



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Τμήμα ΗΜΜΥ. Τομέας Ηλεκτρονικής
Τεχνικές Βελτιστοποίησης

1η Εργαστηριακή Άσκηση

Ελαχιστοποίηση συναρτήσεων πολλών μεταβλητών
χωρίς περιορισμούς με χρήση παραγώγων

Συντάκτης:
Χρήστος Χουτουρίδης
ΑΕΜ:8997
cchoutou@ece.auth.gr

Διδάσκων:
Γ. Ροβιθάκης
rovithak@auth.gr

1. Εισαγωγή

Η παρούσα εργασία αφορά το πρόβλημα της ελαχιστοποίησης μιας δοσμένης συνάρτησης πολλών μεταβλητών $f: \mathbb{R}^n \rightarrow \mathbb{R}$ χωρίς περιορισμούς. Για το σκοπό αυτό κάνουμε χρήση τριών μεθόδων. Της μεθόδου μέγιστης καθόδου (Steepest Descent), της μεθόδου Newton, και της Levenberg-Marquardt. Ακόμα για κάθε μία από αυτές θα υλοποιήσουμε τρεις διαφορετικές τεχνικές υπολογισμού βήματος.

2. Παραδοτέα

Τα παραδοτέα της εργασίας αποτελούνται από:

- Την παρούσα αναφορά.
- Τον κατάλογο **scripts/**, που περιέχει τον κώδικα της MATLAB.
- Το σύνδεσμο με το αποθετήριο που περιέχει όλο το project με τον κώδικα της MATLAB, της αναφοράς και τα παραδοτέα.

3. Προγραμματιστική προσέγγιση

Για τον προγραμματισμό και εκτέλεση των μεθόδων της παρούσας εργασίας έγινε χρήση της MATLAB. Στον κατάλογο **scripts**, περιέχονται όλες οι μέθοδοι και οι τεχνικές υπολογισμού βημάτων με τη μορφή συναρτήσεων καθώς και scripts που τις καλούν. Για κάθε μία μέθοδο (ένα θέμα της εργασίας), υπάρχει το αντίστοιχο script που περιέχει τους υπολογισμούς, τις κλήσεις των μεθόδων και τη δημιουργία των διαγραμμάτων. Για το πρώτο θέμα το αρχείο `Script_1_Plots.m` για το δεύτερο το `Script_2_Steepest_descent.m` και ούτω καθεξής. Στην παρούσα εργασία η υλοποίηση του κώδικα ακολουθεί την τεχνική της προηγούμενης εργασίας και “ομαδοποιεί” αρκετές λειτουργίες. Πιο συγκεκριμένα.

3.1. Κλήση μεθόδων επιλογής βήματος γ_k

Δεδομένου ότι οι μέθοδοι θα πρέπει να καλεστούν και εκτελεστούν με παραπάνω από μία τεχνική επιλογής βήματος γ_k , δημιουργήσαμε εσωτερικά της κάθε μεθόδου ένα κοινό interface για τις μεθόδους επιλογής βήματος. Αυτό έχει τη μορφή: **`gamma_<method>(f, dk, xk)`**, όπου το **`f`** είναι η αντικειμενική συνάρτηση, **`dk`** η τιμή της συνάρτησης κλίσης στο **`xk`** και **`xk`** το σημείο ενδιαφέροντος. Για την κάθε μία από αυτές δημιουργήσαμε ξεχωριστή συνάρτηση που υλοποιεί το παραπάνω interface. Μία για σταθερό βήμα, μία για επιλογή βήματος που ελαχιστοποιεί την $f(x_k + \gamma_k d_k)$ και μία με τη μέθοδο Armijo. Για την επιλογή και κλήση των μεθόδων επιλογής βήματος εισαγάγαμε μία νέα παράμετρο string που χρησιμοποιείται ως enumerator και με βάση αυτή γίνεται η τελική επιλογή. Έτσι για παράδειγμα η κλήση `method_newtown(f, gradf, [0, 0], 0.001, 1000, 'armijo')` υλοποιεί τη μέθοδο newton χρησιμοποιώντας τη μέθοδο Armijo για επιλογή βήματος ενώ η `method_newtown(f, gradf, [0, 0], 0.001, 1000, 'minimized')`, χρησιμοποιεί βήμα που ελαχιστοποιεί την $f(x_k + \gamma_k d_k)$.

3.2. Symbolic expression functions

Μία ακόμη προγραμματιστική τεχνική που ακολουθήθηκε είναι η χρήση **symbolic expression** για την αναπαράσταση των διαφορετικών αντικειμενικών συναρτήσεων. Ο λόγος που επιλέχθηκε είναι η **δυνατότητα εξαγωγής ενός symbolic expression που αναπαριστά την κλίση ∇f και τον Εσσιανό $\nabla^2 f$ μιας συνάρτησης** από την MATLAB, κάνοντας χρήση των εντολών `gradient()` και `hessian()`. Αν αντίθετα χρησιμοποιούσαμε απλές συναρτήσεις, πολυώνυμα ή lambdas για την αναπαράσταση των αντικειμενικών συναρτήσεων, τότε για τον υπολογισμό της κλίσης και του Εσσιανού θα έπρεπε:

- Είτε να υπολογίζαμε αριθμητικά τις παραγώγους gradient και hessian μέσα στις μεθόδους, κάτι που θα εισήγαγε **αχρείαστο αριθμητικό σφάλμα**.

- Είτε να κάναμε χρήση δύο επιπλέον συναρτήσεων (ή πολυωνύμων) για την αναπαράστασή τους, κάτι που ουσιαστικά θα δημιουργούσε **πλεονασμό πληροφορίας εισόδου** και άρα μεγαλύτερη πιθανότητα να κάνουμε λάθος.

Η αναπαράσταση όμως με χρήση symbolic expression είναι πιο “βαριά” όταν χρειάζεται να υπολογίσουμε την τιμή μιας συνάρτησης σε κάποιο σημείο (`subs(expr, number)`). Αυτό είναι κάτι που χρειάζεται εκτενώς στον κώδικά μας. Για το λόγο αυτό, ενώ η συνάρτηση δίνεται ως symbolic expression, μέσω αυτής υπολογίζονται αυτόματα η κλίση, ο Εσσιανός αλλά και οι “κανονικές” συναρτήσεις MATLAB που τις υλοποιούν. Έτσι έχουμε την ακριβή αναπαράσταση της κλίσης και του Εσσιανού ως συναρτήσεις χωρίς να πληρώνουμε το κόστος της `subs()`.

4. Απεικόνιση της συνάρτησης

Η συνάρτηση με την οποία ασχολούμαστε στην παρούσα εργασία είναι η:

$$f(x, y) = x^5 \cdot e^{-x^2 - y^2} \quad (1)$$

Στο παρακάτω σχήμα 1 φαίνεται η τρισδιάστατη απεικόνιση της συνάρτησης.

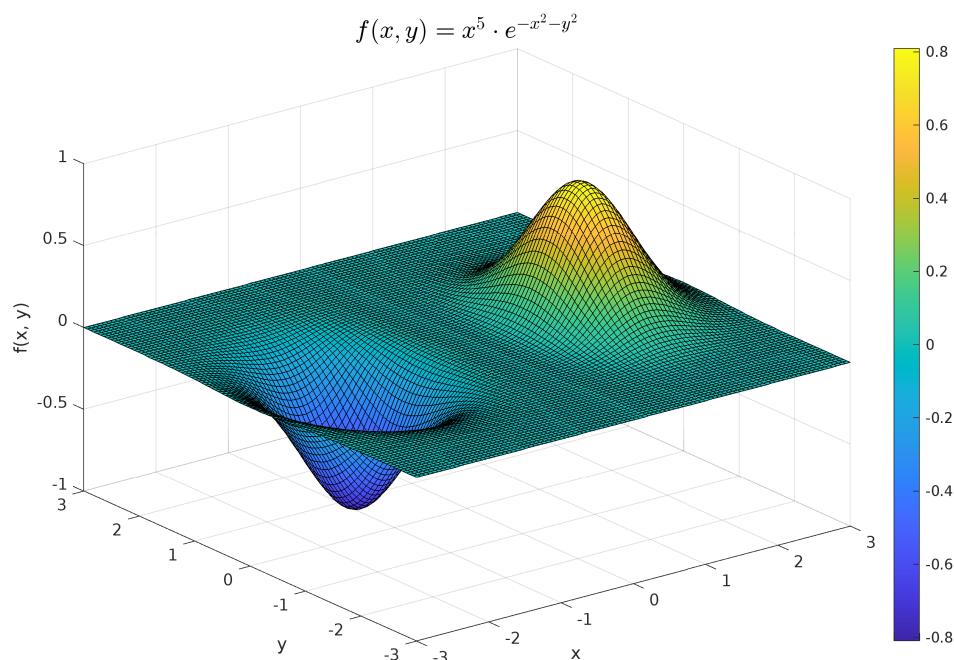
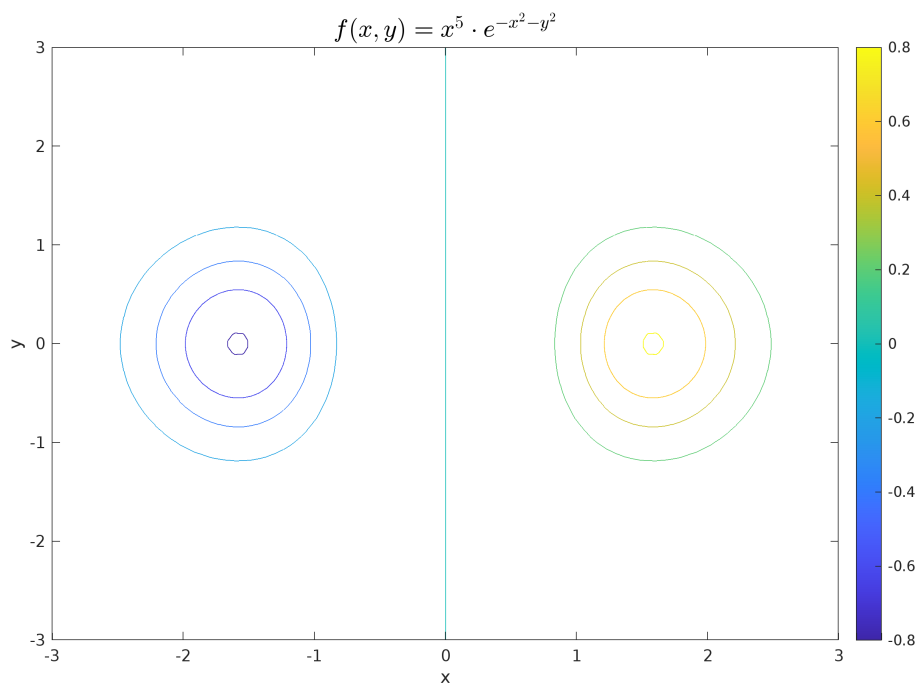


Figure 1: Γραφική παράσταση της f

Από το σχήμα μπορούμε πολύ εύκολα να διακρίνουμε ότι η συνάρτηση έχει ένα ευκρινές μέγιστο και ένα ελάχιστο στο διάστημα $x, y \in [-3, 3]$. Για να πάρουμε μια καλύτερη αίσθηση για το που βρίσκονται αυτά τα τοπικά ακρότατα, παρακάτω παραθέτουμε ένα γράφημα με τις ισοβαρείς καμπύλες της f .

Figure 2: Ισοβαρείς της f

Από το παραπάνω σχήμα 2 φαίνεται ότι το ελάχιστο της f βρίσκεται στο αρνητικό ημιεπίπεδο των x , κοντά στο $y = 0$

Τα διαγράμματα για τη μέθοδο δημιουργούνται εκτελώντας το αρχείο **Script_1_Plots.m**

5. Τεχνικές υπολογισμού βήματος

Πριν προχωρήσουμε στα επόμενα θέματα της εργασίας και στην ανάλυση των μεθόδων υπολογισμού του ελάχιστου, θέλουμε να αναφερθούμε στις διαφορετικές τεχνικές επιλογής βήματος γ_k και ειδικότερα για αυτή της ελαχιστοποίησης της $f(x_k + \gamma_k d_k)$ και την Armijo.

5.1. Ελαχιστοποίηση της $f(x_k + \gamma_k d_k)$

Η μέθοδος αυτή αναζητά την τιμή γ_k που ελαχιστοποιεί την τιμή της συνάρτησης κατά μήκος της κατεύθυνσης d_k . Δηλαδή, λύνουμε το μονοδιάστατο πρόβλημα:

$$\min \gamma_k f(x_k + \gamma_k d_k)$$

Η κατεύθυνση d_k μπορεί να είναι:

- Η αρνητική κλίση $-\nabla f(x_k)$ (Steepest Descent).
- Η Newton direction $-H_k^{-1} \nabla f(x_k)$ (Newton και Levenberg-Marquardt).

Πλεονεκτήματα της μεθόδου είναι η **ακρίβεια**, καθώς το βήμα γ_k υπολογίζεται με βέλτιστο τρόπο για τη συγκεκριμένη κατεύθυνση και η **γρήγορη σύγκλιση**, ειδικά σε προβλήματα όπου η $f(x)$ είναι καμπυλωτή με γ_k επακριβώς ορισμένο. Στα μειονεκτήματα μπορούμε να αναφέρουμε το **υπολογιστικό κόστος**, καθώς ο υπολογισμός του γ_k απαιτεί πολλαπλές αξιολογήσεις της $f(x)$.

5.2. Armijo rule

Η Armijo rule είναι μια προσαρμοστική τεχνική που επιλέγει το γ_k για να εξασφαλίσει επαρκή μείωση της συνάρτησης. Η βασική ιδέα είναι ότι η συνάρτηση πρέπει να μειώνεται "αρκετά" σε κάθε βήμα, χωρίς να χρειάζεται να υπολογίζεται το ακριβές ελάχιστο. Η συνθήκη του Armijo είναι:

$$f(x_k + \gamma_k d_k) \leq f(x_k) + \sigma \gamma_k d_k^T * \nabla f(x_k)$$

Όπου $\sigma \in (0, 0.1)$ είναι μια σταθερά (τυπικά $\sigma = 0.1$) και γ_k αρχικά να ορίζεται ως 1 και να μειώνεται προοδευτικά (π.χ., $\gamma_k = \beta \cdot \gamma_k$) έως ότου ικανοποιηθεί η συνθήκη.

Πλεονεκτήματα της μεθόδου είναι η **σταθερότητα**, καθώς αποτρέπει πολύ μεγάλα βήματα που μπορεί να αυξήσουν την τιμή της $f(x)$, αλλά και η **ανθεκτικότητα**, καθώς λειτουργεί καλά ακόμα και όταν η $f(x)$ δεν συμπεριφέρεται πολύ καλά. Στα μειονεκτήματα μπορούμε να αναφέρουμε την **εξάρτησή της από τις παραμέτρους σ και β** , μια κακή επιλογή των οποίων μπορεί να οδηγήσει σε αργή σύγκλιση.

6. Μέθοδος Μέγιστης Καθόδου - Steepest Descent

Η πρώτη μέθοδος που χρησιμοποιούμε στην εργασία (Θέμα 2), είναι η μέθοδος μέγιστης καθόδου. Είναι μια μέθοδος πρώτης τάξης που χρησιμοποιεί την κατεύθυνση της αρνητικής κλίσης $\nabla f(x, y)$ της f ως κατεύθυνση καθόδου. Η μέθοδος θεωρείται βασική και συχνά χρησιμοποιείται ως εισαγωγή στις μεθόδους βελτιστοποίησης.

Η μέθοδος επιλέγει την κατεύθυνση $d_k = -\nabla f(x_k)$, η οποία είναι η κατεύθυνση της μέγιστης τοπικής μείωσης της συνάρτησης. Στη συνέχεια, υπολογίζεται το βήμα γ_k για να βρεθεί το επόμενο σημείο $x_{k+1} = x_k + \gamma_k d_k$. Για να χρησιμοποιήσουμε τη μέθοδο, η συνάρτηση f **πρέπει να είναι συνεχής και διαφορίσιμη** και η κλίση ∇f να είναι υπολογίσιμη. Επίσης για την εφαρμογή της μεθόδου το αρχικό σημείο θα πρέπει να **μην είναι ακρότατο** της f , δηλαδή $\nabla f(x_0) \neq 0$.

Όλοι οι υπολογισμοί και τα διαγράμματα για τη μέθοδο βρίσκονται στο αρχείο `Script_2_Steepest_descent.m`

6.1. Σημείο εκκίνησης (0,0)

Για το σημείο (0, 0) η κλίση της f είναι: $\nabla f(0, 0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, με αποτέλεσμα η μέθοδος να μην μπορεί να εφαρμοστεί για κανένα τρόπο υπολογισμού βήματος.

6.2. Σημείο εκκίνησης (-1,1)

Για το σημείο (-1, 1) η τιμή της f είναι: $f(-1, 1) = -0.135335$ και το διάνυσμα της κλίσης: $\nabla f(0, 0) = \begin{bmatrix} 0.4060 \\ 0.2707 \end{bmatrix}$, επομένως μπορούμε να εφαρμόσουμε τη μέθοδο.

Σταθερό βήμα

Επιλέγοντας ακρίβεια $\epsilon = 0.0001$, εκτελούμε την μέθοδο `method_steepest_descent()` και υπολογίζουμε τον αριθμό επαναλήψεων για διαφορετικές τιμές γ_k .

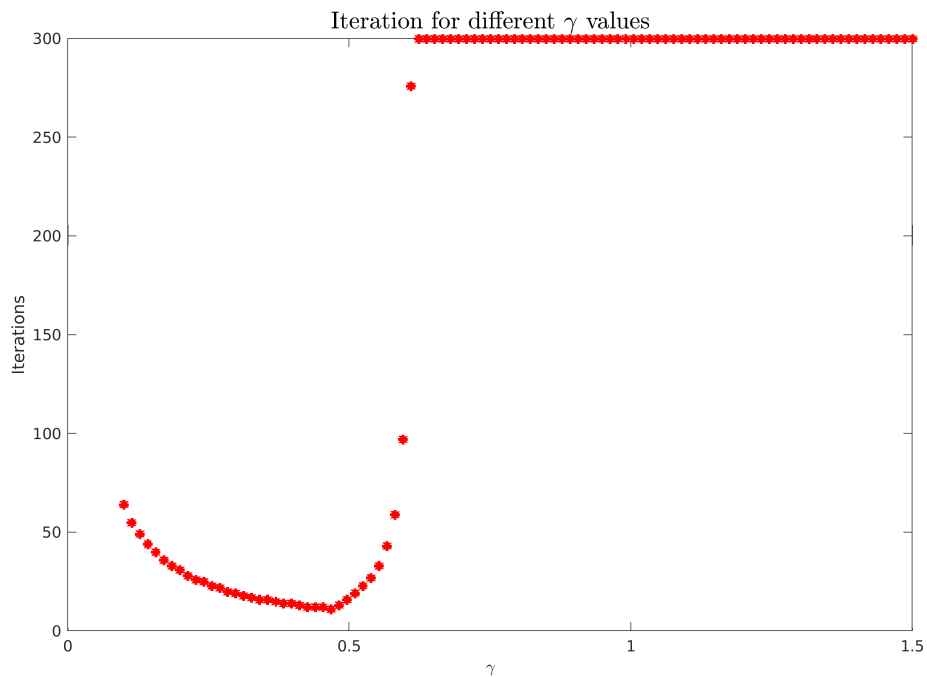


Figure 3: Αριθμός επαναλήψεων για διαφορετικές τιμές γ_k [Μέγιστη Κάθοδος].

Στο παραπάνω σχήμα 3 παρατηρούμε ότι για τιμές του $\gamma_k > 0.61$ η μέθοδος αποκλίνει. Από την παραπάνω διαδικασία επίσης υπολογίζουμε το $\gamma_k = 0,46768$ για το οποίο η μέθοδος συγκλίνει με τα λιγότερα βήματα. Στο παρακάτω σχήμα 4 αναπαριστούμε την πορεία των σημείων καθώς συγκλίνουν στο ελάχιστο.

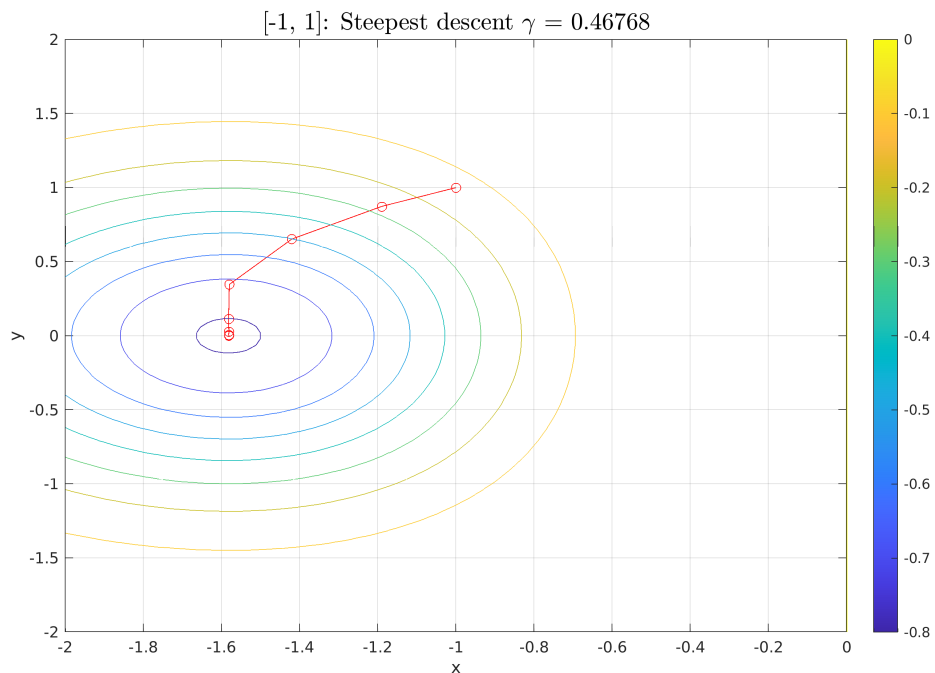


Figure 4: Σύγκλιση της μεθόδου Steepest descent [fixed γ_k].

Ελαχιστοποίηση της $f(x_k + \gamma_k d_k)$

Για την ελαχιστοποίηση της f , χρησιμοποιήθηκε η bisection από την προηγούμενη εργασία, η οποία τροποποιήθηκε ώστε δέχεται functions και όχι symbolic expressions.

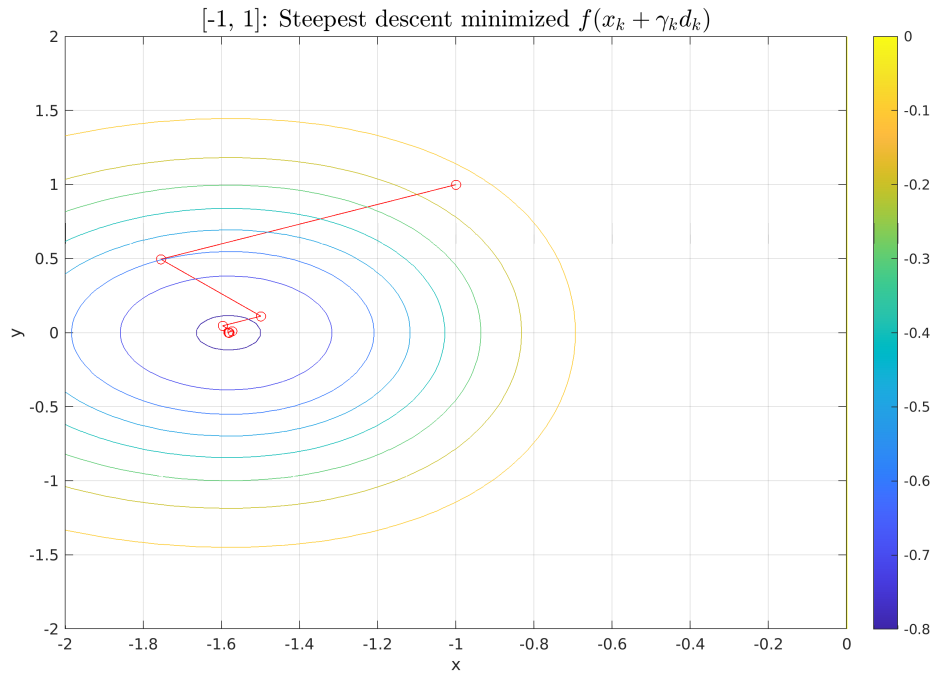


Figure 5: Σύγκλιση της μεθόδου *Steepest descent* [minimized f].

Από το γράφημα φαίνεται τόσο ότι η μέθοδος συγκλίνει κοντά στο ελάχιστο γρηγορότερα, όσο και ότι πραγματοποιεί “διορθώσεις πορείας”.

Armijo rule

Για τη μέθοδο η βασική ιδέα είναι να ξεκινήσει ο αλγόριθμος από ένα μεγάλο $\gamma_k = 1$ και συνεχώς να μειώνεται με βάση τον κανόνα Armijo. Μετά από ένα tuning των παραμέτρων της μεθόδου καταλήξαμε στα $\beta = 0.4$, $\sigma = 0.1$

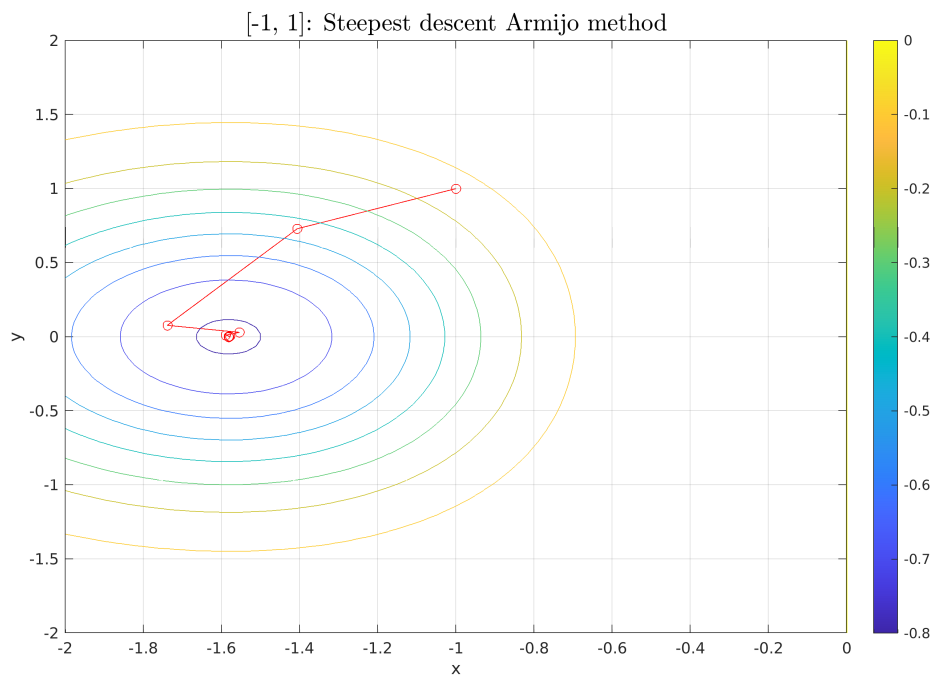


Figure 6: Σύγκλιση της μεθόδου *Steepest descent* [armijo rule].

6.3. Σημείο εκκίνησης (1,-1)

Για το σημείο (1, -1) η τιμή της f είναι: $f(1, -1) = -0.135335$ και το διάνυσμα της κλίσης: $\nabla f(0, 0) = \begin{bmatrix} 0.4060 \\ 0.2707 \end{bmatrix}$, επομένως μπορούμε να εφαρμόσουμε τη μέθοδο.

Σταθερό βήμα

Για σταθερό βήμα εκτελέσαμε διαδοχικά τη μέθοδο `method_steepest_descent()` για να υπολογίσουμε τον αριθμό επαναλήψεων για διαφορετικές τιμές γ_k , όμως σε καμία τιμή ο αλγόριθμος δεν συγκλίνει. Ακόμα και για μεγάλες τιμές του γ_k , ο αλγόριθμος εγκλωβίζεται στο δεξιό ημιεπίπεδο.

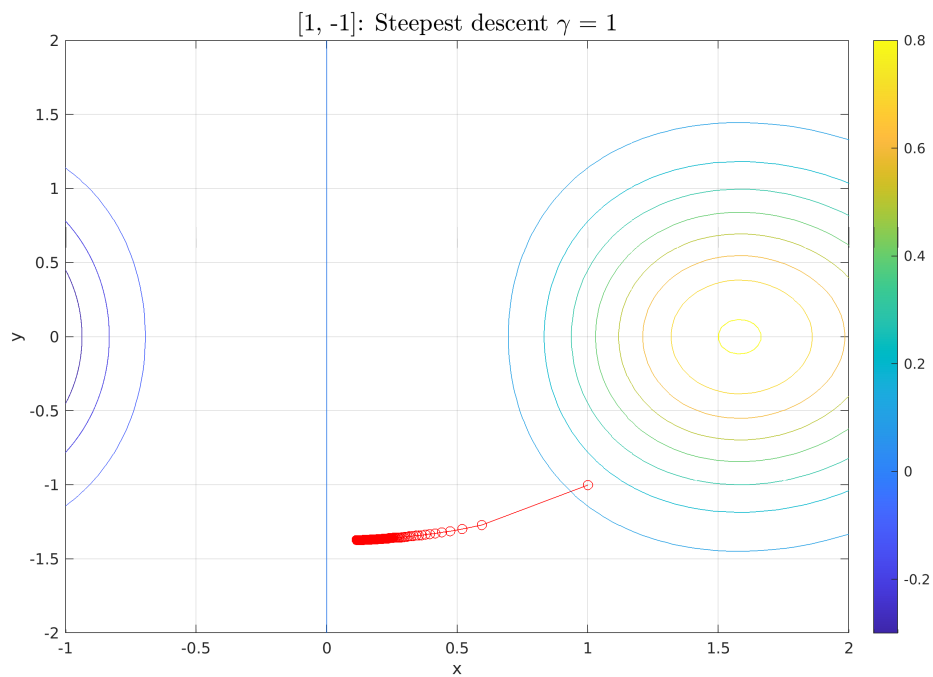


Figure 7: Μη σύγκλιση της μεθόδου Steepest descent [Fixed step].

Ελαχιστοποίηση της $f(x_k + \gamma_k d_k)$

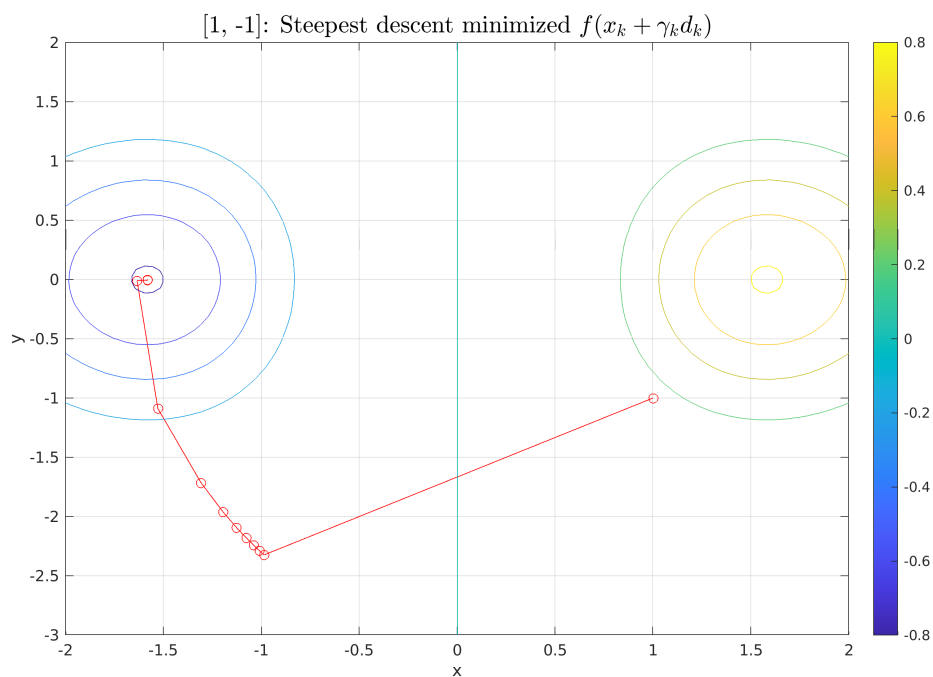


Figure 8: Σύγκλιση της μεθόδου Steepest descent [minimized f].

Από το γράφημα φαίνεται ότι η μέθοδος συγκλίνει, καταφέροντας να περάσει την περιοχή με μηδενικές κλίσεις κοντά στον άξονα των ψ .

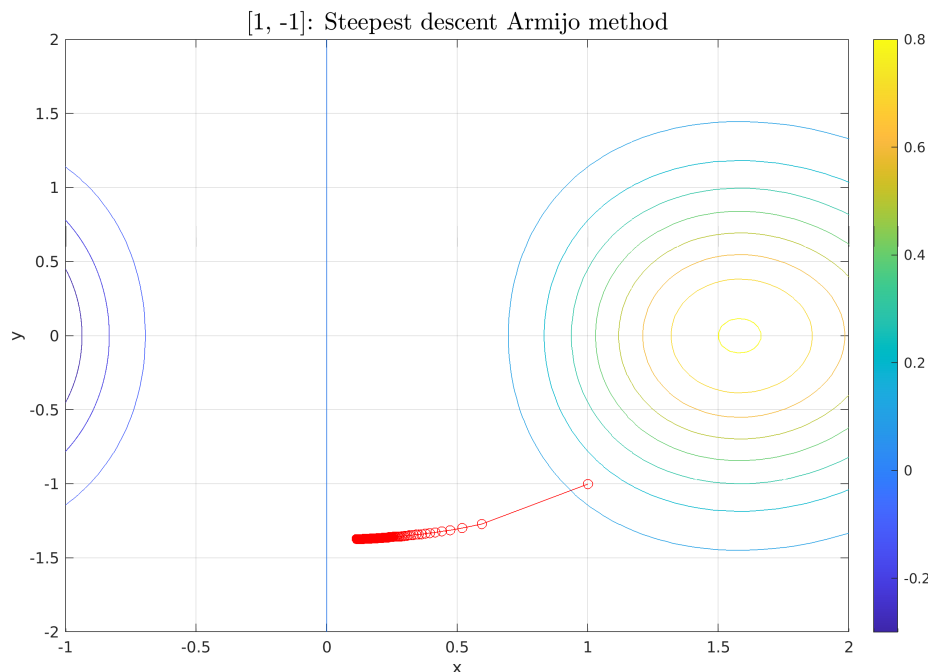
Armijo rule

Figure 9: Μη σύγκλιση της μεθόδου *Steepest descent* [armijo rule].

Αντίθετα η μέθοδος armijo δεν συγκλίνει, καθώς εγκλωβίζεται στο δεξιό ημιεπίπεδο.

7. Μέθοδος Newton

Η δεύτερη μέθοδος που χρησιμοποιούμε στην εργασία (Θέμα 3), είναι η μέθοδος Newton. Η μέθοδος χρησιμοποιεί πληροφορίες δεύτερης τάξης (Hessian) για τη βελτίωση της κατεύθυνσης καθόδου. Αν η συνάρτηση είναι τετραγωνική, η μέθοδος συγκλίνει σε ένα βήμα. Η μέθοδος ορίζει την κατεύθυνση

$$d_k = -H_k^{-1} \nabla f(x_k)$$

Όπου H_k είναι ο Εσσιανός πίνακας της f στο x_k . Το επόμενο σημείο υπολογίζεται ως

$$x_{k+1} = x_k + \gamma_k d_k$$

Με κατάλληλο υπολογισμό του βήματος. Για να λειτουργήσει η μέθοδος η f πρέπει να είναι **δύο φορές διαφορίσιμη** και ο Εσσιανός H_k να είναι **θετικά ορισμένος και αντιστρέψιμος**.

Στα πλεονεκτήματα της μεθόδου είναι η **ταχύτερη σύγκλιση** από την Steepest Descent για κυρτές συναρτήσεις και το γεγονός ότι εκμεταλλεύεται την **πληροφορία καμπυλότητας** της συνάρτησης. Όμως είναι υπολογιστικά δαπανηρή και δεν είναι ανθεκτική σε μη κυρτές συναρτήσεις ή σε περιπτώσεις όπου ο Εσσιανός είναι κακώς ορισμένος.

Όλοι οι υπολογισμοί για τη μέθοδο βρίσκονται στο αρχείο **Script_3_Newton.m**

7.1. Σημείο εκκίνησης (0,0)

Για το σημείο $x_k = (0, 0)$ η κλίση της f είναι: $\nabla f(x_k) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ και ο εσσιανός $H(x_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ με αποτέλεσμα η μέθοδος και εδώ να μην μπορεί να εφαρμοστεί για κανένα τρόπο υπολογισμού βήματος.

7.2. Σημείο εκκίνησης (-1,1)

Για το σημείο $x_k = (-1, 1)$ η κλίση της f είναι: $\nabla f(x_k) = \begin{bmatrix} 0.406 \\ 0.270 \end{bmatrix}$ και ο εσσιανός $H(x_k) = \begin{bmatrix} -0.270 & -0.812 \\ -0.812 & -0.270 \end{bmatrix}$ με ιδιοτιμές $\lambda = \begin{bmatrix} -1.082 \\ 0.541 \end{bmatrix}$. Από τα παραπάνω προκύπτει πως ο Εσσιανός είναι αόριστος και άρα δεν μπορεί να εφαρμοστεί η μέθοδος, για κανένα τρόπο υπολογισμού βήματος.

7.3. Σημείο εκκίνησης (1,-1)

Για το σημείο $x_k = (1, -1)$ η κλίση της f είναι: $\nabla f(x_k) = \begin{bmatrix} 0.406 \\ 0.270 \end{bmatrix}$ και ο εσσιανός $H(x_k) = \begin{bmatrix} 0.270 & 0.812 \\ 0.812 & 0.270 \end{bmatrix}$ με ιδιοτιμές $\lambda = \begin{bmatrix} -0.541 \\ 1.082 \end{bmatrix}$. Και εδώ, από τα παραπάνω προκύπτει πως ο Εσσιανός είναι αόριστος και άρα δεν μπορεί να εφαρμοστεί η μέθοδος, για κανένα τρόπο υπολογισμού βήματος.

8. Μέθοδος Levenberg-Marquardt

Η τελευταία μέθοδος που χρησιμοποιούμε στην εργασία (Θέμα 4), είναι η μέθοδος Levenberg-Marquardt. Πρόκειται για μια τροποποιημένη έκδοση της μεθόδου Newton, η οποία εισάγει έναν παράγοντα απόσβεσης για τη σταθεροποίηση όταν ο εσσιανός δεν είναι θετικά ορισμένος. Για το λόγο αυτό χρησιμοποιεί ένας προσαρμοσμένος εσσιανός $H'_k = H_k + \mu_k I$, όπου $\mu_k > 0$ ένας παράγοντας απόσβεσης. Για μεγάλες τιμές του μ_k η μέθοδος συμπεριφέρεται σαν Gradient Descent.

Όλοι οι υπολογισμοί για τη μέθοδο βρίσκονται στο αρχείο **Script_4_LevMar.m**

9. Σύγκριση των μεθόδων

Εκτελώντας όλους του αλγόριθμους για τα ίδια δεδομένα, **για τον αριθμό επαναλήψεων** έχουμε:

- ...

10. Συμπεράσματα

Οι μέθοδοι της παρούσας εργασίας αποτελούν βασικές τεχνικές για την εύρεση του τοπικού ελαχίστου ...