαρχος, με αποτέλεσμα ο αλγόριθμος να αποκτά συμπεριφορά παρόμοια με αυτή της μεθόδου μέγιστης καθόδου (steepest descent). Η ορθή ρύθμιση του μ_k είναι κρίσιμη για την επίτευξη βέλτιστης σύγκλισης, συνδυάζοντας τα πλεονεκτήματα των δύο προσεγγίσεων, δηλαδή την ταχύτητα της μεθόδου Newton και τη σταθερότητα της μεθόδου μέγιστης καθόδου.

Τέλος, βλέπουμε πως η μέθοδος αυτή συγκλίνει ταχύτερα από την πρώτη, κάτι που περιμέναμε άλλωστε από την θεωρία να συμβαίνει (μιας και η πρώτη μέθοδος γεωμετρικά χρησιμοποιεί κάθετα βήματα).