Μέθοδοι Quasi-Newton

Σταύρος Νικολαΐδης
ΑΕΜ: 3975

Μάιος 2024

1 Εισαγωγή

Ένα βασικό μειονέκτημα των κλασσικών μεθόδων Newton είναι ότι απαιτούν τον υπολογισμό του Εσσιανού πίνακα $\nabla^2 f(x_k)$σε κάθε βήμα k. Για την αντιμετώπιση αυτού του προβλήματος έχουν αναπτυχθεί οι μέθοδοι Quasi-Newton, οι οποίες είναι ιδιαίτερα δημοφιλείς για προβλήματα Μη Γραμμικής Βελτιστοποίησης. Αντί να υπολογίσουμε τον ακριβή Εσσιανό πίνακα, με τις μεθόδους Quasi-Newton ενημερώνουμε μια προσέγγιση του Εσσιανού πίνακα σε κάθε επανάληψη. Επίσης, είναι ενσωματομένες σε πολλά εμπορικά προγράμματα και χρησιμοποιούνται κυρίως για την επίλυση προβλημάτων μεσαίου μεγέθους καθώς απαιτούν αρκετή μνήμη για την αποθήκευση των πινάκων.

2 Quasi-Newton Αλγόριθμος

1. Αρχικοποίηση: Θέτουμε μια πρώτη προσέγγιση της λύσης x_0 και μια πρώτη προσέγγιση για τον Εσσιανό Πίνακα B_0 (π.χ. $B_0 = I$)

2. Κύριο Βήμα: για k = 0, 1...

   (a) Αν το x_k είναι βέλτιστο, ο αλγόριθμος σταματάει. [1]

   (b) Λύνουμε $B_k p = -\nabla f(x_k)$ για p_k.

   (c) Με μια μέθοδο Αναζήτησης Ευθείας υπολογίζουμε $x_{k+1} = x_k + a_k p_k$

   (d) Υπολογίζουμε

   $$s_k = x_{k+1} - x_k$$

   $$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

   (e) Υπολογίζουμε $B_{k+1} = B_k + ...$ χρησιμοποιώντας έναν τύπο ανανέωσης ανάλογα το είδος της Quasi-Newton Μεθόδου.

   (f) Επαναλαμβάνουμε για k+1

[1] Είναι σημαντικό να αναφέρουμε πως για το κρητήριο τερματισμού του αλγορίθμου μπορεί να χρησιμοποιηθούν διάφορες τεχνικές, εμείς όμως θα τερματίζουμε τον αλγόριθμο όταν το μέτρο της παραγώγου για το x_k γίνει μικρότερο από μια σταθερά ανοχής.

3 Μέθοδος BFGS

Ο τύπος ανανέωσης της μεθόδου BFGS είναι:

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

3.1 Υλοποίηση BFGS σε Κώδικα

```python
import numpy as np
from scipy.optimize.linesearch import line_search_wolfe2

def bfgs(f, grad, x0, max_iter=100, tol=1e-5, epsilon=1e-8):
    x_k = x0
    n = len(x0)
    B_k = np.eye(n)  # Initial Hessian approximation
    steps = 0
    x_values = [x_k.copy()]
    f_values = [f(x_k)]
    print(f"Step {steps}: x = {x_k}, f(x) = {f(x_k)}")
    for i in range(max_iter):
        grad_k = grad(x_k)
        # Stopping criterion
        if np.linalg.norm(grad_k) < tol:
            print()
            break

        # Perform Cholesky decomposition of B_k
        try:

            # Perform Cholesky decomposition
            L = np.linalg.cholesky(B_k)

            # Define L transpose
            L_T = L.T

            # Solve Ly = -gradient using forward substitution
            y = np.linalg.solve(L, -grad_k)

            # Solve L_T * p = y using backward substitution
            p_k = np.linalg.solve(L_T, y)


        except np.linalg.LinAlgError:
            print("Cholesky decomposition failed. Using gradient descent direction.")
            p_k = -grad_k

        # Line search method to find step size alpha
        alpha_k = line_search_wolfe2(f, grad, x_k, p_k)[0]
        if alpha_k is None:
            alpha_k = 1e-4  # Default to a smaller step size if line search fails

        # Update x_k
```

```
        x_k1 = x_k + alpha_k * p_k

        # Compute s_k and y_k
        s_k = x_k1 - x_k
        y_k = grad(x_k1) - grad_k

        s_k = s_k.reshape(-1, 1)
        y_k = y_k.reshape(-1, 1)

        Bksk = B_k @ s_k
        skT_Bk_sk = s_k.T @ B_k @ s_k
        ykT_sk = y_k.T @ s_k

        # Add conditions to prevent division by zero or small values
        if skT_Bk_sk < epsilon:
            skT_Bk_sk = epsilon
        if ykT_sk < epsilon:
            ykT_sk = epsilon

        term1 = (Bksk @ Bksk.T) / skT_Bk_sk
        term2 = (y_k @ y_k.T) / ykT_sk

        B_k = B_k - term1 + term2

        # Update x_k for the next iteration
        x_k = x_k1

        steps += 1
        x_values.append(x_k.copy())
        f_values.append(f(x_k))
        print(f"Step {steps}: x = {x_k}, f(x) = {f(x_k)}")
    return x_k, steps, x_values, f_values
```

Listing 1: BFGS Implementation

Για την υλοποίηση σε κώδικα, εκμεταλλευόμαστε την βολικότητα της μεθόδου αποσύνθεσης Cholesky ώστε να βρούμε την κατεύθυνση p_k. Επίσης κάνουμε χρήση μιας έτοιμης μεθόδου Αναζήτησης Ευθείας για να προσδιορίζουμε το βήμα a_k για κάθε επανάληψη.

3.2 BFGS σε Αντικειμενικές Συναρτήσεις

Θα δοκιμάσουμε την μέθοδο BFGS σε διάφορες αντικειμενικές συναρτήσεις για προβλήματα **Ελαχιστοποίησης** ώστε να συγκρίνουμε την σύγκλιση της.

3.2.1 Αντικειμενική Συνάρτηση Quadratic Bowl

Θα χρησιμοποιήσουμε την συνάρτηση Quadratic Bowl:

$$x_0^2 + x_1^2$$

Η οποία έχει παραγώγους πρώτης τάξης:

$$f_{x_0} = 2 * x_0$$

$$f_{x_1} = 2 * x_1$$

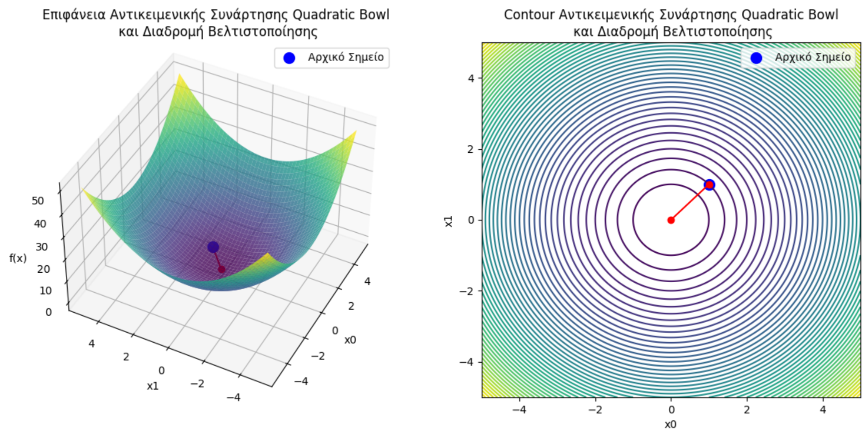


Figure 1: Μονοπάτι Βελτιστοποίησης και Contour της Αντικειμενικής Συνάρτησης Quadratic Bowl (Μέθοδος BFGS)

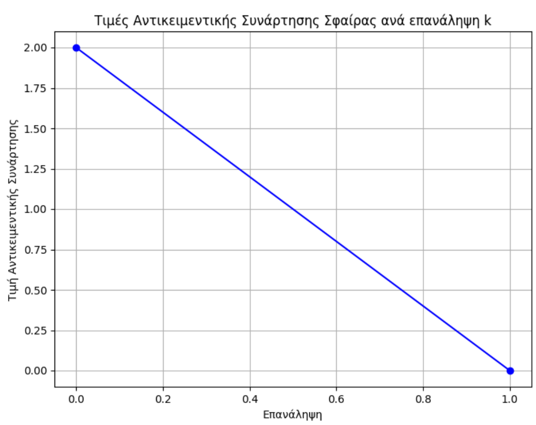
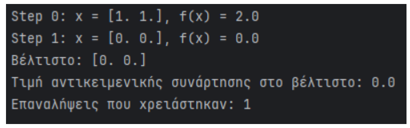


Figure 2: Τιμές Αντικειμενικής Συνάρτησης Quadratic Bowl ανά επανάλαψη k (Μέθοδος BFGS)

Όπως παρατηρούμε στα Figures 1 έως 3 η μέθοδος συγκλίνει σε ένα βήμα στον ολικό ελαχιστοποιητή $x^* = (0, 0)$, πράγμα αναμενόμενο καθώς έχουμε να κάνουμε με μια τετραγωνική συνάρτηση.



```
Step 0: x = [1. 1.], f(x) = 2.0
Step 1: x = [0. 0.], f(x) = 0.0
Βέλτιστο: [0. 0.]
Τιμή αντικειμενικής συνάρτησης στο βέλτιστο: 0.0
Επαναλήψεις που χρειάστηκαν: 1
```

Figure 3: Επαναλήψεις για προσσέγγιση βελτίστου Αντικειμενικής Συνάρτησης Quadratic Bowl (Μέθοδος BFGS)

3.2.2 Αντικειμενική Συνάρτηση Himmelblau

Θα χρησιμοποιήσουμε την συνάρτηση Himmelblau:

$$f(x_0, x_1) = (x_0^2 + x_1 - 11)^2 + (x_0 + x_1^2 - 7)^2$$

Η οποία έχει παραγώγους πρώτης τάξης:

$$f_{x_0} = 4x_0(x_0^2 + x_2 - 11) + 2(x_0 + x_1^2 - 7)$$
$$f_{x_1} = 2(x_0^2 + x_1 - 11) + 4x_2(x_0 + x_1^2 - 7)$$

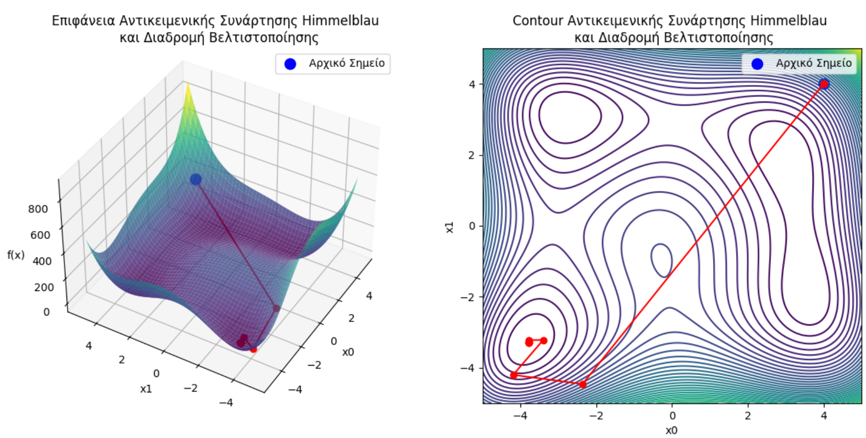


Figure 4: Μονοπάτι Βελτιστοποίησης και Contour της Αντικειμενικής Συνάρτησης Himmelblau (Μέθοδος BFGS)

Παρατηρούμε στα Figures 4 έως 6 ότι η μέθοδος με αρχικό σημείο το $(4, 4)$ συγκλίνει στον τοπικό βελτιστοποιητή $x^* = (-3.77931027, -3.28318606)$ μετά από 9 βήματα και όχι μετά από ένα καθώς η συνάρτηση Himmelblau δεν είναι τετραγωνική. Παρατηρούμε επίσης ότι στην αρχή κάνει ένα αρκετά μεγάλο βήμα, πράγμα που οφείλεται στην Μέθοδο Αναζήτησης Ευθείας η οποία προσδιορίζει το βήμα σε κάθε επανάληψη.

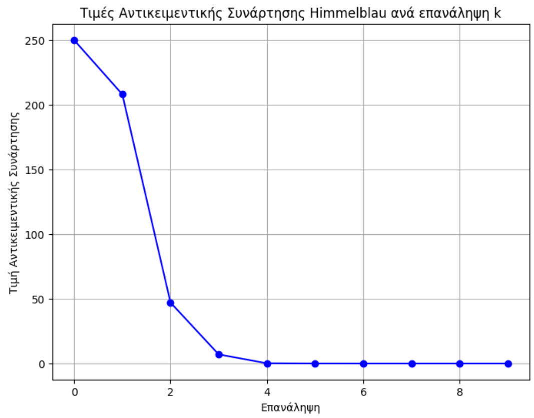
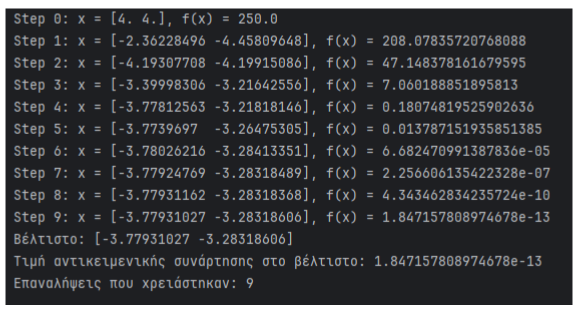


Figure 5: Τιμές Αντικειμενικής Συνάρτησης Himmelblau ανά επανάλαψη k (Μέθοδος BFGS)



```
Step 0: x = [4. 4.], f(x) = 250.0
Step 1: x = [-2.36228496 -4.45809648], f(x) = 208.07835720768088
Step 2: x = [-4.19307708 -4.19915086], f(x) = 47.148378161679595
Step 3: x = [-3.39998306 -3.21642556], f(x) = 7.060188851895813
Step 4: x = [-3.77812563 -3.21818146], f(x) = 0.18074819525902636
Step 5: x = [-3.7739697  -3.26475305], f(x) = 0.013787151935851385
Step 6: x = [-3.78026216 -3.28413351], f(x) = 6.682470991387836e-05
Step 7: x = [-3.77924769 -3.28318489], f(x) = 2.256606135422328e-07
Step 8: x = [-3.77931162 -3.28318368], f(x) = 4.343462834235724e-10
Step 9: x = [-3.77931027 -3.28318606], f(x) = 1.847157808974678e-13
Βέλτιστο: [-3.77931027 -3.28318606]
Τιμή αντικειμενικής συνάρτησης στο βέλτιστο: 1.847157808974678e-13
Επαναλήψεις που χρειάστηκαν: 9
```

Figure 6: Επαναλήψεις για προσσέγγιση βελτίστου Αντικειμενικής Συνάρτησης Himmelblau (Μέθοδος BFGS)

Σημείωση: Κάτι που είναι εξίσου σημαντικό να αναφέρουμε, είναι πως αν χρησιμοποιήσουμε διαφορετικό αρχικό σημείο η μέθοδος μπορεί να συγκλίνει σε διαφορετικό βελτιστοποιητή. Για παράδειγμα, με αρχικό σημείο το $(1, 4)$, έναντι του $(4, 4)$, η μέθοδος προσέγγισε σε 10 βήματα τον ολικό ελαχιστοποιητή $x^* = (3, 2)$ της συνάρτησης. Αυτό επιβεβαιώνει την τοπική σημασία των Μεθόδων Newton γενικότερα, δηλαδή το ότι έχει σημασία για την σύγκλιση το ποια ήταν η αρχική μας προσέγγιση.

3.2.3 Αντικειμενική Συνάρτηση Rosenbrock

Θα χρησιμοποιήσουμε την συνάρτηση Rosenbrock:

$$f(x_0, x_1) = 100(x_1 - x_0^2)^2 + (x_0 - 1)^2$$

Η οποία έχει παραγώγους πρώτης τάξης:

$$f_{x_0} = -400x_0(x_1 - x_0^2) + 2(x_0 - 1)$$
$$f_{x_1} = 200(x_1 - x_0^2)$$

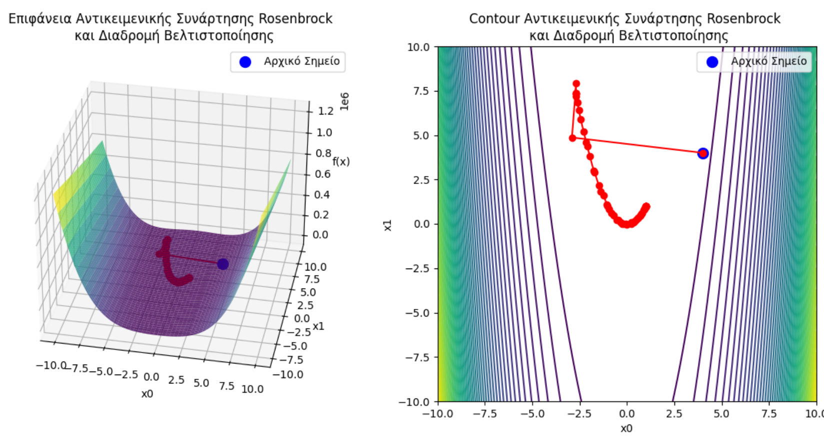


Figure 7: Μονοπάτι Βελτιστοποίησης και Contour της Αντικειμενικής Συνάρτησης Rosenbrock (Μέθοδος BFGS)

Βλέπουμε από τα Figures 7 και 8 ότι τα πράγματα εδώ είναι ακόμα χειρότερα καθώς η μέθοδος συγκλίνει μετά από 52 βήματα σε μια προσέγγιση του βελτιστοποιητή $x^* = (1, 1)$.

Σημείωση 1: Αν χρησιμοποιήσουμε την μέθοδο με σημεία πιο κοντά στον βελτιστοποιητή, η μέθοδος θα χρειαστεί λιγότερες επαναλήψεις όμως και πάλι θα είναι αρκετές. Για παράδειγμα με αρχικό σημείο το $(0, 0)$ χρειάστηκε 19 επαναλήψεις για να προσεγγίσει τον βελτιστοποιητή.

Σημείωση 2: Επίσης παρατηρούμε ότι στα πρώτα δύο βήματα η μέθοδος παρέκκλινε από την λύση πριν αρχίσει να την προσεγγίζει.

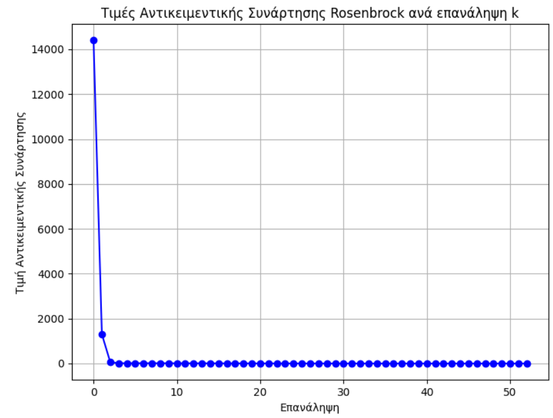


Figure 8: Τιμές Αντικειμενικής Συνάρτησης Rosenbrock ανά επανάλαψη k (Μέθοδος BFGS)

4 Μέθοδος SR1

Ο τύπος ανανέωσης της μεθόδου SR1 (Symmetric Rank-One) είναι:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$$

Είναι σημαντικό να σημειώσουμε πως το κύριο μειωνέκτημα αυτής της μεθόδου είναι πως δεν είναι σίγουρο πως ο πίνακας B_{k+1} που θα προσεγγίσει θα είναι πάντα θετικά ορισμένος.

4.1 Υλοποίηση SR1 σε Κώδικα

Η υλοποίηση σε κώδικα μένει σχεδόν ίδια με αυτή της μεθόδου BFGS. Το μόνο που αλλάζει είναι ο τρόπος ανανέωσης του πίνακα B_k όπως αναφέραμε και πριν.

4.2 SR1 σε Αντικειμενικές Συναρτήσεις

Θα δοκιμάσουμε την μέθοδο SR1 σε διάφορες αντικειμενικές συναρτήσεις ώστε να συγκρίνουμε την σύγκλιση της. Θα χρησιμοποιήσουμε τις ίδιες ακριβώς συναρτήσεις που χρησιμοποιήσαμε για την μέθοδο BFGS.

4.2.1 Αντικειμενική Συνάρτηση Quadratic Bowl

Εδώ παρατηρήθηκε για διάφορα αρχικά σημεία πως η μέθοδος δεν μπορεί να συγκλίνει γρήγορα ή και μερικές φορές και καθόλου και αντίθετα να αποκλίνει. Αυτό λογικά συνέβη διότι ο πίνακας B_{k+1} που προσέγγισε ίσως δεν ήταν θετικά ορισμένος με αποτέλεσμα να αποτύχει η μέθοδος.

4.2.2 Αντικειμενική Συνάρτηση Himmelblau

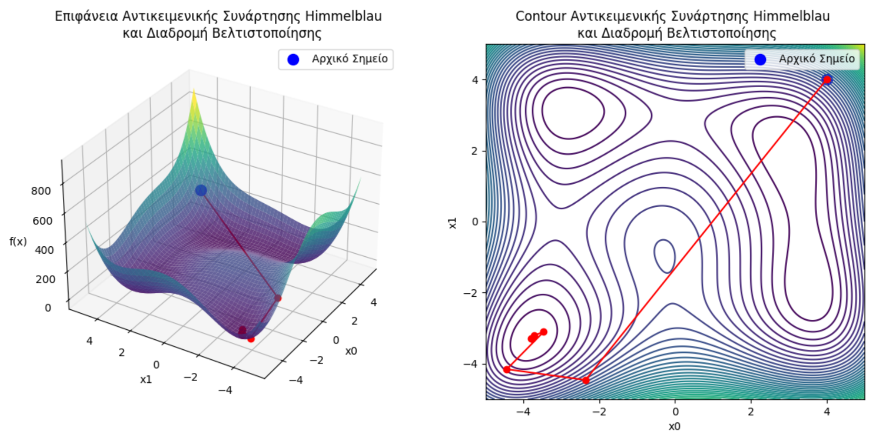


Figure 9: Μονοπάτι Βελτιστοποίησης και Contour της Αντικειμενικής Συνάρτησης Himmelblau (Μέθοδος SR1)

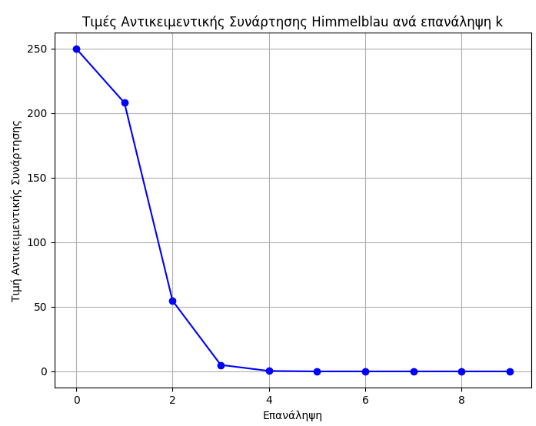


Figure 10: Τιμές Αντικειμενικής Συνάρτησης Himmelblau ανά επανάληψη k (Μέθοδος SR1)

Εδώ όπως βλέπουμε από τα Figures 9 και 10 έχουμε παρόμοια αποτελέσματα με την μέθοδο BFGS.

4.2.3 Αντικειμενική Συνάρτηση Rosenbrock

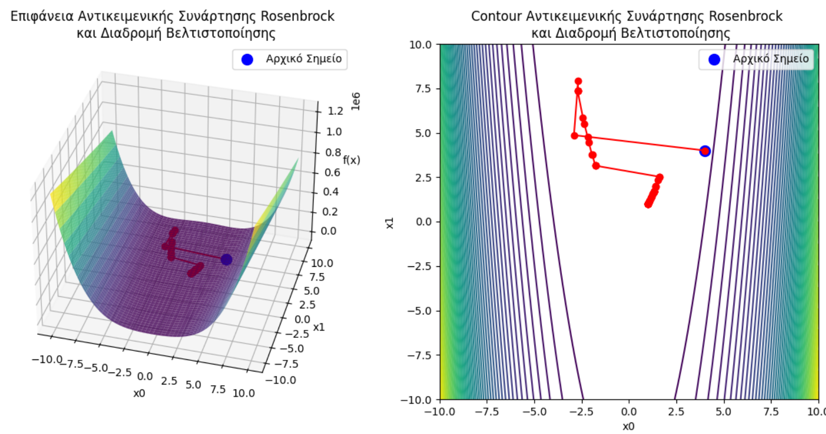


Figure 11: Μονοπάτι Βελτιστοποίησης και Contour της Αντικειμενικής Συνάρτησης Rosenbrock (Μέθοδος SR1)

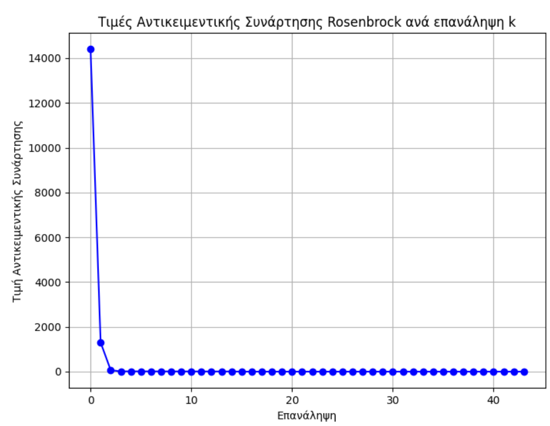


Figure 12: Τιμές Αντικειμενικής Συνάρτησης Rosenbrock ανά επανάλαψη k (Μέθοδος SR1)

Εδώ όπως βλέπουμε από τα Figures 11 και 12 έχουμε λίγο καλύτερα αποτελέσματα από την BFGS καθώς παρατηρούμε σύγκλιση σε 43 βήματα αντί για 52.