ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Βελτιστοποίησης

2η Εργαστηριακή Άσκηση

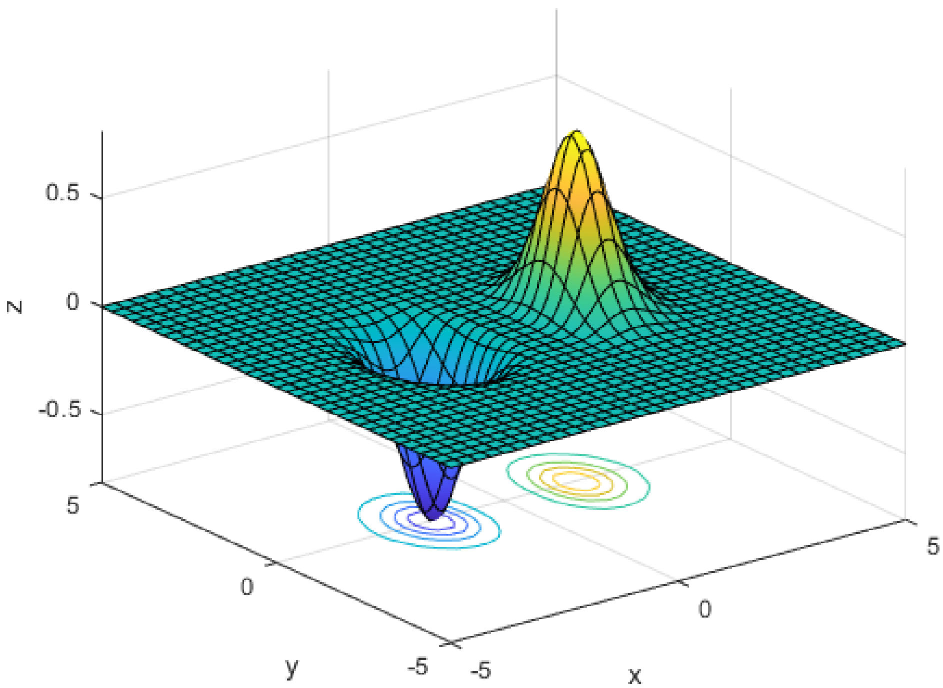
Ελαχιστοποίηση με χρήση παραγώγων

- Ονοματεπώνυμο: Κυπριανίδης Άρης-Ευτύχιος
- ΑΕΜ: 10086
- Email: akyprian@ece.auth.gr

ΘΕΣΣΑΛΟΝΙΚΗ 2022

Θέμα 1

Η συνάρτηση μας είναι η $f(x,y) = x^5 e^{(-x^2-y^2)}$.



Γραφική παρασταση της $f(x,y) = x^5 e^{(-x^2-y^2)}$

Το ελάχιστο έχει τιμή -0.8111 και βρίσκεται στο σημείο (-1.5811, 0). Παρατηρούμε ότι υπάρχουν τοπικά ελάχιστα για την περιοχή γύρω από το x = 0. Αυτός είναι και ένας από τους λόγους που ο αλγόριθμος δεν θα φτάνει στο ελάχιστο.

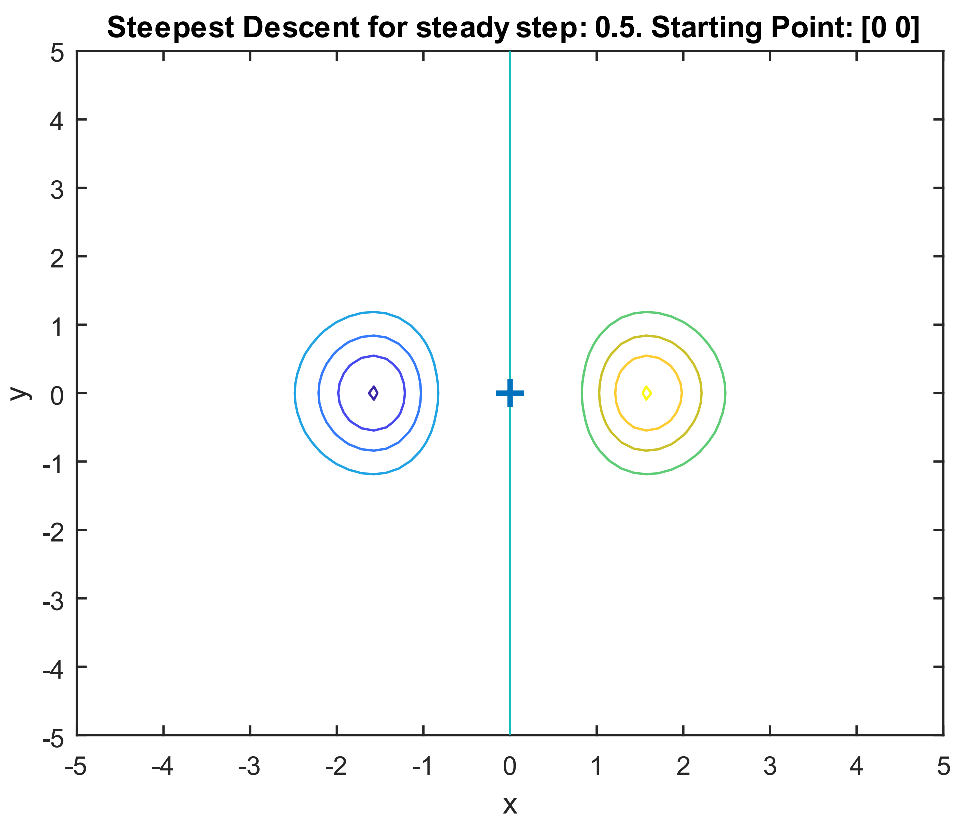
Θέμα 2

Μέθοδος μέγιστης καθόδου (Steepest Descent)

Αρχεία: steepest_descent.m, thema_2.m

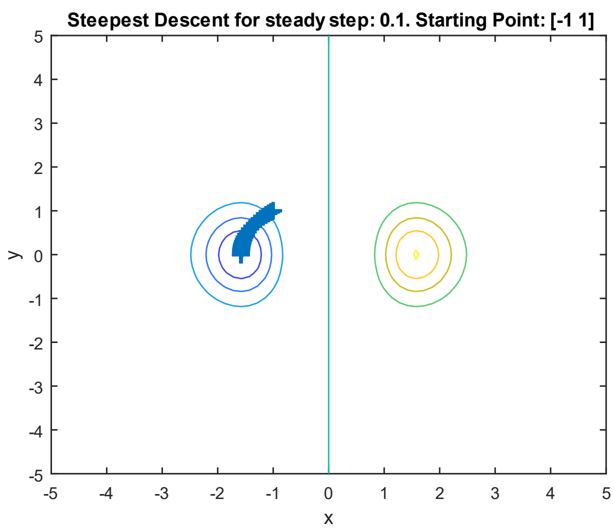
- Σημείο [0, 0]

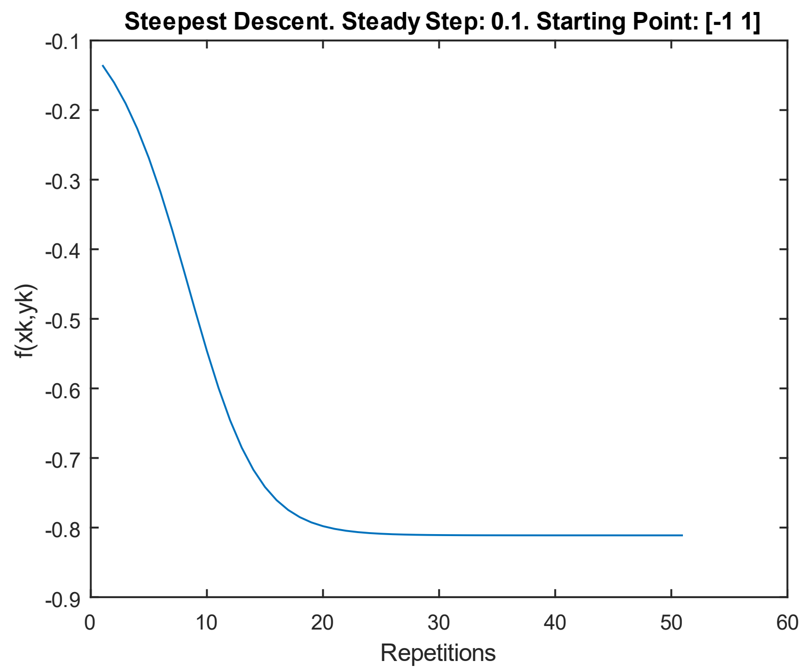
Λόγω της φύσης του αλγορίθμου, το σημείο δεν θα μπει μέσα στον αλγόριθμο γιατί θα έχει κλίση 0, οπότε μένει στο σημείο [0, 0]. Το ίδιο συμβαίνει για οποιοδήποτε βήμα.

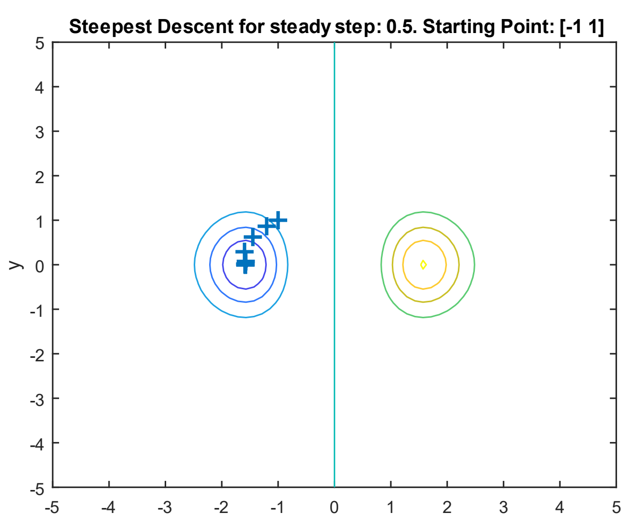


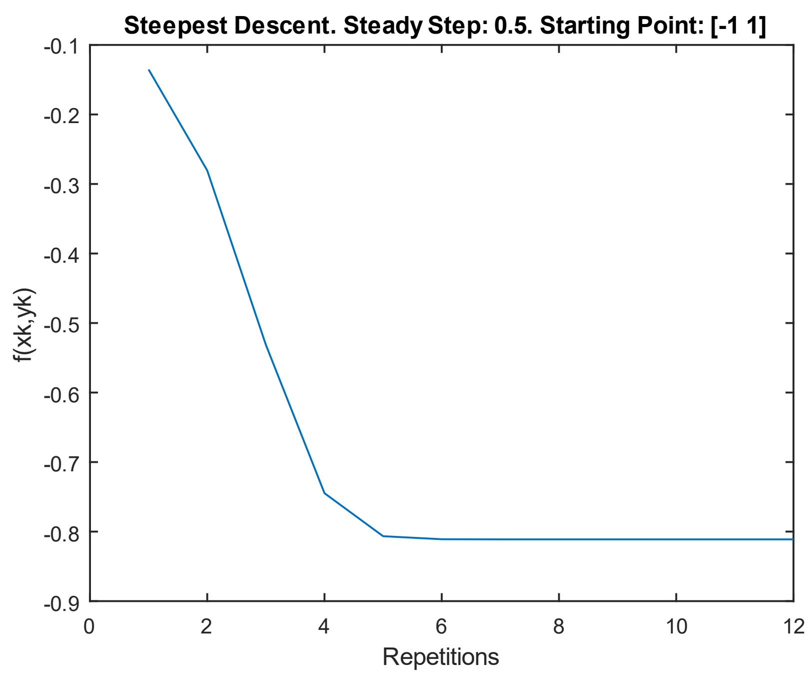
- Σημείο [-1, 1]
    1. Για σταθερά βήματα: 0.1, 0.5, 5

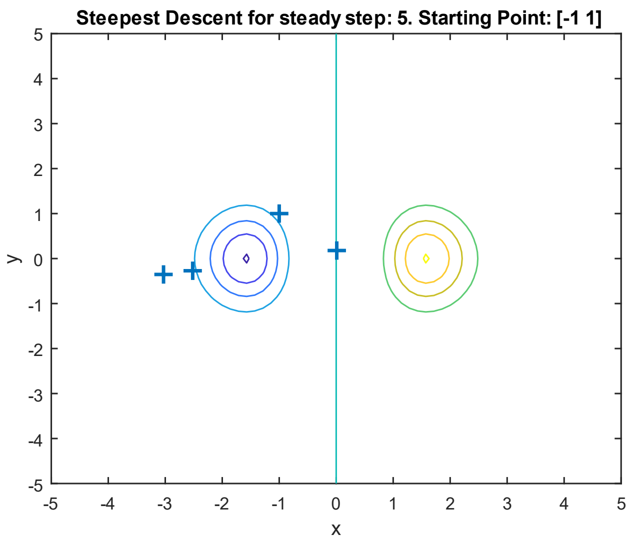
Το σταθερό βήμα σε αυτό το σημειό για μικρές τιμές καταφέρνει να φτάσει στο ελάχιστο.

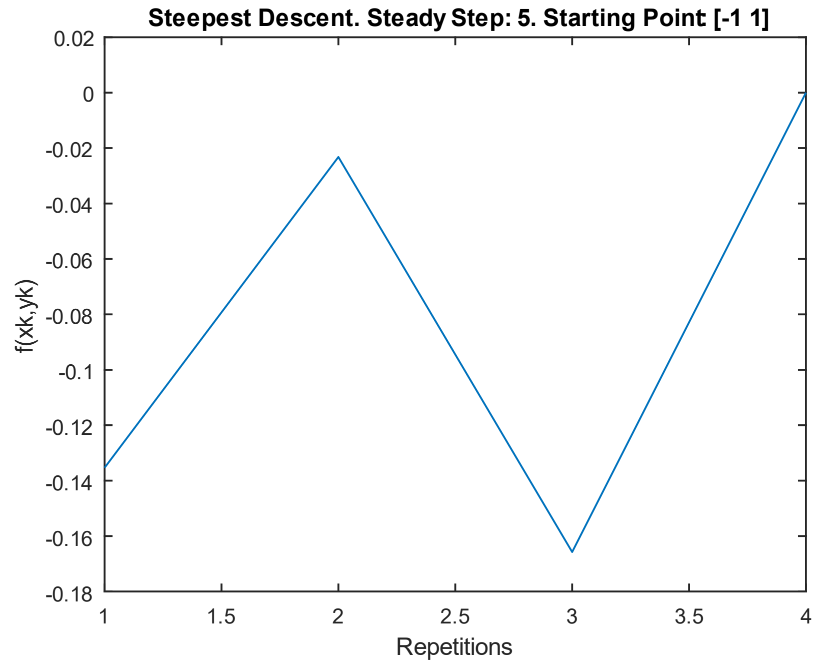












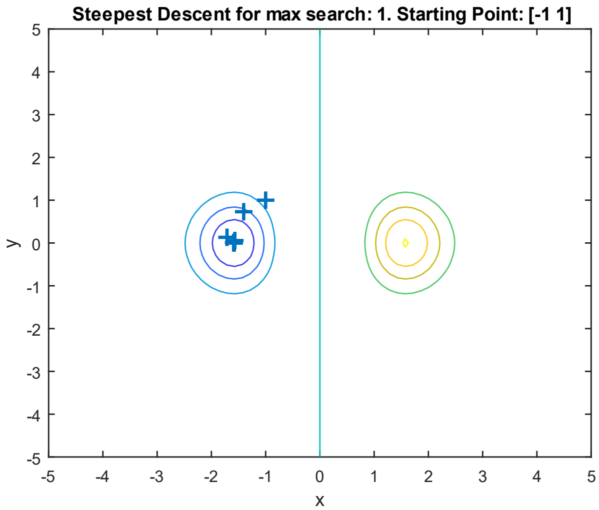
Παρατηρούμε από το βήμα 0.1, 0.5 χρειάζονται πολύ λιγότεροι υπολογισμοί για να φτάσουμε στο ακρότατο.Για βήμα 1, ο αλγόριθμος δεν τερματίζει γιατί πηγαίνει γύρω από το ακρότατο χωρίς να φτάνει σε αυτό. Γι'αυτό παρουσιάζουμε για βήμα 5 που φαίνεται πως ο αλγόριθμος δεν λειτουργεί επιθυμητά.

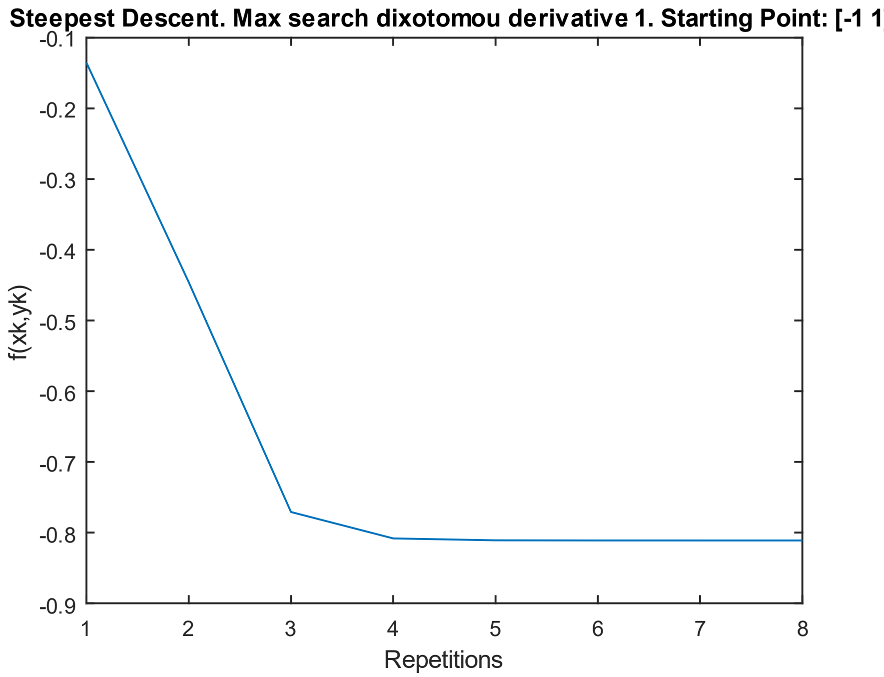
Στα αριστερά δείχνουμε τα σημεία πως κινούνται πάνω στην κάτοψη της συνάρτησης, ενώ στα δεξιά δείχνουμε την τιμή της συνάρτησης για τα σημεία που υπολογίζουμε. Δεξιά θέλουμε η τιμή να καταλήγει στο -0.8111, όπως φαίνεται στα πρώτα δύο διαγράμματα.

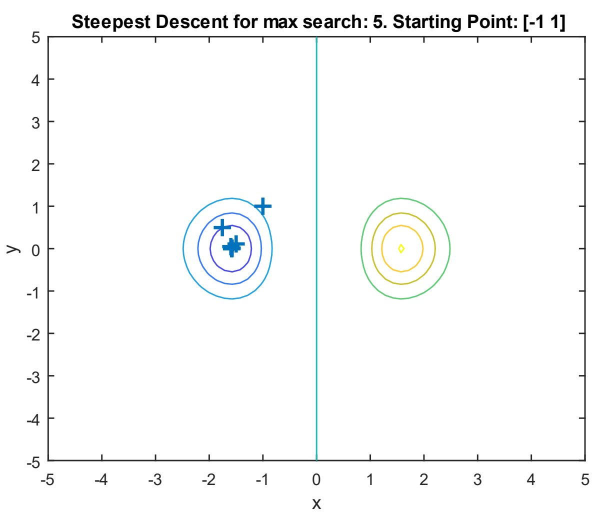
2. Για βήμα μέσω dixotomouDerivative:

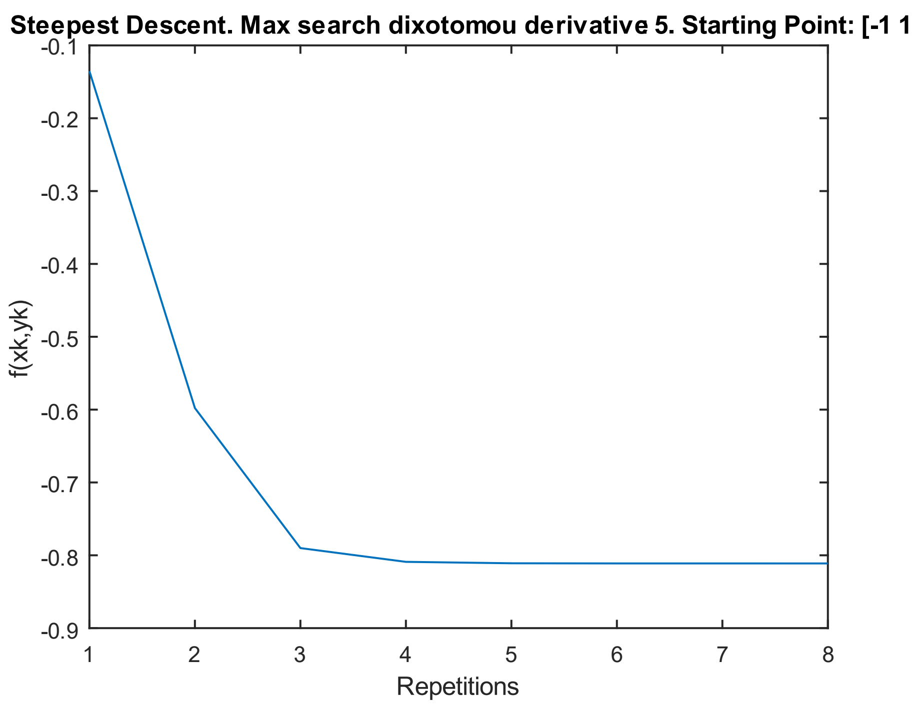
Για την ελαχιστοποίηση του βήματος χρησιμοποιούμε την συνάρτηση *dixotomouDerivative.m* παραλλαγμένη ώστε να γυρνάει την διάμεσο αντί του διαστήματος.

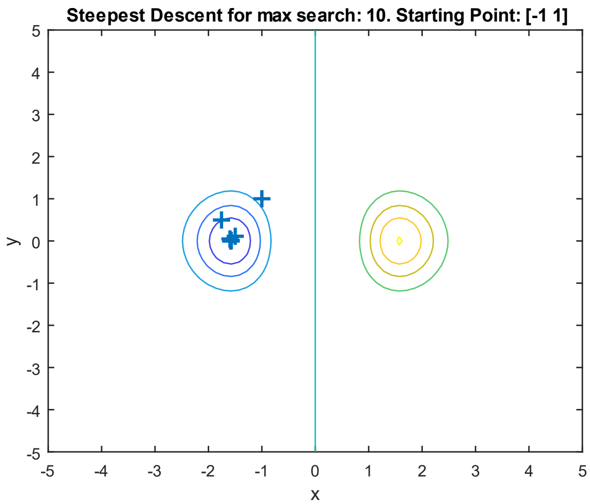
Μέγιστο δυνατό βήμα: 1, 5, 10

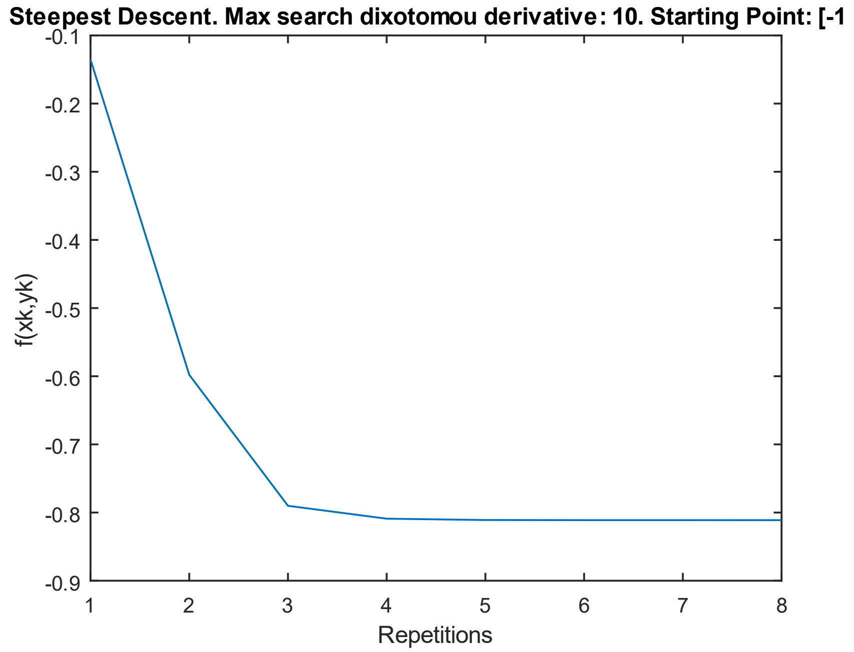










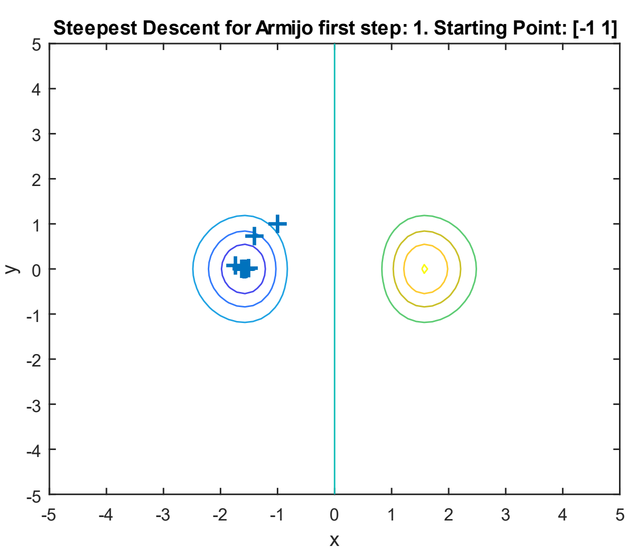


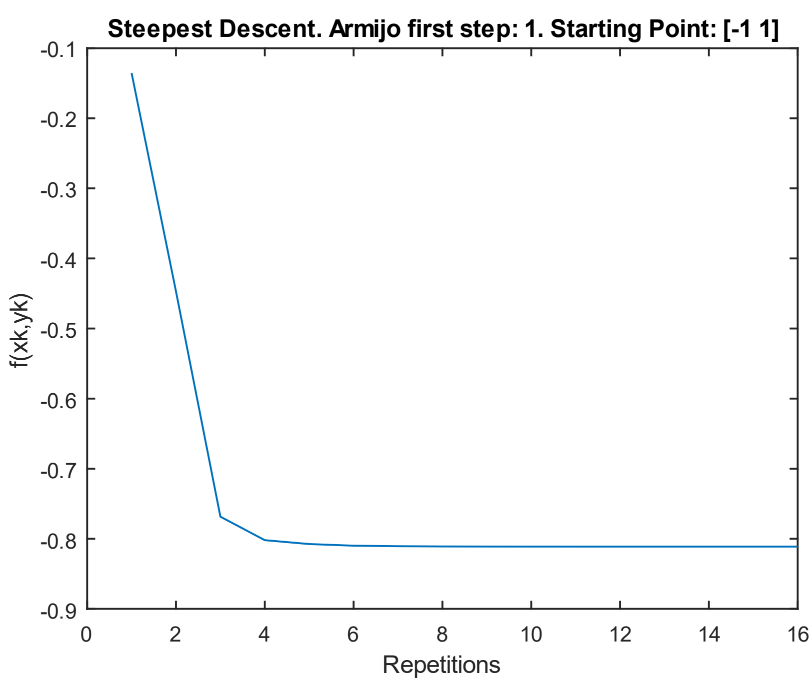
Ο αλγόριθμος είναι ίδιος για όλα τα βήματα. Καταφέρνει για λίγες επαναλήψεις να είναι στο ελάχιστο, σε σχέση με το σταθερό βήμα.

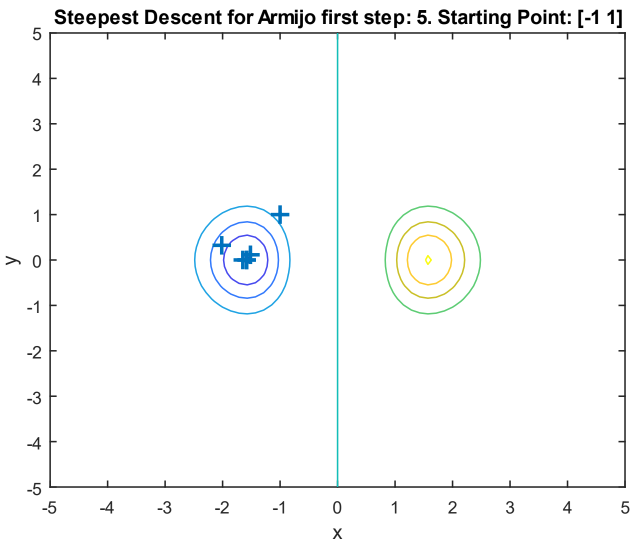
3. Για βήμα μέσω κανόνα Armijo:

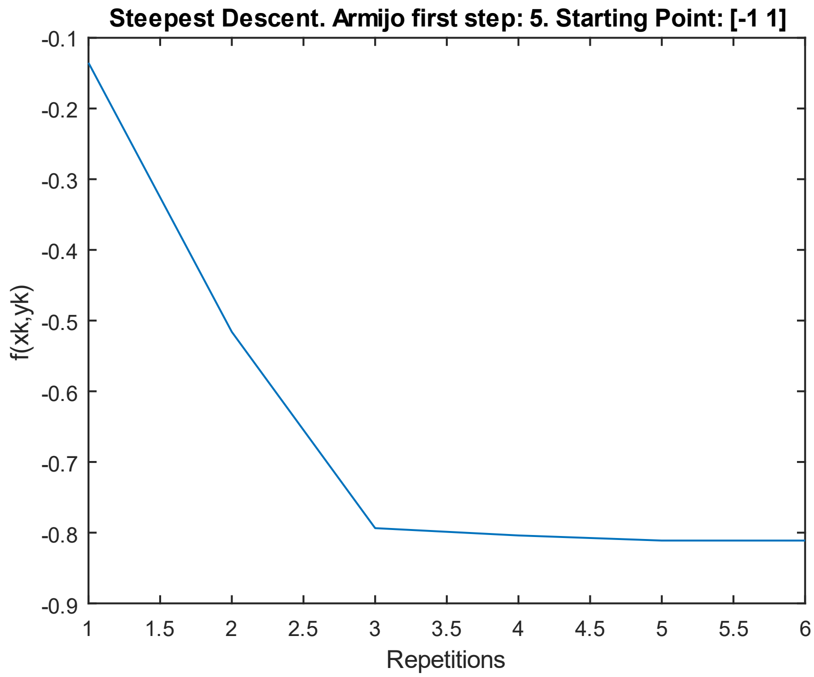
Για τον Armijo διαλέγουμε σταθερά $\alpha = 0.1$, $\beta = 0.5$.

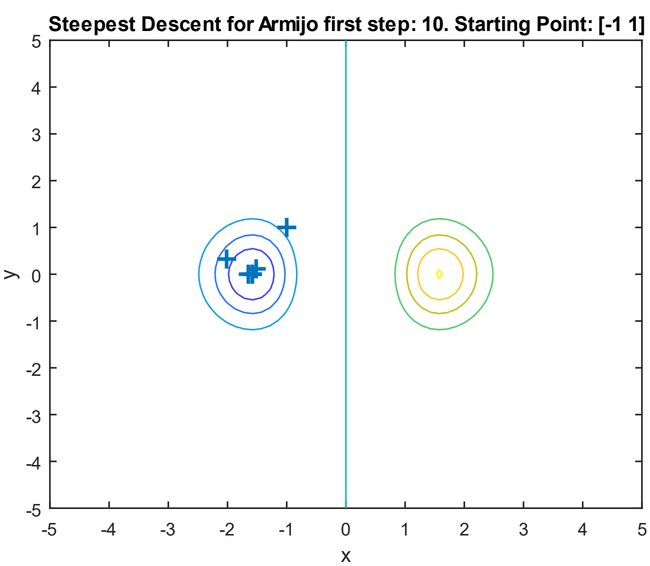
Για αρχικό βήμα s: 1, 5, 10

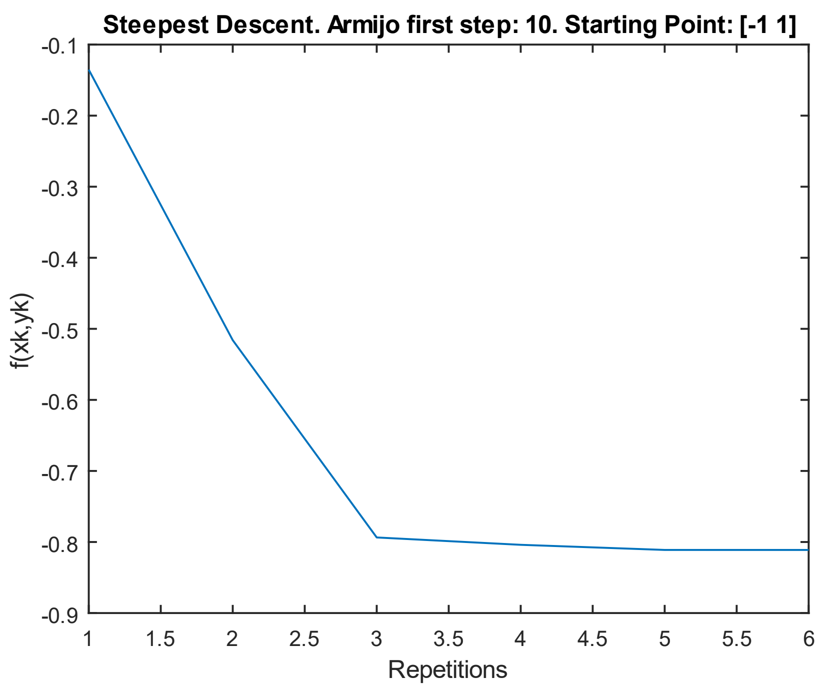










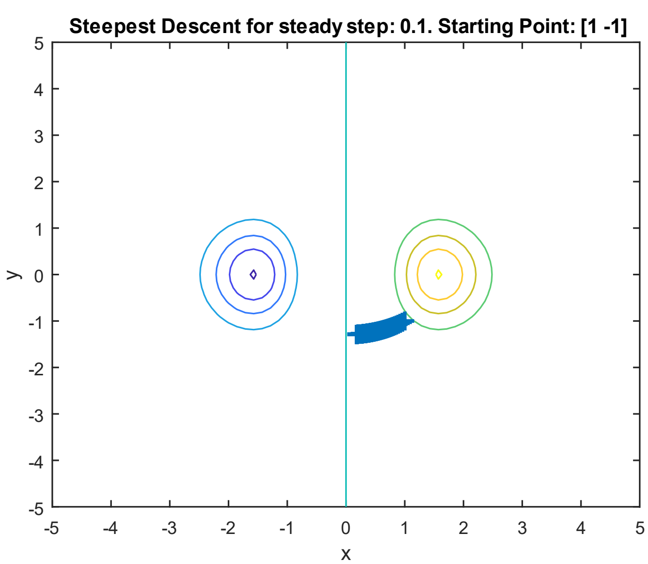


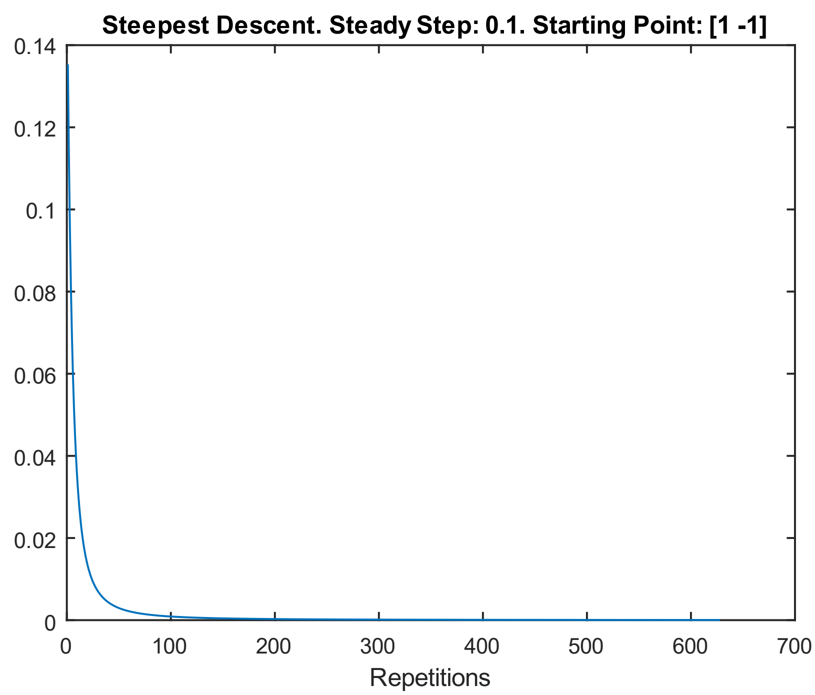
Για μικρό αρχικό βήμα, παρατηρούμε περισσότερες επαναλήψεις σε σχέση με το dixotomouDerivative, ενώ για μεγάλα αρχικά βήματα το ξεπερνάει.

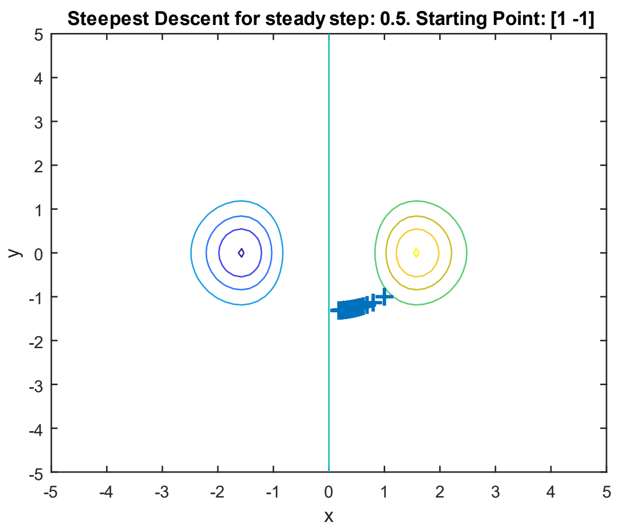
Σχόλια

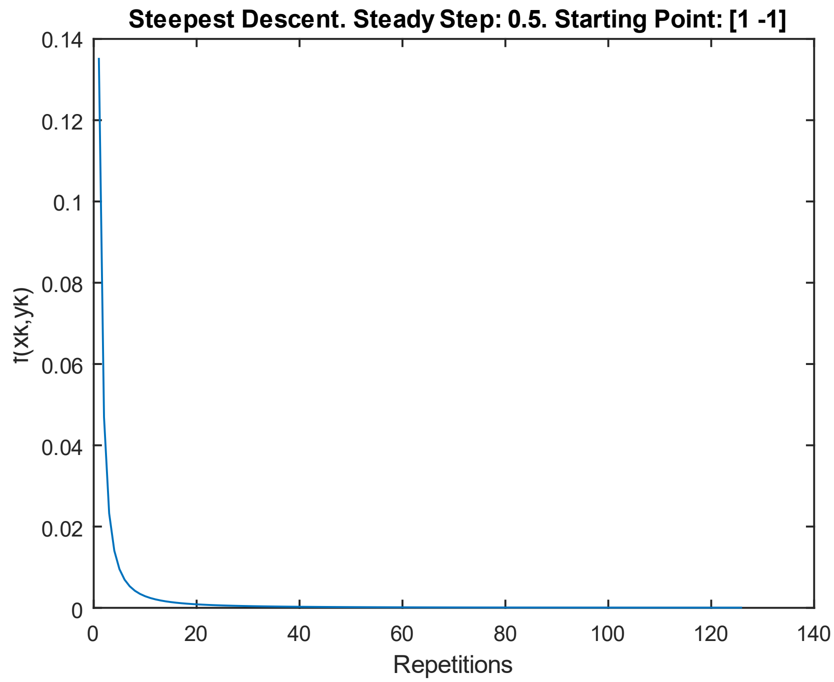
Όλες οι επιλογές των βημάτων πετυχαίνουν την εύρεση του ελαχίστου (πέρα από μεγάλα σταθερά βήματα). Ένας λόγος που συμβαίνει αυτό είναι ότι το αρχικό σημείο μας βρίσκεται αρκετά κοντά σε μεγάλες κλίσεις προς το ολικό ελάχιστο και έτσι αποφεύγει να «κολλήσει» σε τοπικό ελάχιστο, όπως θα δούμε παρακάτω.

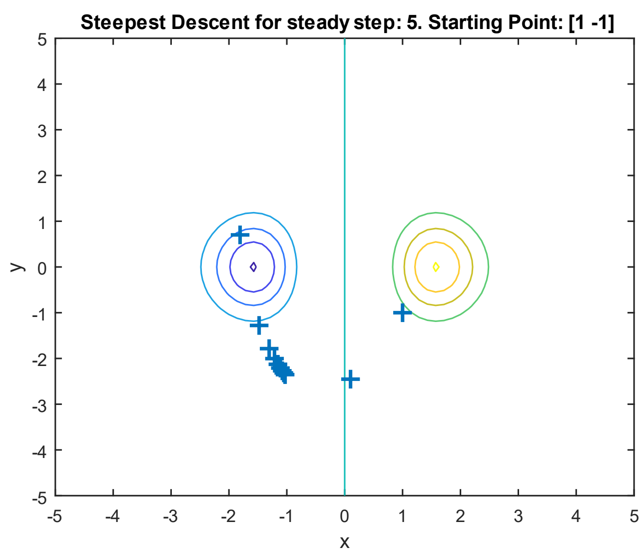
- Σημείο [1, -1]
    1. Για σταθερά βήματα 0.1, 0.5, 5

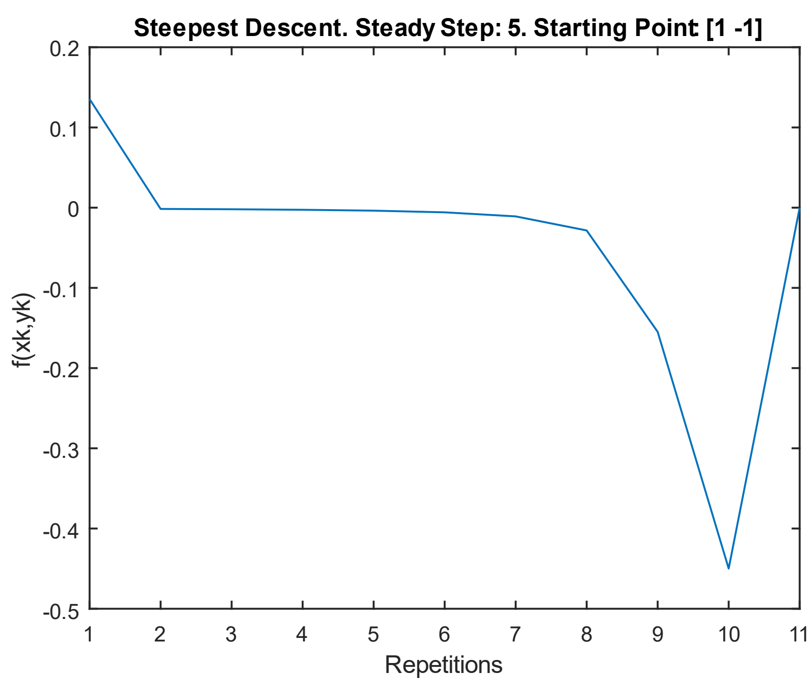






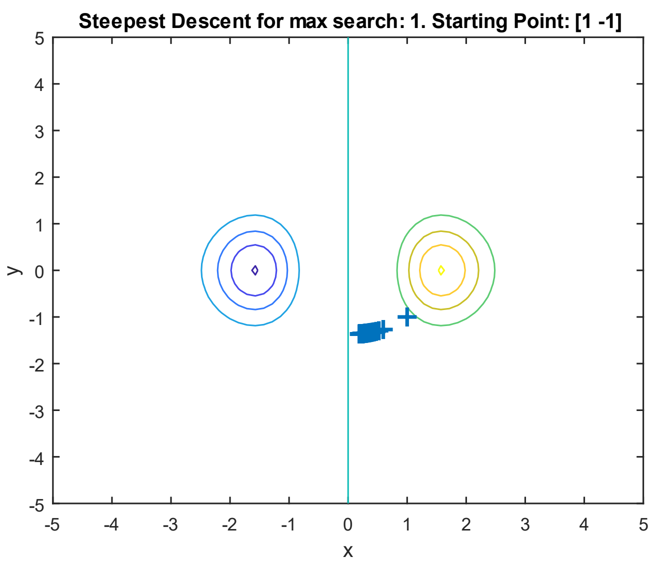


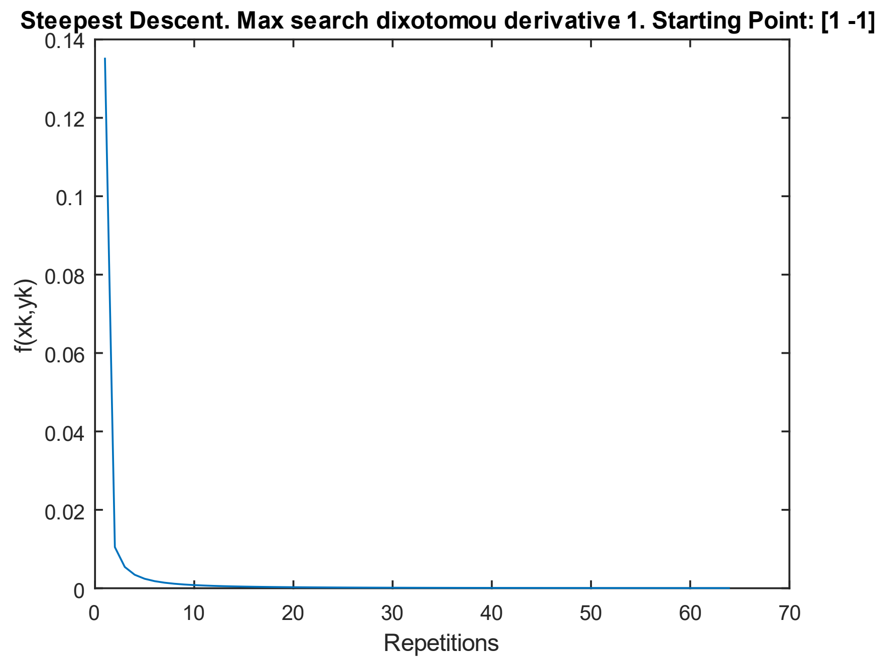


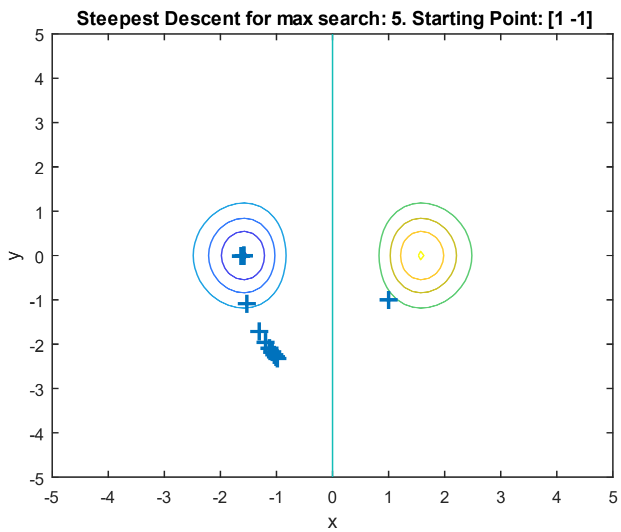


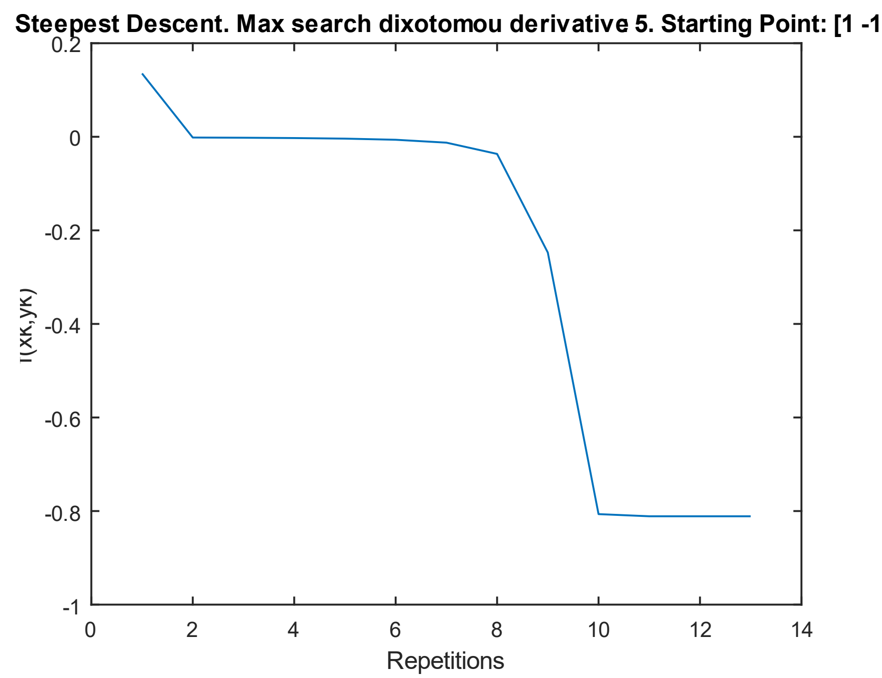
Σε αντίθεση με τον προηγούμενο σημείο, το μικρότερο βήμα δεν βοηθάει σε αυτήν την περίπτωση. Καταρχάς, κανεί πολλούς υπολογισμούς και καταλήγει σε ένα τοπικό ελάχιστο. Για βήμα 5 βλέπουμε πως καταφέρνει να ξεπεράσει αυτό το πρόβλημα αλλά δεν μπορεί να προσεγγίσει το ολικό ακρότατο με ακρίβεια.

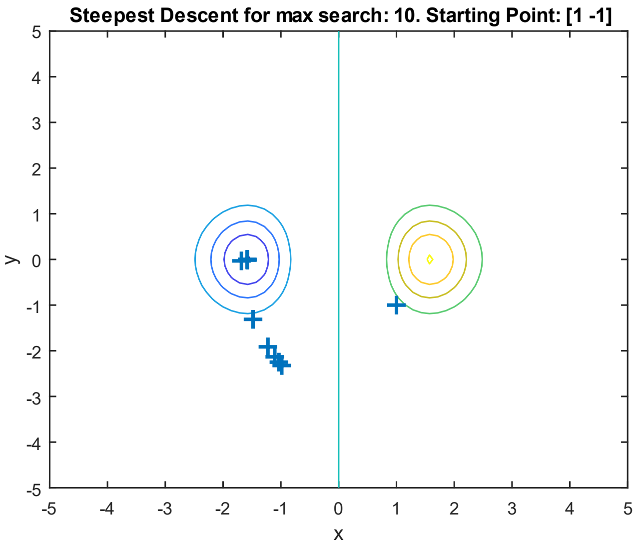
2. Για βήμα μέσω dixotomouDerivative 1, 5, 10:

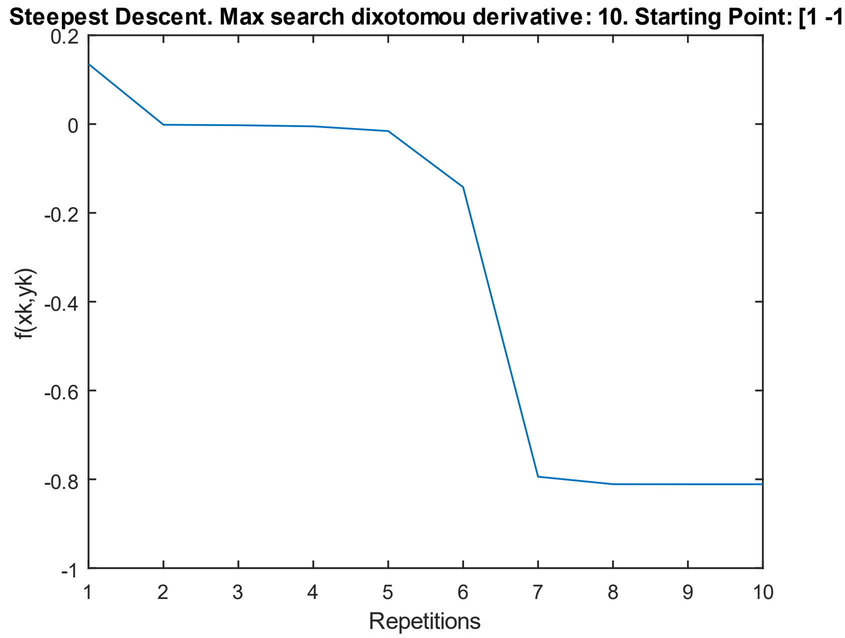






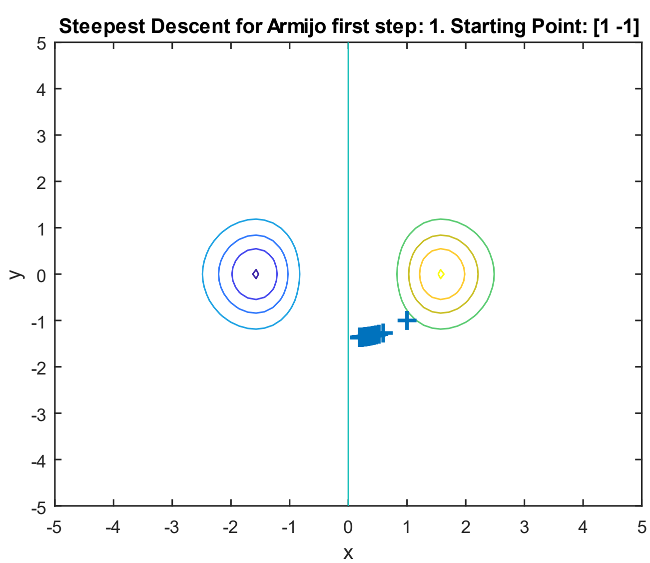


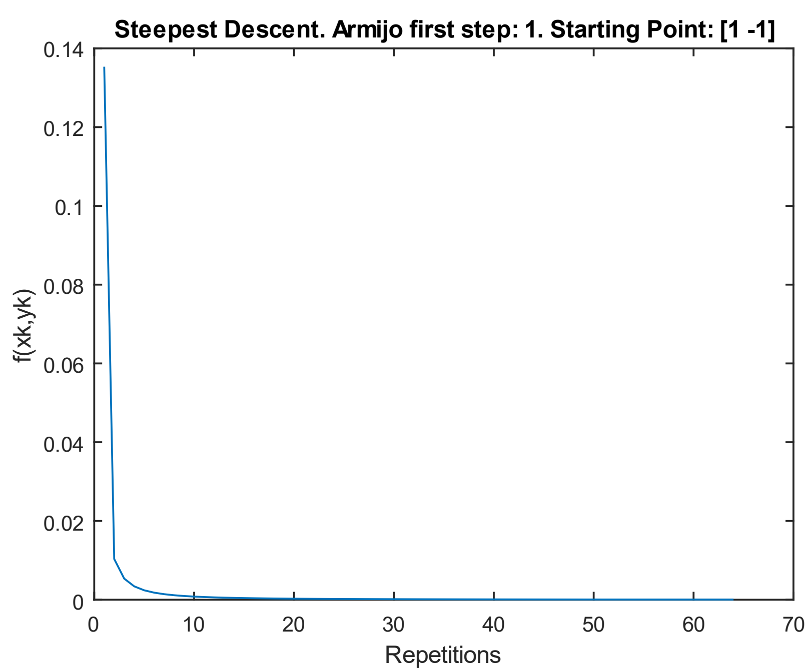


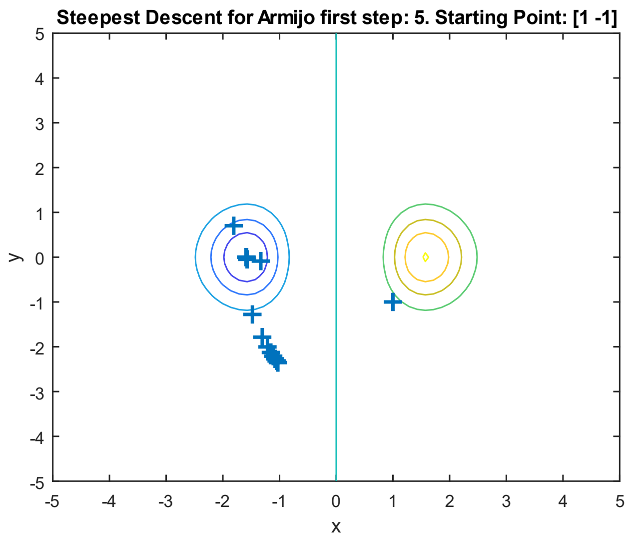


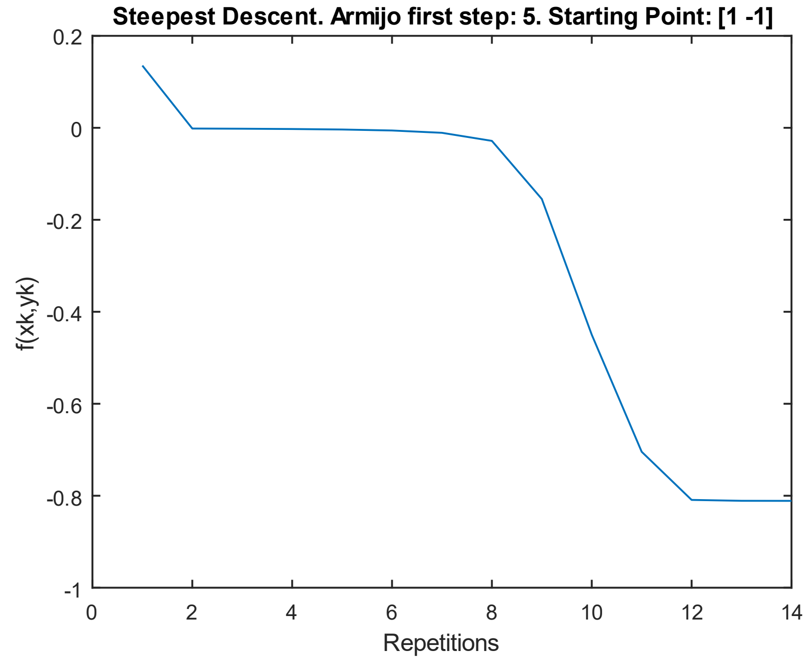
Παρατηρούμε ότι για μικρό μέγιστο δυνατό βήμα, η μέθοδος δεν αγνοεί το τοπικό ελάχιστο. Για μέγιστο δυνατό βήμα 5 και 10 το προσπερνάει και το μεγαλύτερο βρίσκει το ελάχιστο σε λιγότερο χρόνο.

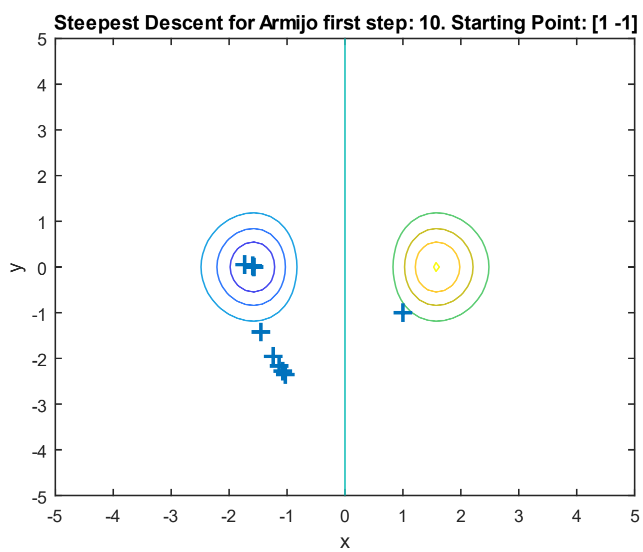
3. Για βήμα μέσω κανόνα Armijo 1, 5, 10:

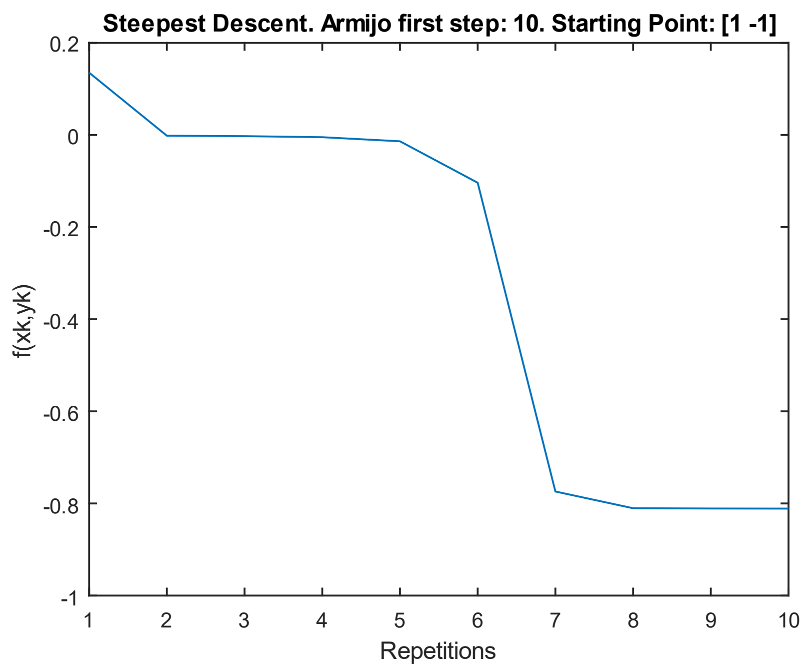












Βλέπουμε πως για μεγάλα αρχικά βήματα, κάνει λιγότερες επαναλήψεις, παρόμοια με την προηγούμενη επιλογή βήματων.

Και ομοίως για μικρό αρχικό βήμα δεν προσπερνάει το τοπικό ελάχιστο.

<u>Σχόλια</u>

Είναι εμφανές ο λόγος που δεν φτάνει πάντα ο αλγόριθμος στο ολικό ελάχιστο, όπως επίσης είναι λογικό ότι με δυναμικό βήμα αρκετά μεγάλο για τον εντοπισμό της κλίσης επιτυγχάνει.

Θέμα 3

Μέθοδος Newton

Αρχεία: newton.m, thema_3.m

Για να πραγματοποιηθεί η υλοποίση της μεθόδου Newton, απαιτείται ο Εσσιανός πίνακας της συνάρτησης να είναι θετικός. Για τα 3 σημεία που έχουμε ο Εσσιανός πίνακας είναι:

- για [0, 0]:

```
[0, 0]
[0, 0]
```

- για [-1, 1]:

```
[-2*exp(-2), -6*exp(-2)]
[-6*exp(-2), -2*exp(-2)]
```

- Για [1, -1]:

```
[2*exp(-2), 6*exp(-2)]
[6*exp(-2), 2*exp(-2)]
```

Για κανένα από τα τρία αρχικά σημεία δεν έχουμε θετικά ορισμένο πίνακα. Το αποτέλεσμα είναι η μέθοδος να μην τερματίζει ή να βγάζει λανθασμένα αποτελέσματα.

Θέμα 4

Μέθοδος Levenberg-Marquardt

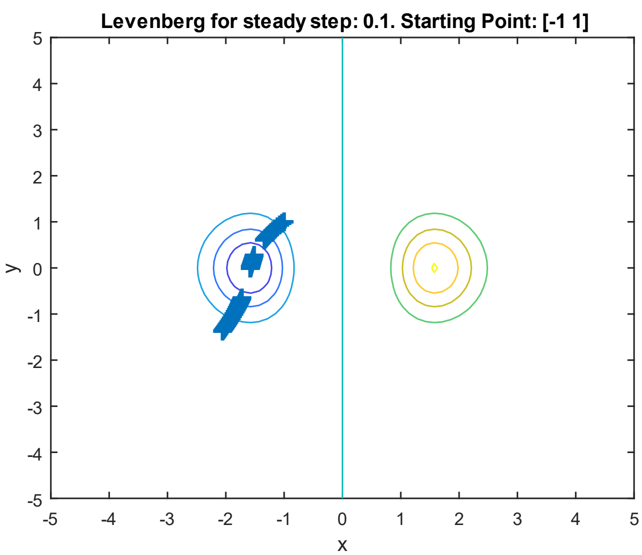
Αρχεία: levenberg.m, thema_4.m

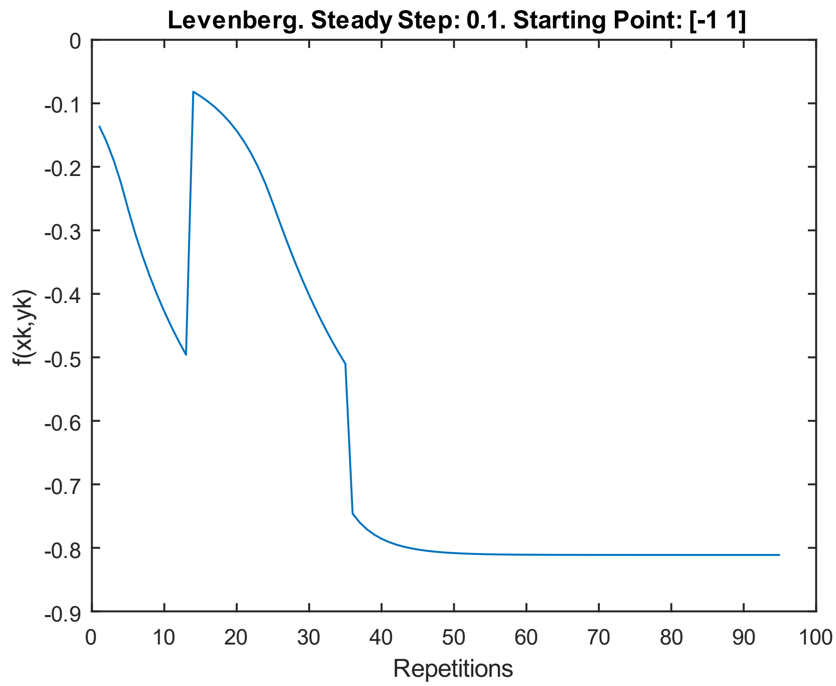
- Σημείο [0, 0]

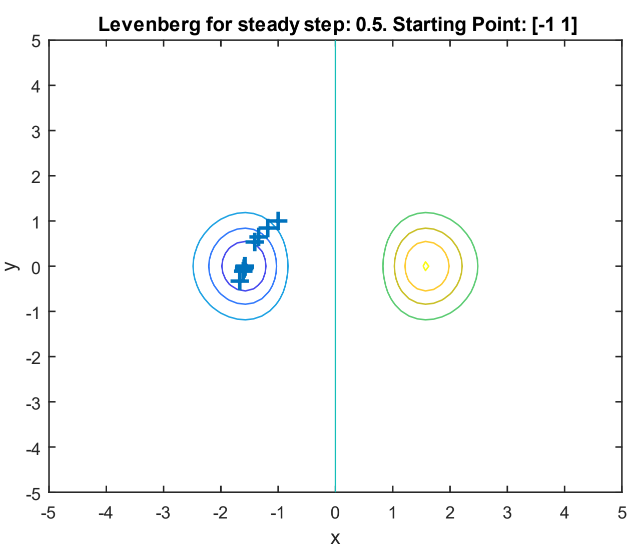
Λόγω της φύσης του αλγορίθμου, το σημείο δεν θα μπει μέσα στον αλγόριθμο γιατί θα έχει κλίση 0, οπότε μένει στο σημείο [0, 0]. Το ίδιο συμβαίνει για οποιοδήποτε βήμα, όπως και στους προηγούμενους αλγορίθμους.

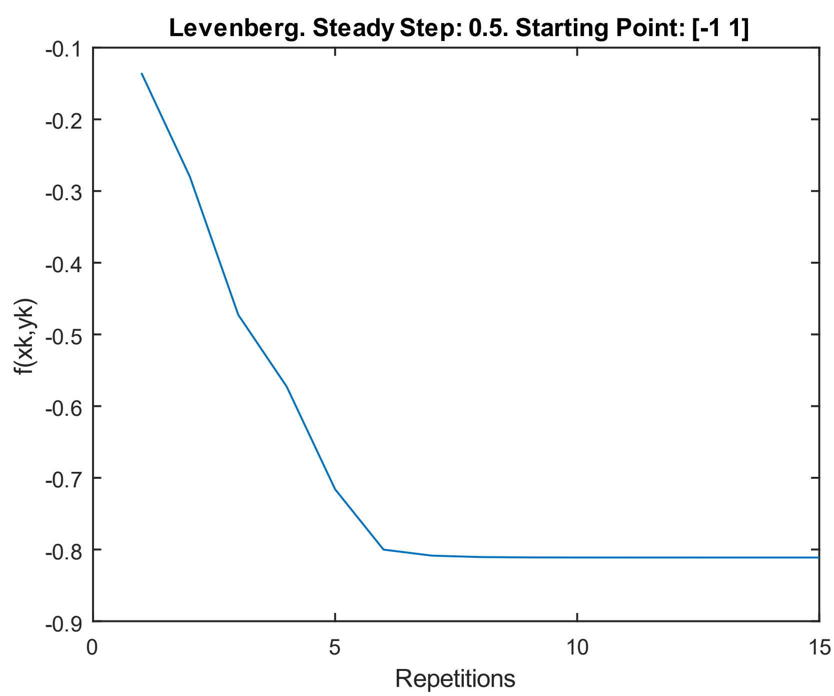
- Σημείο [-1, 1]

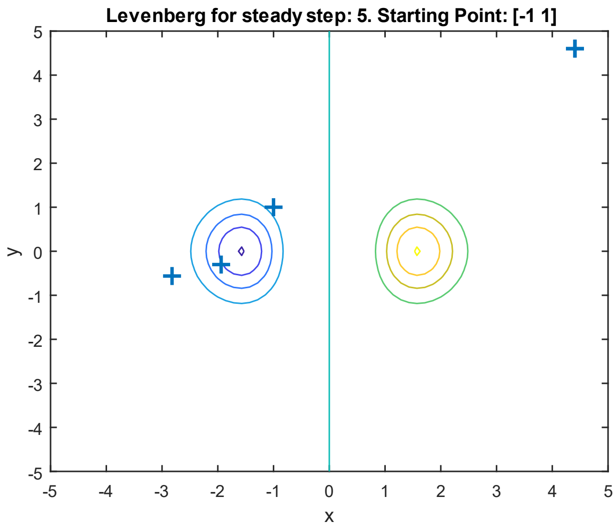
1. Για σταθερό βήμα 0.1, 0.5, 5:

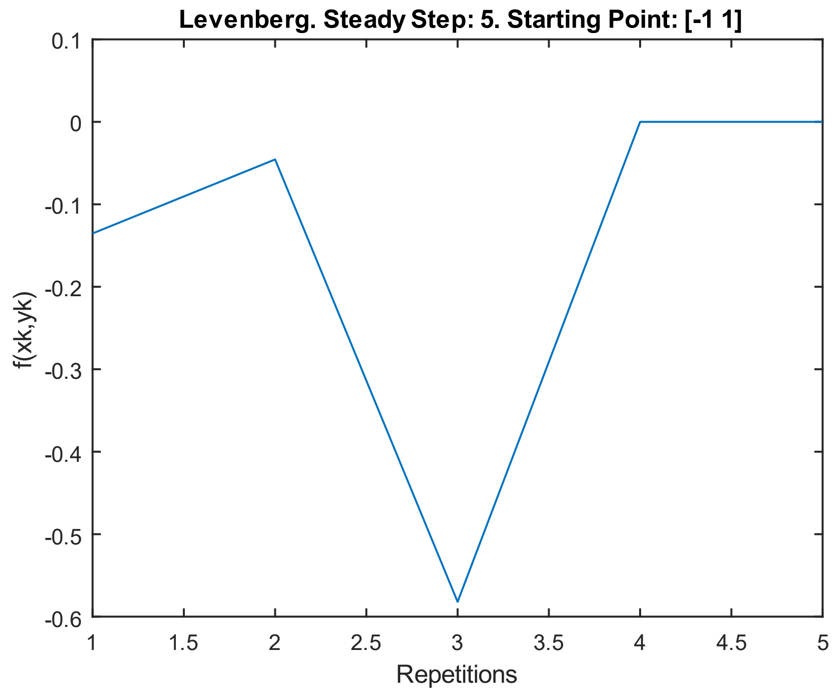






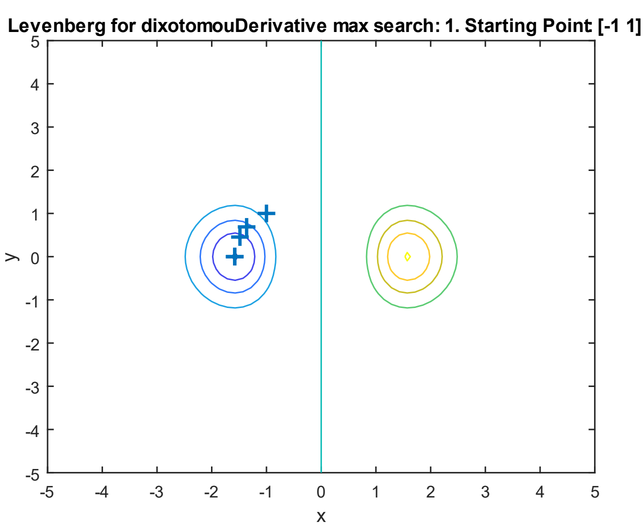


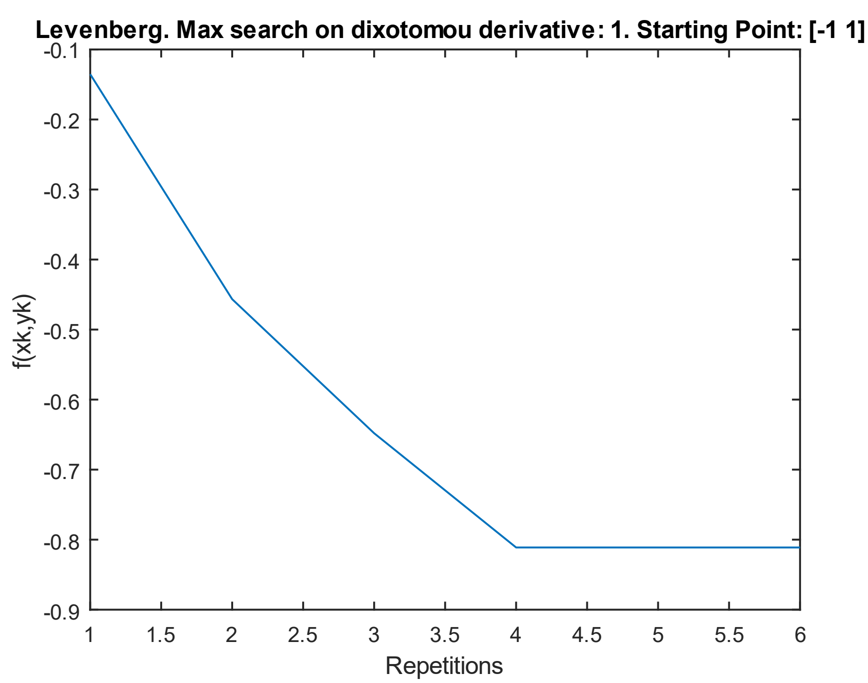


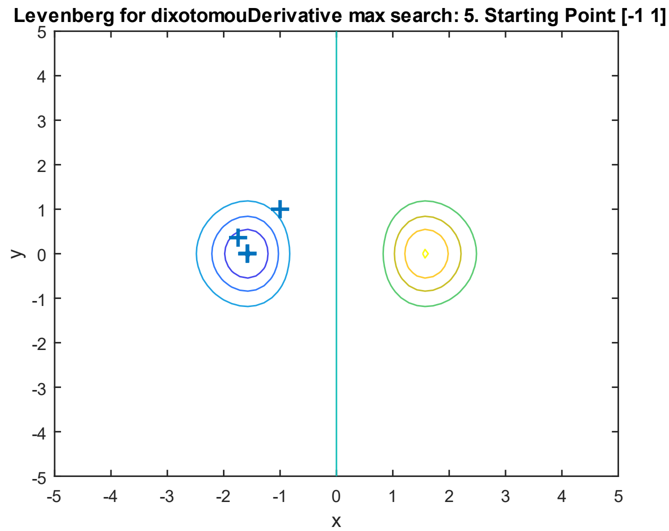


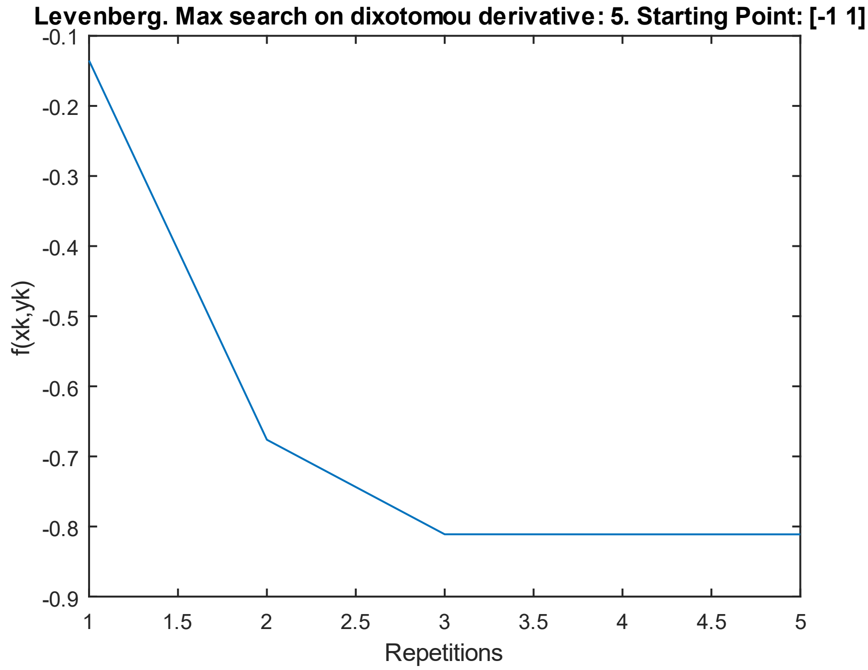
Σύμφωνα με τα παραπάνω διαγράμματα, παρατηρείτε ιδιαίτερα παράξενη κίνηση για μικρά βήματα, που παρόλα αυτά βρίσκει το ακρότατο, και για μεγάλο βήμα φαίνεται πως δεν λειτουργεί σωστά η μέθοδος.

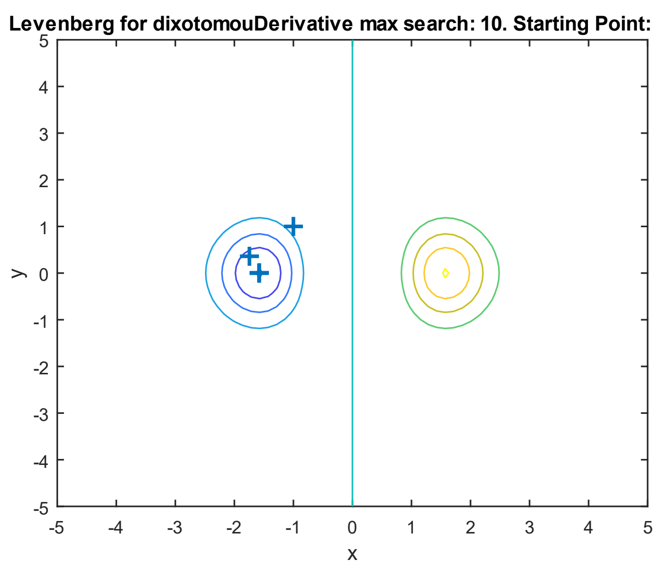
2. Για βήμα μέσω dixotomouDerivative 1, 5, 10:

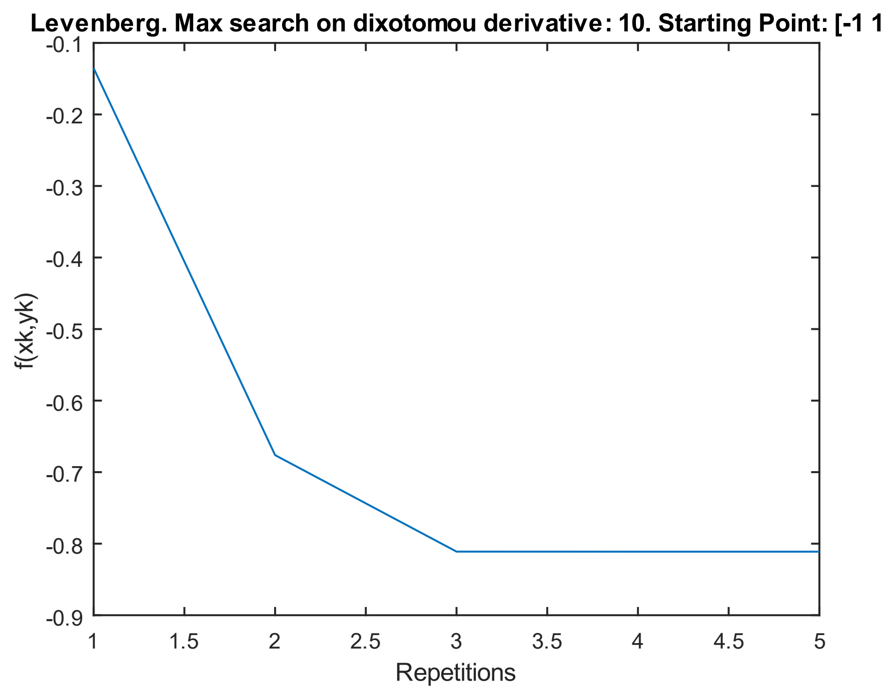






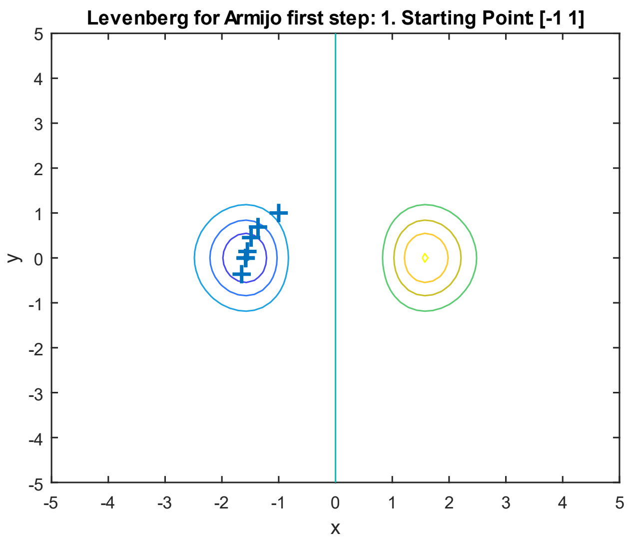


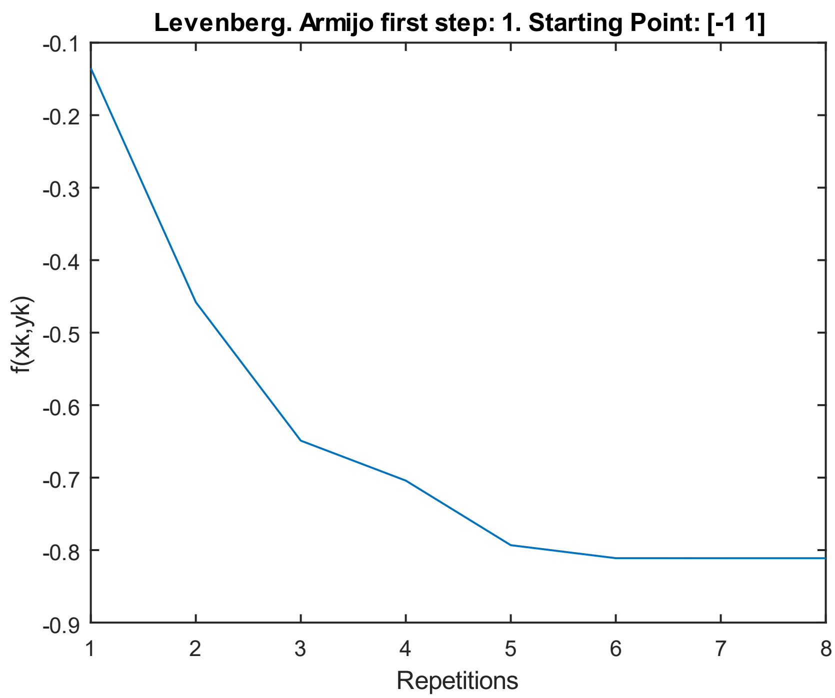


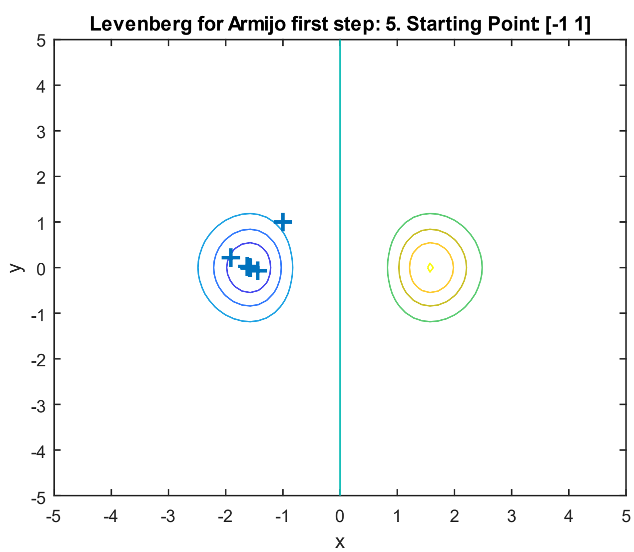


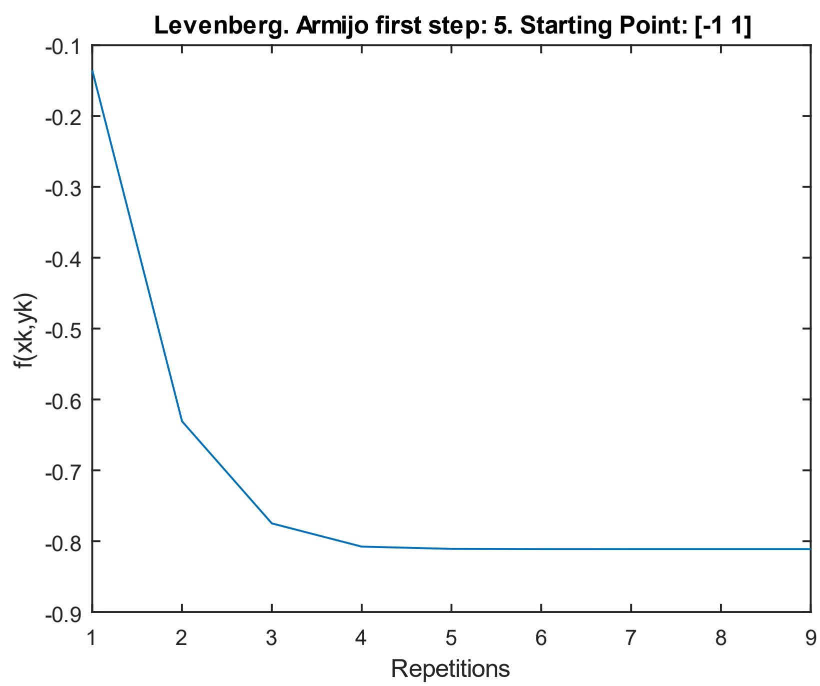
Η ελαχιστοποίηση του βήματος επιτυγχάνει ελάχιστες επαναλήψεις, λιγότερες σε σχέση με την μέθοδο Steepest Descent.

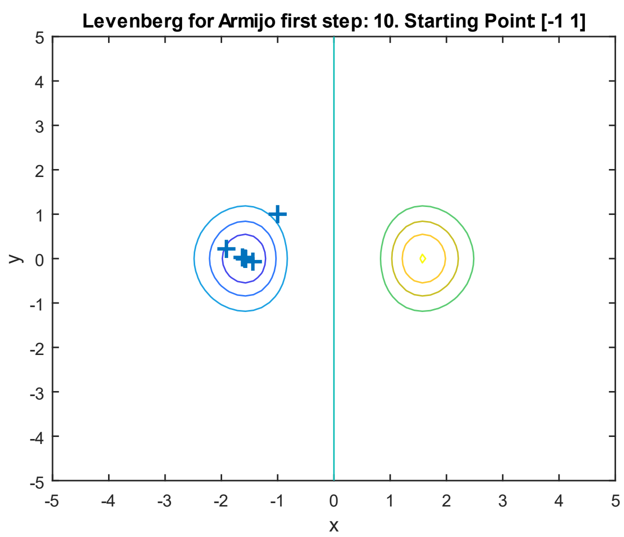
3. Για βήμα μέσω κανόνα Armijo 1, 5, 10:

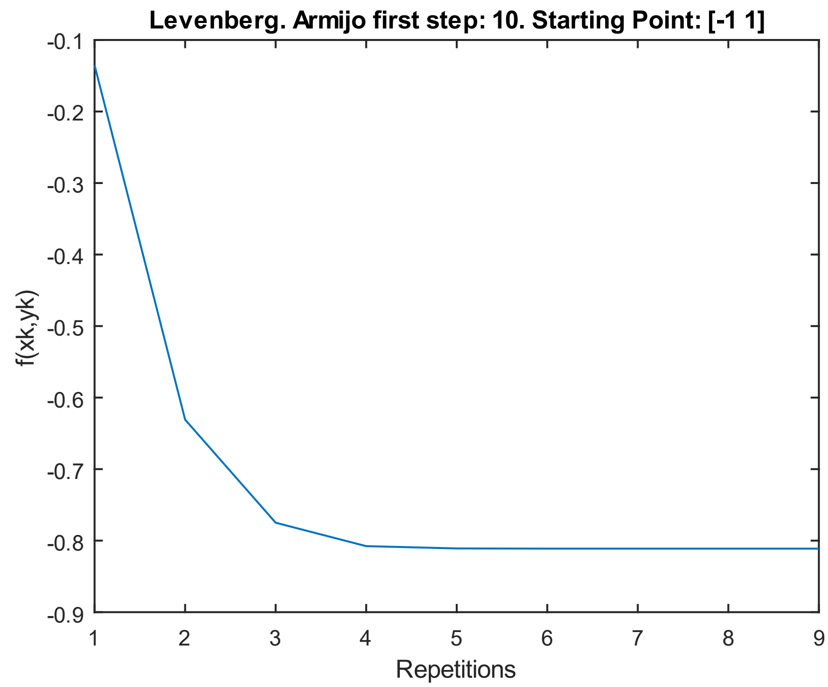






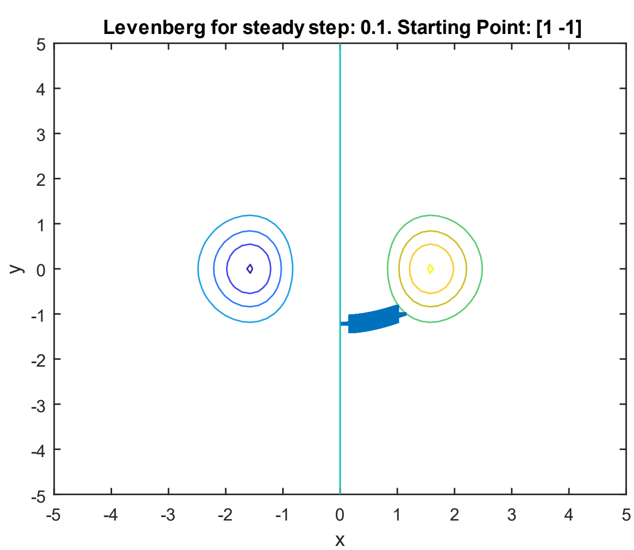


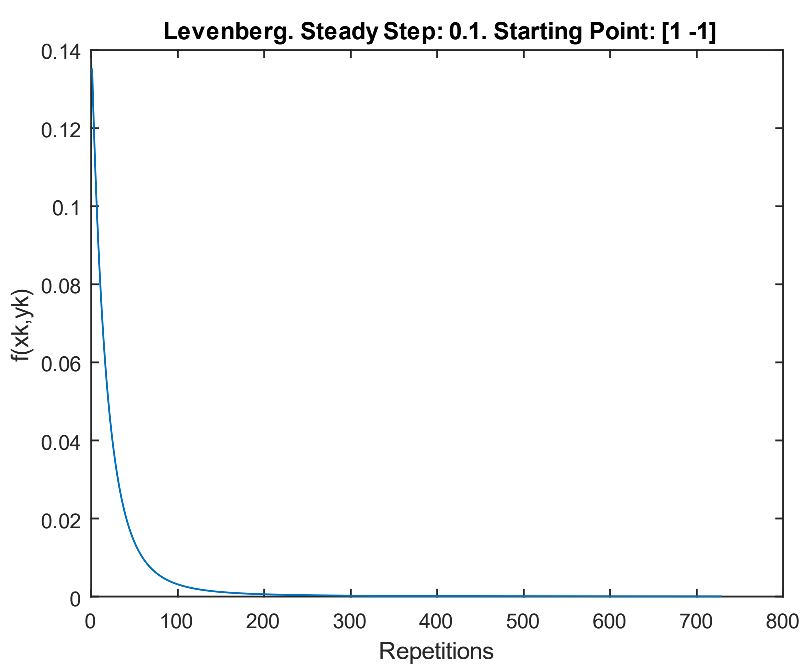


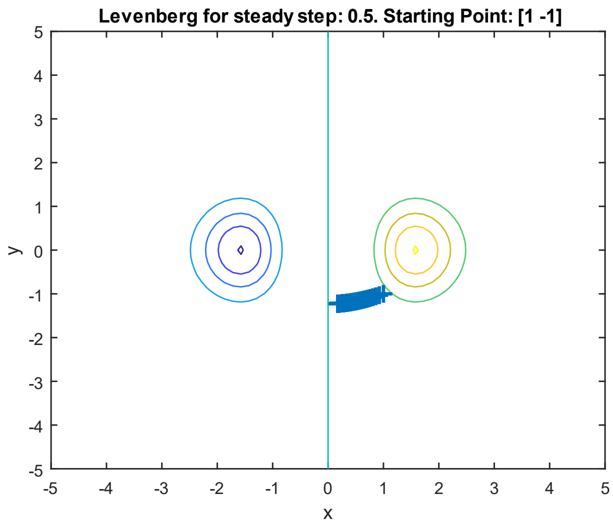


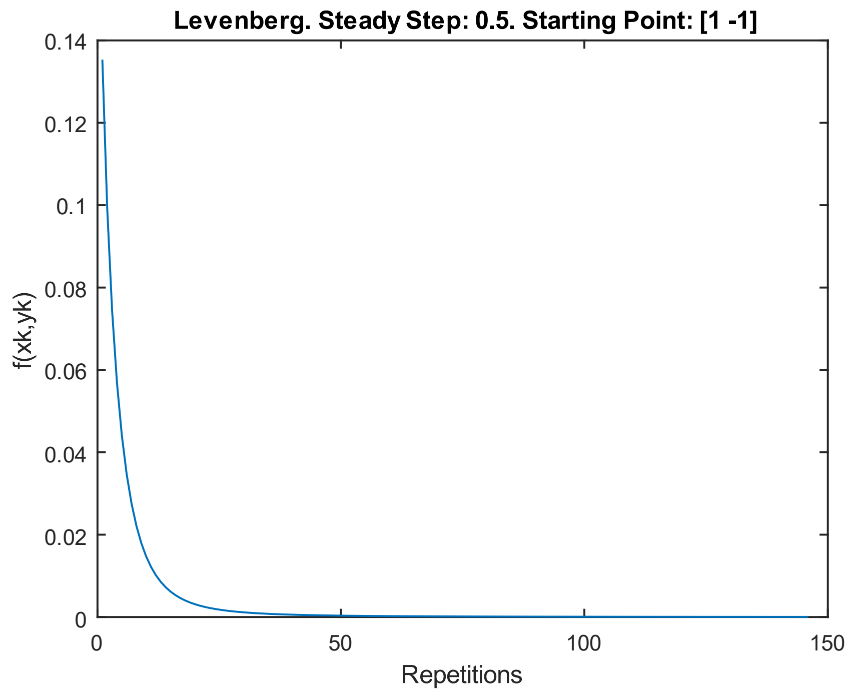
Για μικρό αρχικό βήμα βλέπουμε η μέθοδος να κερδίζει μερικές επαναλήψεις σε σχέση με τα μεγάλα αρχικά βήματα. Επίσης, παίρνει περισσότερο χρόνο απότι στο Steepest Descent.

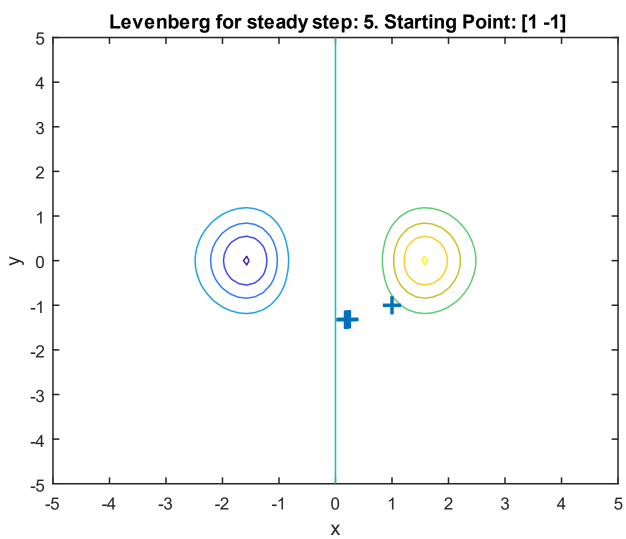
- Σημείο [1, -1]
    1. Για σταθερό βήμα 0.1, 0.5, 5:

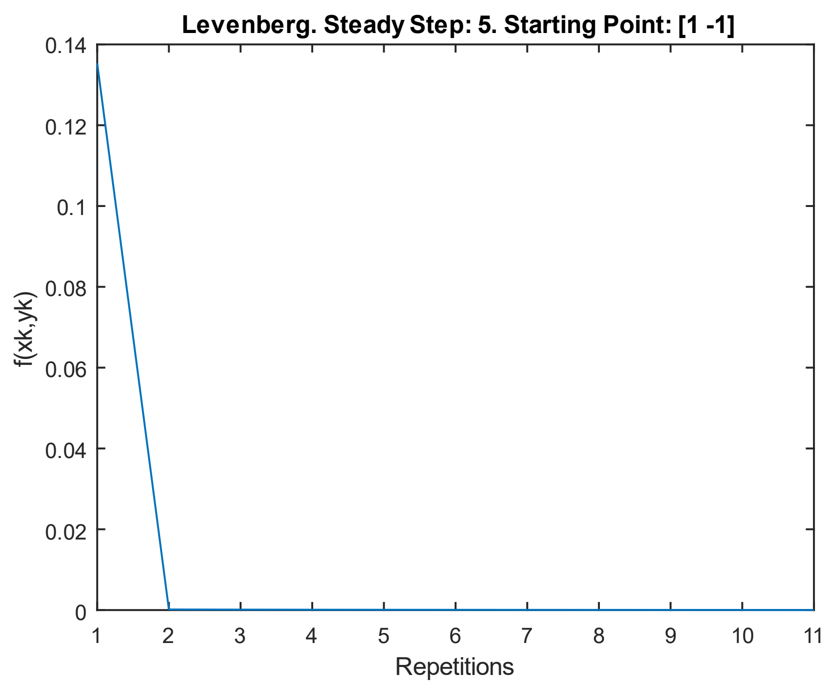






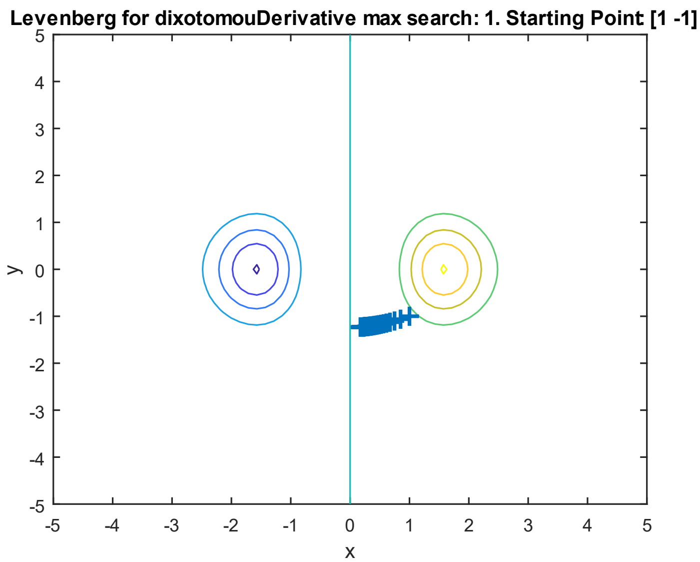


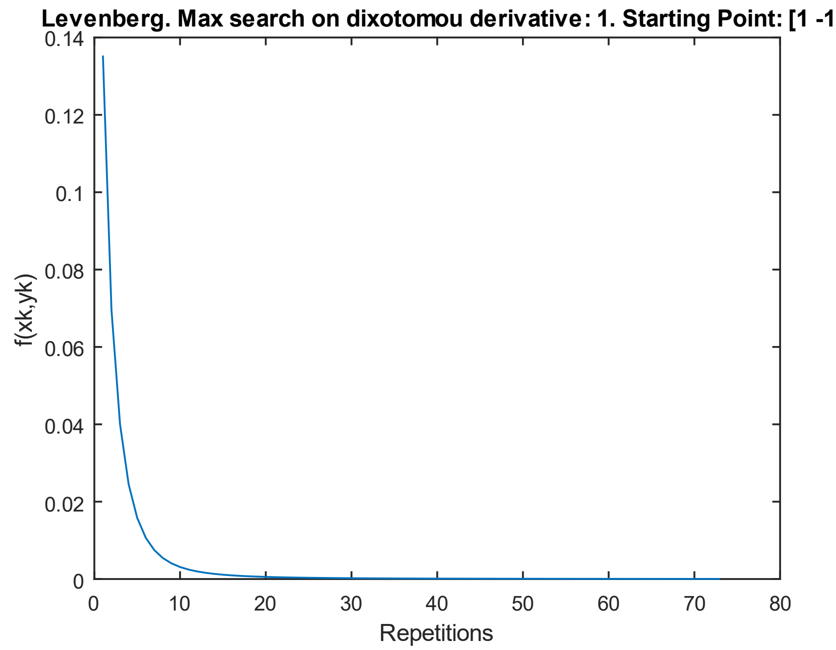


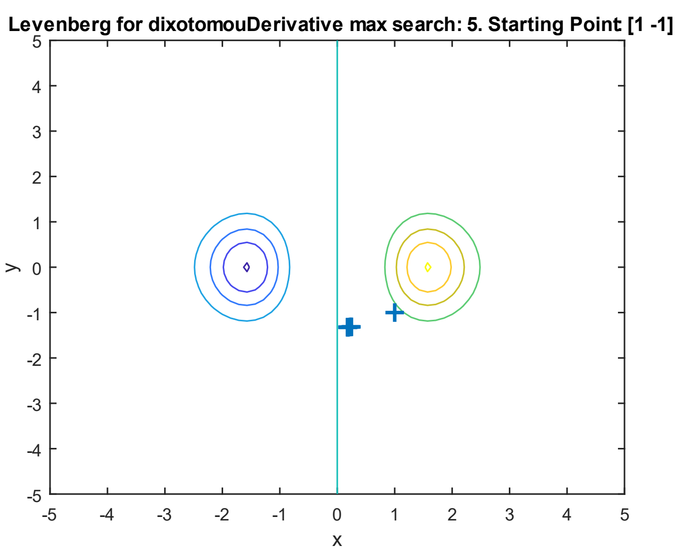


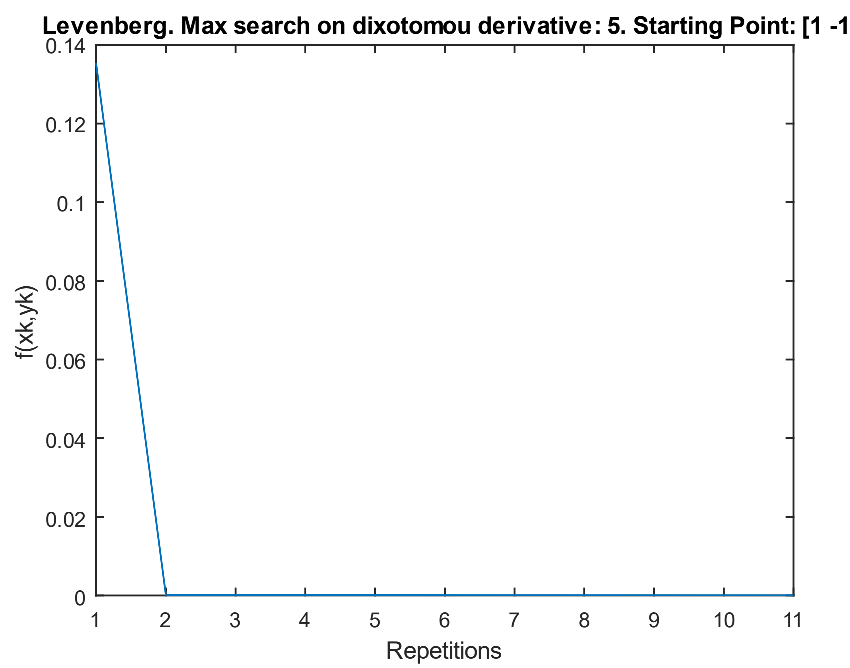
Όπως είναι λογικό, ο αλγόριθμος δεν μπορεί να ξεπεράσει το τοπικό ελάχιστο.

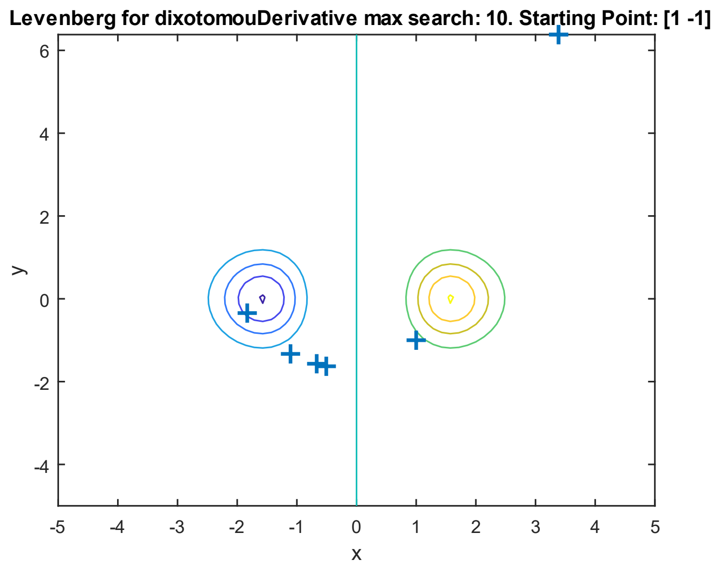
2. Για βήμα μέσω dixotomouDerivative 1, 5, 10, 15:

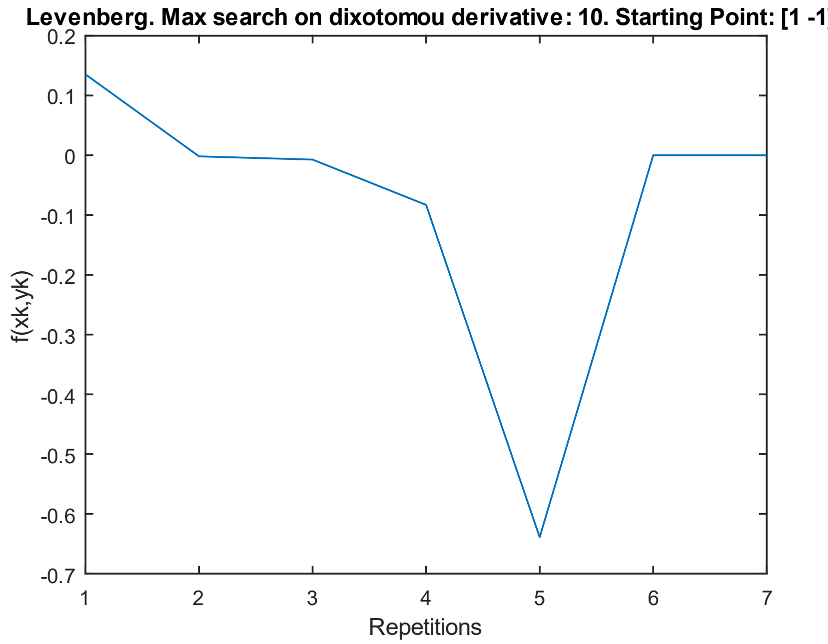


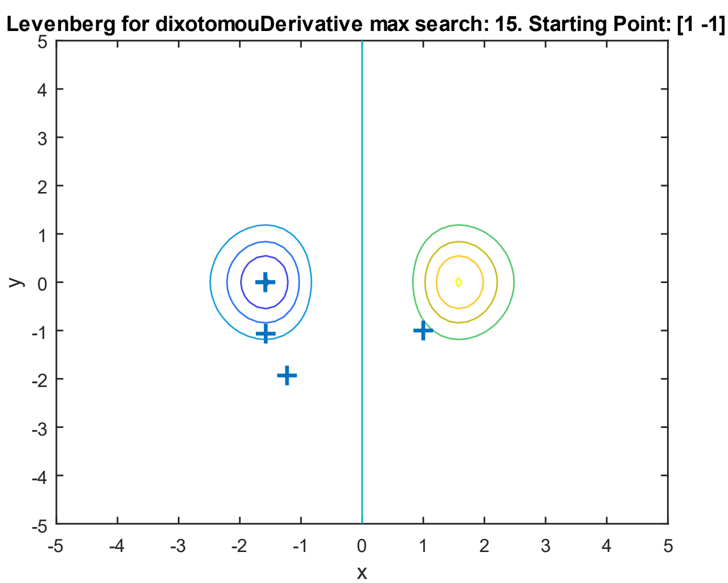


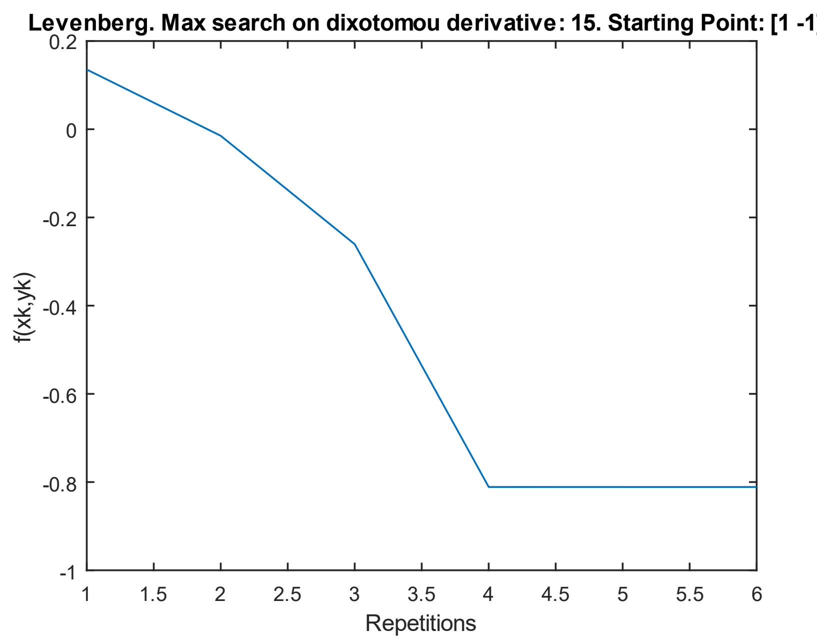






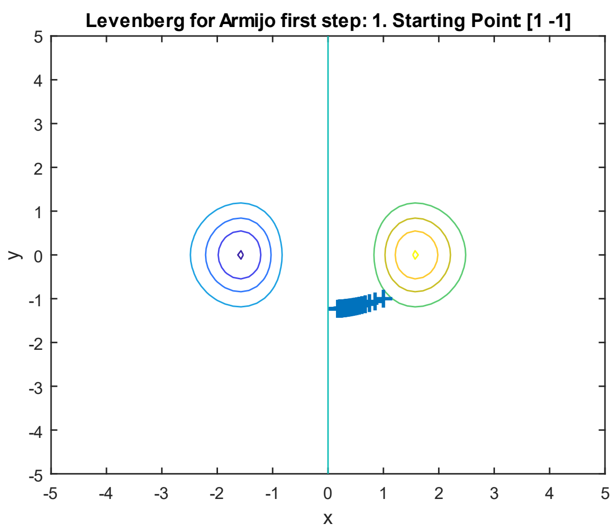


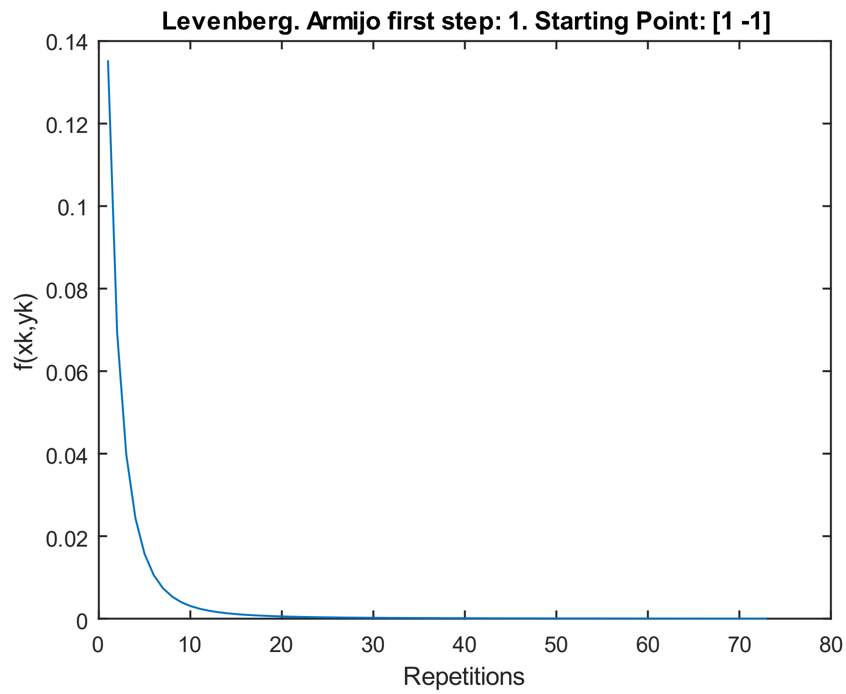


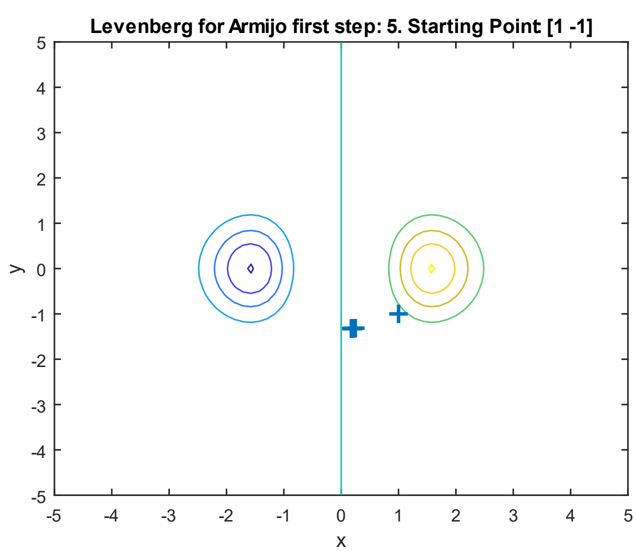


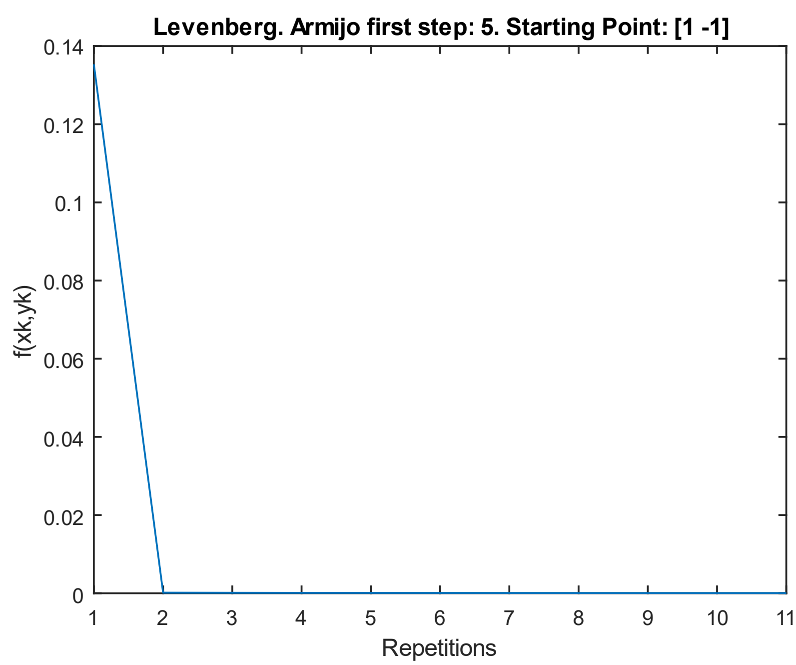
Για την levenberg χρειάζεται αρκετά μεγάλο μέγιστο δυνατό βήμα για να μπορέσει να λειτουργεί σωστά, σε αντίθεση με το θέμα 2. Ακόμα και για 10, καταφέρνει να αγνοήσει το ολικό ελάχιστο, και δεν βγάζει πολύ νόημα.

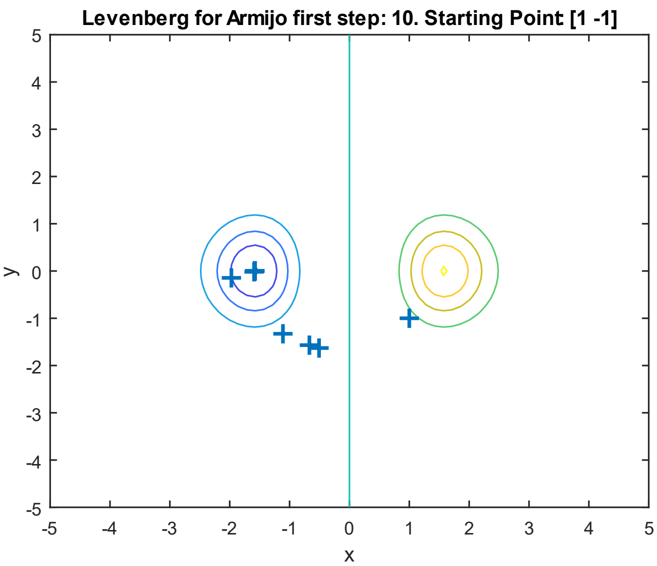
3. Για βήμα μέσω κανόνα Armijo 1, 5, 10:

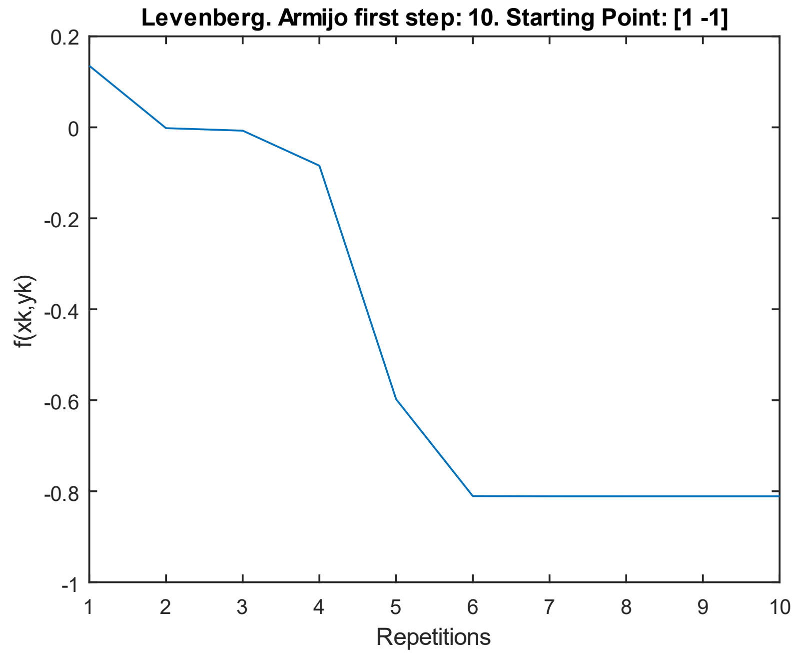












Σε αντίθεση με τον Steepest Descent, που χρειαζόταν αρχικό βήμα 5 για να λειτουργήσει, ο Levenberg χρειάζεται αρχικό βήμα 10. Οι επαναλήψεις όμως είναι ίδιες.

Συμπεράσματα:

Όπως είναι αναμενόμενο, το σταθερό βήμα δεν είναι ιδανικό για εύρεση ελαχίστου καθώς είτε θα εγκλωβίζεται σε τοπικά ακρότατα είτε ,σε περίπτωση που έχει αρκετά μεγάλο βήμα για να τα ξεπεράσει, δεν θα μπορεί να συγκλίνει κοντά στο ολικό ελάχιστο. Ο Newton για αυτήν την συνάρτηση και γιαυτά τα αρχικά σημεία δεν μπορεί να εφαρμοστεί. Ο Steepest Descent και ο Levenberg-Marquardt με την μέθοδο Armijo και την ελαχιστοποίηση του βήματος μέσω της μεθόδου διχοτόμου με χρήση παραγώγων δείχνουν να λειτουργούν για αρκετά μεγάλα βήματα, με σχετικά παρόμοιες επαναλήψεις. Από άποψη χρόνου, ο Steepest Descent φαίνεται να τρέχει πιο γρήγορα και δεν χρειάζεται και μεγάλα βήματα για να επιτύγχει.

Όλες οι μέθοδοι είναι για μέτρο κλίσης > epsilon = 0.001.