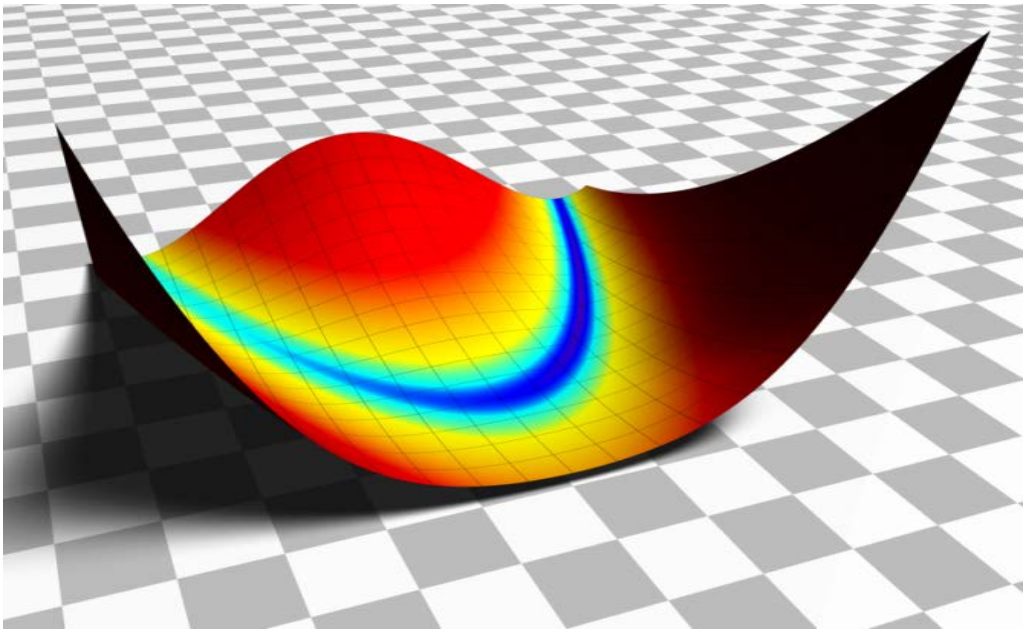
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΡΙΘΜΗΤΙΚΗ ΔΙΕΡΕΥΝΗΣΗ ΜΕΘΟΔΩΝ QUASI-NEWTON ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΕΛΑΧΙΣΤΟΥ ΧΩΡΙΣ ΠΕΡΙΟΡΙΣΜΟΥΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σωτήριος Ι.Λαμπρόπουλος



Επιβλέπων : Νικόλαος Μαράτος
Καθηγητής Ε.Μ.Π

Αθήνα, Νοέμβριος 2012

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΡΙΘΜΗΤΙΚΗ ΔΙΕΡΕΥΝΗΣΗ ΜΕΘΟΔΩΝ QUASI-NEWTON ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΕΛΑΧΙΣΤΟΥ ΧΩΡΙΣ ΠΕΡΙΟΡΙΣΜΟΥΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σωτήριος Ι.Λαμπρόπουλος

Επιβλέπων : Νικόλαος Μαράτος
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Νοεμβρίου 2012.

.....	.....	.....
Ν.Μαράτος	Τ.Κουσιουρής	Κ.Τζαφέστας
Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Αν.Καθηγητής ΕΜΠ

Αθήνα, Νοέμβριος 2012

.....
Σωτήριος Ι. Λαμπρόπουλος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας ήταν η αριθμητική διερεύνηση και σύγκριση τριών διαφορετικών quasi-newton μεθόδων για προβλήματα ελαχίστου χωρίς περιορισμούς. Συγκεκριμένα, γίνεται σύγκριση των δύο κλασσικών quasi-newton μεθόδων BFGS και DFP με μια νέα προτεινόμενη μέθοδο. Αρχικά, δίνεται έμφαση στην ανάλυση της σκέψης που μας οδηγεί στην εύρεση και χρήση quasi-newton μεθόδων, λόγω της δυσκολίας υπολογισμού της εσσιανής μήτρας της μεθόδου newton. Στην συνέχεια, παρατίθενται οι πλέον αποδοτικοί τρόποι για την εύρεση του μήκους βήματος καθώς και ο αλγόριθμος της νέας μεθόδου. Τονίζεται πως το χαρακτηριστικό της νέας μεθόδου που την διαφοροποιεί από τις κλασσικές είναι οτι εν αντιθέσει με τις κλασσικές μεθόδους που κάνουν χρήση μόνο πληροφοριών της προηγούμενης επανάληψης για την ανανέωση των μητρών τους, η νέα μέθοδος χρησιμοποιεί πληροφορία προηγούμενων επαναλήψεων, ανάλογων των μεταβλητών της υπό ελαχιστοποίηση συνάρτησης. Στα πλαίσια της εργασίας υλοποιήθηκαν στο περιβάλλον του MATLAB έξι διαφορετικά προγράμματα των μεθόδων, στα οποία χρησιμοποιήθηκαν συναρτήσεις, οι οποίες αντλήθηκαν από την βιβλιογραφία προκειμένου να αποφανθούμε για την αποτελεσματικότητα της νέας προτεινόμενης μεθόδου συγκριτικά με τις κλασσικές. Τα αποτελέσματα των πειραμάτων καταγράφηκαν σε πίνακες και μέσω αυτών πραγματοποιήθηκε η σύγκριση μεταξύ των μεθόδων. Εν τέλει, διαπιστώνεται από τα αριθμητικά αποτελέσματα ότι η νέα μέθοδος είναι αξιόπιστη, αποδοτική, εύρωστη και κρίνεται κατάλληλη για την επίλυση προβλημάτων ελαχίστου χωρίς περιορισμούς.

Λέξεις κλειδιά:

βελτιστοποίηση, προβλήματα χωρίς περιορισμούς, επαναληπτικοί αλγόριθμοι, αριθμητικά παραδείγματα, ψευδο-νευτώνιοι αλγόριθμοι, κυρτές συναρτήσεις, κατεύθυνση έρευνας, μήκος βήματος , ικανές συνθήκες ελαχίστου, αναγκαίες συνθήκες ελαχίστου, νέα μέθοδος.

Abstract

The scope of this thesis was the numerical investigation and comparison of three different quasi-newton methods for unconstrained minimization problems. In particular, there was made a comparison between the two classical quasi newton methods BFGS, DFP and the new proposed one. Initially, great emphasis was placed on the analysis of the thought that leads us in inventing and using quasi newton methods, due to the difficulty of computing the hessian matrix of newton method. After that, there is a reference in the most efficient algorithms for finding the appropriate step length and also the new method is presented. It is pointed out that the feature that differentiates our new method from the classical ones is that this method in contrast with others, which use data from the previous iteration only to update their formulas, uses data from a wide range of previous iterations equal to the number of variables of the function to be minimized. For the purpose of this thesis, six programs were implemented on matlab environment, in which we used functions derived from bibliography so that we can assess the efficiency of the new method compared to the other ones. The numerical results of the experiments were recorded in tables and thereby we compared the methods. To conclude, it is verified from the numerical experiments that the new method is reliable, efficient, robust and is deemed appropriate for the solution of unconstrained minimization problems.

Keywords:

optimization, unconstrained minimization, iterative methods, numerical examples, quasi-newton algorithms, convex functions, line search, step length, necessary minimization conditions, sufficient minimization conditions, new method.

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ στον Τομέα Σημάτων, Ελέγχου και Ρομποτικής .

Θα ήθελα να ευχαριστήσω, πρώτα απ'όλα, τον καθηγητή μου κύριο Νικόλαο Μαράτο, ο οποίος με εμπιστεύτηκε για την συγκεκριμένη διπλωματική εργασία και μου παρείχε τις πολύτιμες γνώσεις του και την βοήθειά του. Η συνεργασία μας σε όλη τη διάρκεια της εργασίας ήταν άψογη και η καθοδήγησή του καταλυτική για την ολοκλήρωση της εργασίας.

Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου που στέκονται πάντα δίπλα μου και με εμψυχώνουν για την επίτευξη των στόχων μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Γενικές πληροφορίες για τη βελτιστοποίηση

Στα Μαθηματικά, την Επιστήμη Υπολογιστών ή στη Διοικητική επιστήμη, η μαθηματική βελτιστοποίηση ή αλλιώς ο μαθηματικός προγραμματισμός αναφέρεται στην επιλογή της καλύτερης λύσης μέσα από ένα σύνολο πιθανών εναλλακτικών. Στην απλούστερη περίπτωση, ένα πρόβλημα βελτιστοποίησης συνίσταται στη μεγιστοποίηση ή ελαχιστοποίηση μιας πραγματικής συνάρτησης επιλέγοντας συστηματικά τιμές εισόδου από ένα σύνολο επιτρεπτών τιμών και υπολογίζοντας την τιμή της συνάρτησης κόστους.

Πολλά πρακτικά προβλήματα σχεδίασης ή λήψης απόφασης μπορούν να εκφρασθούν σαν προβλήματα βελτιστοποίησης. Ας υποθέσουμε ότι οι μεταβλητές $\left[x_1, x_2, x_3, \ldots x_n\right]$ και οι μεταξύ τους σχέσεις περιγράφουν ένα φαινόμενο ή διαδικασία ή σύστημα. Το πρόβλημα σχεδίασης η απόφασης συνίσταται στο να διαλέξουμε τιμές για τις μεταβλητές αυτές. Αν μπορέσουμε να εκφράσουμε ποσοτικά (μέσω μια βαθμωτής συνάρτησης f) την σχετική αξία της κάθε απόφασης, τότε η μεγιστοποίηση (ή ελαχιστοποίηση) του κριτηρίου ποιότητας f θα δώσει την καλύτερη-βέλτιστη λύση στο πρόβλημα σχεδίασης η απόφασης.Οι παράγοντες που ενδεχομένως περιορίζουν τις τιμές που μπορεί να πάρουν οι μεταβλητές f λαμβάνονται υπόψη σαν περιορισμοί κατά τη βελτιστοποίηση του κριτηρίου ποιότητας f .

Βεβαίως υπάρχουν πολλά πρακτικά προβλήματα στα οποία δεν είναι δυνατόν να βρεθεί κάποιο κριτήριο ποιότητας f που να εκφράζει ικανοποιητικά τη σχετική αξία των αποφάσεων ή και προβλήματα στα οποία μπορούν να προταθούν περισσότερα από ένα κριτήρια ποιότητας.

Υπάρχουν δύο βασικοί τρόποι επίλυσης του προβλήματος βελτιστοποίησης: ο αναλυτικός και ο αριθμητικός-επαναληπτικός. Όσον αφορά τον αναλυτικό τρόπο μπορούμε να χρησιμοποιήσουμε τις ικανές συνθήκες για το πρόβλημα βελτιστοποίησης. Όμως ο κατευθείαν υπολογισμός της λύσης $\tilde{x}$ με τον τρόπο αυτό είναι πρακτικά αδύνατος γιατί απαιτεί τη λύση συστήματος μη γραμμικών εξισώσεων ή συστημάτων μη-γραμμικών εξισώσεων και ανισοτήτων. Ο μόνος πρακτικός τρόπος επίλυσης μη-γραμμικών προβλημάτων βελτιστοποίησης στηρίζεται στη χρήση αλγορίθμων. Οι αλγόριθμοι αυτοί είναι αριθμητικές μέθοδοι επαναληπτικού χαρακτήρα σχεδιασμένοι για χρήση σε ηλεκτρονικό υπολογιστή, εκμεταλλευόμενοι προφανώς την ιδιότητα του για ταχείς επαναληπτικούς υπολογισμούς. Η λύση $\tilde{x}$ του προβλήματος βελτιστοποίησης προσεγγίζεται διαδοχικά από μια ακολουθία τιμών $\left[x_0, x_1, x_2, \ldots, x_k, x_{k+1}, \ldots\right]$ όπου x_0 είναι μια δεδομένη αρχική τιμή. Ο αλγόριθμος είναι μια απεικόνιση που παράγει το επόμενο στοιχείο x_{k+1} της ακολουθίας από το προηγούμενο x_k .

1.2 Δομή της εργασίας

Μετά τα εισαγωγικά στοιχεία περί βελτιστοποίησης, το δεύτερο κεφάλαιο περιγράφει τη γενική μορφή του προβλήματος βελτιστοποίησης και τις βασικές γενικές συνθήκες για την επίλυση του. Στο τρίτο κεφάλαιο παρατίθενται οι βασικές ιδιότητες των αλγορίθμων βελτιστοποίησης και η μέθοδος Newton, ενώ στο τέταρτο κεφάλαιο γίνεται γενική ανάλυση των μεθόδων quasi-newton και ειδικότερα των αλγορίθμων BFGS και DFP. Στην συνέχεια, στο πέμπτο κεφάλαιο αναφέρονται οι βασικές μέθοδοι για την εύρεση του μήκους βήματος και αναλύονται εκτενέστερα όσες χρησιμοποιήθηκαν στα προγράμματα. Το έκτο κεφάλαιο αφιερώνεται στην ανάλυση και παράθεση της νέας προτεινόμενης μεθόδου ενώ στο έβδομο κεφάλαιο γίνεται σύγκριση των μεθόδων μέσω αριθμητικών πειραμάτων και γραφικών παραστάσεων. Στο όγδοο κεφάλαιο αναφέρονται τα συμπεράσματα που εξάγονται από την εκτέλεση των προγραμμάτων στο Matlab, των οποίων ο κώδικας παρατίθεται με τις παραιτήσεις επεξηγήσεις στο παράρτημα.

ΚΕΦΑΛΑΙΟ 2

Πρόβλημα βελτιστοποίησης

2.1 Γενική μορφή προβλήματος

Το πρόβλημα βελτιστοποίησης έχει την εξής μορφή:

$$\text{minimize}\{ f(x) : x \in F\}$$

Όπου $f : \mathbb{R}^n \to \mathbb{R}$ είναι μια πραγματική συνάρτηση n μεταβλητών $x = \left[x_1, x_2, \ldots, x_n \right]^{\mathrm{T}}$ και $F \subset \mathbb{R}^n$ είναι το επιτρεπτό σύνολο:

1. για προβλήματα χωρίς περιορισμούς (unconstrained problems): $F = \mathbb{R}^n$

2. για προβλήματα με περιορισμούς $F = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \le 0\}$ $h : \mathbb{R}^n \to \mathbb{R}^m$ και $g : \mathbb{R}^n \to \mathbb{R}^l$ είναι πραγματικές συναρτήσεις.

Οι συναρτήσεις f, h και g συνήθως υποθέτουμε ότι είναι συνεχώς διαφορίσιμες. Πάντοτε υποθέτουμε ότι είναι συνεχείς.

Ορισμός: Ένα σημείο $\tilde{x} \in F$ λέγεται τοπικό ελάχιστο της f (local minimum) πάνω στο σύνολο F αν υπάρχει $\varepsilon > 0$ τέτοιο ώστε $f(\tilde{x}) \le f(x)$ για κάθε x που ικανοποιεί: $x \in F$ και $\| x - \tilde{x} \| \le \varepsilon$. Αν η αυστηρή ανισότητα ικανοποιείται για κάθε $x \in F$, $x \ne \tilde{x}, \| x - \tilde{x} \| \le \varepsilon$,τότε το $\tilde{x}$ λέγεται αυστηρό τοπικό ελάχιστο.

Ορισμός: Ένα σημείο $\tilde{x} \in F$ λέγεται γενικό ελάχιστο(global minimum) της F πάνω στο σύνολο F αν $f(\tilde{x}) \le f(x)$ για κάθε $x \in F$.Αν $f(\tilde{x}) < f(x)$ για κάθε $x \in F, x \ne \tilde{x},$ τότε το $\tilde{x}$ λέγεται αυστηρό γενικό ελάχιστο.

Θεώρημα Weierstrass: (ικανές συνθήκες ύπαρξης γενικού ελαχίστου).

Αν η $f : \mathbb{R}^n \to \mathbb{R}$ είναι συνεχής και το $F \subset \mathbb{R}^n$ είναι συμπαγές (κλειστό και φραγμένο) τότε η f έχει ένα γενικό ελάχιστο πάνω στο F .

Το θεώρημα Weierstrass είναι ένα θεώρημα υπάρξεως και δεν προσφέρει έναν αναλυτικό χαρακτηρισμό του γενικού ελαχίστου. Έτσι ο πρακτικός υπολογισμός του είναι πολύ δύσκολος αν δεν κάνουμε περιοριστικές υποθέσεις για την f και το F .

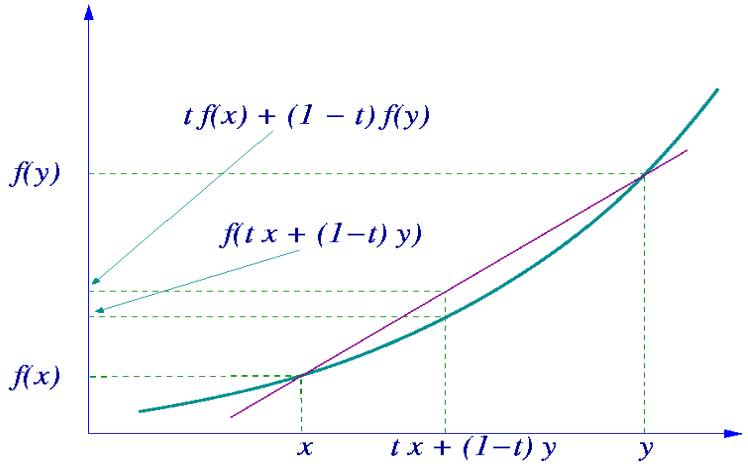
Αντίθετα ένα τοπικό ελάχιστο του προβλήματος βελτιστοποίησης μπορεί να χαρακτηριστεί αναλυτικά (συνθήκες βελτίστου) και πρακτικά να προσδιοριστεί με επαναληπτικές μεθόδους, οι οποίες βασικά συγκρίνουν κοντινές τιμές της f και επομένως βρίσκουν ένα τοπικό και όχι κατ'ανάγκην το γενικό ελάχιστο.

2.2 Κυρτά προβλήματα

Γι'αυτή την κατηγορία των προβλημάτων βελτιστοποίησης κάθε τοπικό ελάχιστο είναι και γενικό.

Μια συνάρτηση $f : F \subset \mathbb{R}^n \to \mathbb{R}$ που ορίζεται πάνω σ'ένα κυρτό σύνολο F, λέγεται κυρτή αν για κάθε $x, y \in F$ και για κάθε $a \in [0,1]$ ισχύει:

$$f\left(ax+(1-a)y\right) \le af\left(x\right)+(1-a)\,f\left(y\right)$$



Η f λέγεται αυστηρά κυρτή αν η προηγούμενη ανισότητα είναι αυστηρή για κάθε $x \neq y$ και $a \in (0,1)$.

Μια συνάρτηση f που ορίζεται πάνω σ'ένα κυρτό σύνολο F λέγεται κοίλη αν η $-f$ είναι κυρτή.

ΙΔΙΟΤΗΤΕΣ ΚΥΡΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ

i) αν σε f_1 και f_2 είναι κυρτές πάνω στο, τότε και η $f_1 + f_2$ είναι κυρτή πάνω στο F.

ii) αν η f είναι κυρτή, τότε και η af είναι κυρτή για κάθε $a \ge 0$.

iii) αν η f είναι κυρτή πάνω στο F, τότε το σύνολο $\left\{x \in F : f\left(x\right) \le c\right\}$ είναι κυρτό για κάθε πραγματικό c.

Θεώρημα: Αν f είναι μια κυρτή συνάρτηση που ορίζεται πάνω σ'ένα κυρτό σύνολο $F \subset \mathbb{R}^n$, τότε i) το σύνολο των σημείων που είναι γενικά ελάχιστα της f πάνω στο F είναι κυρτό ii) κάθε τοπικό ελάχιστο της f είναι και γενικό ελάχιστο.

2.3 Αναγκαίες συνθήκες για τοπικό ελάχιστο χωρίς περιορισμούς

Θεώρημα: (αναγκαίες συνθήκες 1ης τάξης)

Αν η $f:\mathbb{R}^n \to \mathbb{R}$ είναι συνεχώς διαφορίσιμη και $\tilde{x} \in \mathbb{R}^n$ είναι ένα τοπικό ελάχιστο της f, τότε $\nabla f(\tilde{x}) = 0$.

Τα σημεία στα οποία η κλίση της συνάρτησης f μηδενίζεται, λέγονται στάσιμα σημεία της f. Το σύνολο $D = \{\tilde{x} \in \mathbb{R}^n : \nabla f(x) = 0\}$ των στάσιμων σημείων της f περιέχει όλα τα τοπικά ελάχιστα και μέγιστα της f αλλά επίσης αν υπάρχουν και σημεία που δεν είναι ούτε τοπικά ελάχιστα ούτε τοπικά μέγιστα.

Θεώρημα: (αναγκαίες συνθήκες 2ης τάξης)

Αν η $f:\mathbb{R}^n \to \mathbb{R}$ είναι δύο φορές συνεχώς διαφορίσιμη και $\tilde{x}$ είναι ένα τοπικό ελάχιστο της f στο $\mathbb{R}^n$, τότε i) $\nabla f(\tilde{x}) = 0$

και ii) $y^T \frac{\partial^2 f(\tilde{x})}{\partial x^2} y \geq 0$ για κάθε $y \in \mathbb{R}^n$.(δηλαδή η μήτρα 2ων παραγώγων της f είναι θετικά ορισμένη).

Το σύνολο των σημείων που ικανοποιούν τις αναγκαίες συνθήκες ελαχίστου 2ης τάξης είναι ένα υποσύνολο των στάσιμων σημείων της f που περιέχει όλα τα τοπικά ελάχιστα της f στο $\mathbb{R}^n$ αλλά επίσης αν υπάρχουν και σημεία που δεν είναι ούτε ελάχιστα ούτε μέγιστα. Δεν περιέχει κανένα τοπικό μέγιστο της f.

2.4 Ικανές συνθήκες για τοπικό ελάχιστο χωρίς περιορισμούς

Θεώρημα: (ικανές συνθήκες 2ης τάξης)

Αν η $f:\mathbb{R}^n \to \mathbb{R}$ είναι συνεχώς διαφορίσιμη και για κάποιο $\tilde{x} \in \mathbb{R}^n$ έχουμε:

i) $\nabla f(\tilde{x}) = 0$

ii) $\frac{\partial^2 f(\tilde{x})}{\partial x^2}$ είναι θετικά ορισμένη τότε το $\tilde{x}$ είναι αυστηρό τοπικό ελάχιστο της f.

Συνοψίζοντας έχουμε ότι:

i) Αν $\nabla f(\tilde{x}) = 0$ και $\frac{\partial^2 f(\tilde{x})}{\partial x^2} > 0$ τότε το $\tilde{x}$ είναι αυστηρό τοπικό ελάχιστο.

ii) Αν $\nabla f(\tilde{x}) = 0$ και $\frac{\partial^2 f(\tilde{x})}{\partial x^2} < 0$ τότε το $\tilde{x}$ είναι αυστηρό τοπικό μέγιστο.

iii) Αν $\nabla f(\tilde{x}) = 0$ και $\frac{\partial^2 f(\tilde{x})}{\partial x^2} \geq 0$ τότε το $\tilde{x}$ είναι τοπικό ελάχιστο ή σημείο καμπής ή σαγματικό σημείο.

iv) Αν $\nabla f(\tilde{x}) = 0$ και $\frac{\partial^2 f(\tilde{x})}{\partial x^2} \leq 0$ τότε το $\tilde{x}$ είναι τοπικό μέγιστο ή σημείο καμπής ή σαγματικό σημείο.

v) Αν $\nabla f(\tilde{x}) = 0$ και $\frac{\partial^2 f(\tilde{x})}{\partial x^2}$ είναι αόριστη, τότε το $\tilde{x}$ είναι σαγματικό σημείο.

Πόρισμα: Αν η f είναι κυρτή και διαφορίσιμη,τότε κάθε σημείο $\tilde{x}$ που ικανοποιεί $\nabla f(\tilde{x}) = 0$ είναι γενικό ελάχιστο(global minimum) της f .

Στις περιπτώσεις (iii), (iv) δεν μπορούμε να προσδιορίσουμε το είδος του στάσιμου σημείου $\tilde{x}$ παρά μόνο αν χρησιμοποιήσουμε παραπάνω πληροφορία για τη συνάρτηση f , για παράδειγμα παραγώγους τάξης ανώτερης της $2^{ας}$.

Το σημείο $(\tilde{x}, \tilde{y}) \in \mathbb{R}^{n+m}$ λέγεται σαγματικό της συνάρτησης $f(x, y)$, $f : \mathbb{R}^{n+m} \to \mathbb{R}$,αν $f(\tilde{x}, y) \leq f(\tilde{x}, \tilde{y}) \leq f(x, \tilde{y})$ $\forall x : \| x - \tilde{x} \| \leq \varepsilon$ και $\forall y : \| y - \tilde{y} \| \leq \varepsilon$. Δηλαδή για σταθερό $y = \tilde{y}$ η f (θεωρούμενη σαν συνάρτηση μόνο του x) έχει τοπικό ελάχιστο στο $\tilde{x}$ και για σταθερό $x = \tilde{x}$ η f (θεωρούμενη σαν συνάρτηση μόνο του y) έχει τοπικό μέγιστο στο $\tilde{y}$.

ΚΕΦΑΛΑΙΟ 3

Ανάλυση αλγορίθμων

3.1 Βασικές ιδιότητες αλγορίθμων

Υπάρχει μεγάλη ποικιλία αλγορίθμων βελτιστοποίησης , όλοι όμως έχουν ορισμένα βασικά κοινά χαρακτηριστικά: είναι i) επαναληπτικές μέθοδοι και ii) μέθοδοι καθόδου.

i) Οι αλγόριθμοι είναι επαναληπτικές μέθοδοι σημαίνει οτι αρχίζουν από ένα δεδομένο αρχικό σημείο x_0 και κατασκευάζουν μια ακολουθία σημείων $\{x_i\}_{i=0}^{\infty}$ σύμφωνα με τον εξής κανόνα: κάθε νέο σημείο x_{i+1} της ακολουθίας προκύπτει από το προηγούμενο μέσω μιας απεικόνισης $A:\mathbb{R}^n \to \mathbb{R}^n:\ x_{i+1} = A(x_i)$.

Η απεικόνιση A χαρακτηρίζει τον αλγόριθμο και διαφέρει από αλγόριθμο σε αλγόριθμο.

ii) Οι αλγόριθμοι είναι μέθοδοι καθόδου σημαίνει ότι κάθε νέο σημείο x_{i+1} της ακολουθίας που κατασκευάζει ένας αλγόριθμος μειώνει της τιμή $z(x_{i+1})$ μια συνεχούς συνάρτησης z , η οποία χαρακτηρίζει το προς επίλυση πρόβλημα βελτιστοποίησης για παράδειγμα $z(x) = f(x)$ για το πρόβλημα χωρίς περιορισμούς $\min_x f(x)$. Δηλαδή ισχύει $z(x_{i+1}) < z(x_i)$. Η ιδιότητας της καθόδου παίζει σημαντικό ρόλο(είναι αναγκαία συνθήκη αλλά όχι και ικανή) για τη σύγκλιση ενός αλγορίθμου.

Κάθε αλγόριθμος βελτιστοποίησης εξετάζεται αναφορικά με δυο βασικές ιδιότητες i)γενική σύγκλιση ii)ταχύτητα σύγκλισης. Με ορισμένες προϋποθέσεις $\lim_{x\to\infty} x_k = \tilde{x}$.Αυτή η ιδιότητα εξασφαλίζει ότι το σημείο που προσεγγίζεται από την ακολουθία $[x_0,\ldots,x_k]$ είναι πράγματι το $\tilde{x}$.Η ταχύτητα σύγκλισης είναι ένα μέτρο του πόσο γρήγορα η ακολουθία $\{x_i\}_{i=0}^{\infty}$ που δημιουργείται από τον αλγόριθμο συγκλίνει στο $\tilde{x}$,προσεγγίζει δηλαδή την επιθυμητή τιμή.Σχεδόν πάντοτε ένας αλγόριθμος A απαρτίζεται από δύο στοιχεία: την κατεύθυνση έρευνας $h(x_i)$ και το μήκος βήματος λ_i με τέτοιο τρόπο ώστε: $x_{i+1} = x_i + \lambda_i h(x_i)$.

Γενικότερα πολλοί αλγόριθμοι έχουν προταθεί για την λύση του μη-περιορισμένου προβλήματος ,οι οποίοι χωρίζονται σε κατηγορίες ανάλογα με τις απαιτήσεις που έχουν ως προς την υπολογιστική διαθεσιμότητα της f και των παραγώγων της f . Έτσι όλοι οι αλγόριθμοι για την λύση του προβλήματος βελτιστοποίησης προϋποθέτουν οτι, αν δοθεί ένα $x \in \mathbb{R}^n$, είναι διαθέσιμο ένα υποπρόγραμμα που παίρνει το δεδομένο x σαν είσοδο και επιστρέφει την τιμή $f(x)$. Οι αλγόριθμοι που έχουν μόνο την παραπάνω απαίτηση λέγονται μέθοδοι άμεσης έρευνας(direct search methods).Πολλοί αλγόριθμοι,πέρα από την απαίτηση αυτή , προϋποθέτουν ότι είναι διαθέσιμο ένα

δεύτερο υποπρόγραμμα που παίρνει σαν είσοδο ένα δεδομένο $x \in \mathbb{R}^n$ και επιστρέφει την κλίση $\nabla f(x)$ της συνάρτησης f στο σημείο x. Οι αλγόριθμοι αυτοί λέγονται μέθοδοι 1ης τάξης και συγκλίνουν ταχύτερα από τις μεθόδους άμεσης έρευνας και χρησιμοποιούνται ευρέως στην πράξη για τη λύση του προβλήματος.

Τέλος ο αλγόριθμος $Newton$ απαιτεί επιπλέον την ύπαρξη ενός υποπρογράμματος, το οποίο για δεδομένο $x \in \mathbb{R}^n$,επιστρέφει τη μήτρα $2^{ων}$ παραγώγων $\frac{\partial^2 f(x)}{\partial x^2}$ της συνάρτησης f στο σημείο x.Πρακτικά η απαίτηση αυτή είναι πολλές φορές υπερβολική, είτε γιατί οι δεύτερες παράγωγοι της συνάρτησης f δεν είναι διαθέσιμες, είτε διότι ο υπολογισμός τους είναι κοπιώδης . Όταν όμως μπορεί να χρησιμοποιηθεί , ο αλγόριθμος $Newton$ συγκλίνει ταχύτερα από όλες τις άλλες μεθόδους. Ανεξάρτητα από τις παραπάνω ιδιότητες που είναι χαρακτηριστικές του κάθε αλγορίθμου, κάθε "καλός" αλγόριθμος θα πρέπει να διαθέτει τα εξής:

1) να είναι εύρωστος(robust): θα πρέπει να αποδίδει καλά σε ένα μεγάλο εύρος προβλημάτων , για κάθε λογική επιλογή των αρχικών τιμών των μεταβλητών.

2)να είναι αποδοτικός(efficient): δεν θα πρέπει να έχει μεγάλη χωρική και χρονική πολυπλοκότητα.

3)να είναι ακριβής (accuracy): θα πρέπει δηλαδή να βρίσκει μια λύση με ακρίβεια,χωρίς να είναι πολύ ευαίσθητος σε λάθη στα δεδομένα ή σε αριθμητικά λάθη στρογγυλοποίησης που συμβαίνουν όταν προγραμματίζεται ο αλγόριθμος σε κάποιο ειδικό περιβάλλον(MATLAB).

Όλοι οι παραπάνω αναφερόμενοι αλγόριθμοι έχουν την γενική μορφή που δόθηκε προηγουμένως. Διαφέρουν μεταξύ τους κυρίως ως προς τον τρόπο που ο καθένας ορίζει την κατεύθυνση έρευνας h_i.

Πιο συγκεκριμένα , όσον αφορά της μέθοδο της κλίσης, αν x_i είναι το τρέχον σημείο στην επανάληψη i ,ο αλγόριθμος ορίζει ως κατεύθυνση έρευνας h_i την κατεύθυνση $h_i = -\nabla f(x_i)$. Η μέθοδος της κλίσης συνιστά έναν απλό αλγόριθμο, ο οποίος συγκλίνει πάντοτε , ωστόσο η ταχύτητα σύγκλισής του είναι μόνο γραμμική.

3.2 Μέθοδος Newton

Η μέθοδος $NEWTON$ υποθέτει ότι η συνάρτηση f είναι 2 φορές παραγωγίσιμη και απαιτεί η μήτρα $2^{ων}$ παραγώγων $\frac{\partial^2 f(x)}{\partial x^2}$ να είναι υπολογιστικά διαθέσιμη, εντούτοις προσφέρει μεγάλη ταχύτητα σύγκλισης. Η καθαρή μορφή της μεθόδου αυτής είναι ουσιαστικά μια εφαρμογή της μεθόδου $NEWTON - RAPHSON$ (για την λύση μη-γραμμικών εξισώσεων) στις εξισώσεις $\nabla f(x) = 0$ που ορίζουν τα επιθυμητά σημεία του προβλήματος ελαχιστοποίησης.

Αν x_k είναι ένα δεδομένο σημείο,τότε κάθε άλλο σημείο $x \in \mathbb{R}^n$ εκφράζεται σαν $x = x_k + h$ όπου $h \in \mathbb{R}^n$. Τότε η εξίσωση $\nabla f(x) = 0$ γράφεται ως $\nabla f(x_k + h) = 0$ (1). Αναπτύσσοντας σε σειρά $Taylor$ γύρω από το x_k έχουμε $\nabla f(x_k + h) = \nabla f(x_k) + \frac{\partial^2 f(x_k)}{\partial x^2} h + O(\| h \|^2)$. Αν από την σχέση αυτή παραλείψουμε τον όρο $O(\| h \|^2)$ παίρνουμε την εξής προσεγγιστική μορφή της (1): $\nabla f(x_k) + \frac{\partial^2 f(x_k)}{\partial x^2} h = 0$. Λύνοντας το σύστημα των γραμμικών εξισώσεων ως προς h παίρνουμε την κατεύθυνση έρευνας $NEWTON$: $h_k = -\left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k)$. Η μέθοδος $NEWTON$ στην καθαρή της μορφή χρησιμοποιεί ακριβώς αυτή την μετατόπιση h_k για να ορίσει το νέο σημείο x_{k+1} βάσεις της επαναληπτικής σχέσης

$$x_{k+1} = x_k + h_k = x_k - \left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k) \text{ (2).}$$

Το βασικότερο πλεονέκτημα της μεθόδου είναι η μεγάλη ταχύτητα σύγκλισης. Ισχύει μάλιστα πως αν η ακολουθία που κατασκευάζεται από την επαναληπτική σχέση (2) συγκλίνει σ΄ένα σημείο $\tilde{x}$ που ικανοποιεί $\frac{\partial^2 f(\tilde{x})}{\partial x^2} > 0$ και $\nabla f(\tilde{x}) = 0$ τότε η τάξη σύγκλισής της είναι τετραγωνική.

<u>Γενική σύγκλιση:</u> Οι ιδιότητες γενικής σύγκλισης της μεθόδου στην καθαρή της μορφή δεν είναι καλές.Η ακολουθία που κατασκευάζεται από την επαναληπτική μέθοδο συγκλίνει σε μια λύση $\tilde{x}$ του προβλήματος μόνο εάν το αρχικό σημείο x_0 βρίσκεται αρκετά κοντά στο $\tilde{x}$. Για αρχικά σημεία μακριά από το ελάχιστο δεν υπάρχει καμιά εγγύηση της σύγκλισης της μεθόδου . Γι' αυτόν τον λόγο τροποποιούμε την μέθοδο $NEWTON$ με τους δυο ακόλουθους τρόπους προκειμένου να εξασφαλίζεται γενική σύγκλιση.

1).Εισαγωγή μήκους βήματος στην κατεύθυνση h_k. Είναι δυνατόν για μη- τετραγωνικές συναρτήσεις f η τιμή $f(x_{k+1})$ της f στο $x_{k+1} = x_k - \left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k)$ να είναι μεγαλύτερη από $f(x_k)$ λόγω μη-τετραγωνικών όρων της f. Αντιμετωπίσουμε αυτή την κατάσταση με την εισαγωγή μήκους βήματος λ_k στην κατεύθυνση h_k οπότε και η προηγούμενη σχέση διαμορφώνεται ως εξής: $x_{k+1} = x_k - \lambda_k \left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k)$. Αποδεικνύεται βέβαια ότι η επανάληψη αυτή δεν έχει τετραγωνική ταχύτητα σύγκλισης παρά μόνο αν $\lambda_k = 1$ για κάθε $k \geq K$. Επομένως επιθυμούμε το μήκος βήματος να

διατηρείται ίσο με $\lambda_k = 1$ όποτε αυτό είναι δυνατό, κάτι το οποίο μπορούμε να το εξασφαλίσουμε με χρήση του κανόνα $Armijo$ για την έρευνα γραμμής κατά μήκος της κατεύθυνσης h_k.

2).Ιδιότητα καθόδου. Η ιδιότητα της καθόδου εξασφαλίζεται αν για μικρά $\lambda \geq 0$ ισχύει : $f(x_k + \lambda h_k) - f(x_k) < 0$. Όμως από το ανάπτυγμα $Taylor$ έχουμε :

$f(x_k + \lambda h_k) = f(x_k) + \lambda \nabla f(x_k)^{\mathrm{T}} h_k + O(\| \lambda h_k \|^2)$. Κατά συνέπεια η ιδιότητα της καθόδου εξασφαλίζεται μόνο αν $\nabla f(x_k)^{\mathrm{T}} h_k < 0$.Όπως έχουμε ήδη πει $h_k = -\left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k)$ για τη μέθοδο $Newton$. Επομένως $\nabla f(x_k)^{\mathrm{T}} h_k = -\nabla f(x_k)^{\mathrm{T}} \left[\frac{\partial^2 f(x_k)}{\partial x^2} \right]^{-1} \nabla f(x_k)$, δηλαδή η ιδιότητα της καθόδου εξασφαλίζεται μόνο αν η μήτρα δευτέρων παραγώγων $\frac{\partial^2 f(x_k)}{\partial x^2}$ είναι θετικά ορισμένη. Όμως σε σημεία που βρίσκονται μακριά από ένα τοπικό ελάχιστο της f είναι δυνατόν η μήτρα αυτή να μην είναι θετικά ορισμένη.

Προκειμένου να αντιμετωπίσουμε αυτήν την κατάσταση η μέθοδος $Newton$ τροποποιείται ως εξής: χρησιμοποιούμε την κλασσική κατεύθυνση h_k όπως ορίζεται μόνο όταν $\frac{\partial^2 f(x_k)}{\partial x^2} > 0$.Αν $\frac{\partial^2 f(x_k)}{\partial x^2}$ δεν είναι θετικά ορισμένη χρησιμοποιούμε μια τροποποιημένη κατεύθυνση έρευνας.

ΚΕΦΑΛΑΙΟ 4

Μέθοδοι Quasi-Newton

4.1 Γενική ανάλυση μεθόδων Quasi-Newton

Οι Quasi-Newton μέθοδοι, βρίσκονται κατά κάποιον τρόπο, μεταξύ της μεθόδου της κλίσης και της μεθόδου $Newton$. Οι μέθοδοι αυτές επιτυγχάνουν υπεργραμμική ταχύτητα σύγκλισης , χωρίς να απαιτούν υπολογισμό δευτέρων παραγώγων της f. Ορίζουν την κατεύθυνση έρευνας με τρόπο παρόμοιο με την μέθοδο $Newton$, αντί όμως για την αντίστροφη μήτρα δευτέρων παραγώγων , χρησιμοποιούν μια μήτρα S_k, η οποία αποτελεί προσέγγιση της $\left[\frac{\partial^2 f(x_k)}{\partial x^2}\right]^{-1}$. Η μήτρα S_k ενημερώνεται σε κάθε επανάληψη χρησιμοποιώντας την πληροφορία που αποκτήθηκε από τον υπολογισμό της κλίσης $\nabla f(x_k)$ της συνάρτησης.

Η βασική επαναληπτική σχέση που χρησιμοποιούν οι QUASI-NEWTON μέθοδοι είναι η εξής: $x_{k+1} = x_k + \lambda_k h_k = x_k - \lambda_k S_k \nabla f(x_k)$ και $h_k = -S_k \nabla f(x_k)$. Η ίδια επαναληπτική σχέση χρησιμοποιείται και από τις μεθόδους κλίσης και $NEWTON$.

Η μήτρα S_k υπολογίζεται αναδρομικά ως εξής: $S_{k+1} = S_k + B_k$, όπου η μήτρα B_k επιλέγεται έτσι ώστε να ισχύουν τα παρακάτω

i) Η μήτρα S_k να είναι συμμετρική για κάθε k (εφόσον προσεγγίζει την συμμετρική $\left[\frac{\partial^2 f(x_k)}{\partial x^2}\right]^{-1}$)

ii) να είναι θετικά ορισμένη (για να εξασφαλίζει την ιδιότητα της καθόδου).

Η ιδιότητα της καθόδου εξασφαλίζεται αν $\nabla f(x_k)^{\mathrm{T}} h_k < 0$

Για τις $QUASI - NEWTON$ μεθόδους έχουμε ότι: $\nabla f(x_k)^{\mathrm{T}} h_k = -\nabla f(x_k)^{\mathrm{T}} S_k \nabla f(x_k)$

Επομένως αν $S_k > 0$ η ιδιότητα της καθόδου είναι εξασφαλισμένη.

iii) να ικανοποιείται η εξίσωση $quasi\ \ newton$ ($quasi\ newton\ \ equation$ - $secant\ equation$) $S_{k+1} q_k = p_k$ για κάθε k όπου $p_k = x_{k+1} - x_k$ $q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

Οι δύο πιο γνωστές μέθοδοι $quasi\ \ newton$ είναι η μέθοδος DFP και η μέθοδος BFGS.

ΜΕΘΟΔΟΣ Davidon-Fletcher-Powell(DFP)

Η μέθοδος αυτή στην επανάληψη k, ενημερώνει την μήτρα S_k με μια διόρθωση βαθμού 2, της μορφής(άθροισμα 2 ενημερώσεων βαθμού 1).

$$S_{k+1} = S_k + a_k y_k y_k^{\mathrm{T}} - \beta_k z_k z_k^{\mathrm{T}}$$

Με τον τρόπο αυτό εξασφαλίζεται η συμμετρία και το θετικά ορισμένο της μήτρας S_k. Επιπρόσθετα ,για τετραγωνικές συναρτήσεις , η μέθοδος DFP έχει την εξής πολύ ενδιαφέρουσα ιδιότητα: οι κατευθύνσεις που κατασκευάζει είναι συζυγείς.

Η μέθοδος DFP συγκεντρώνει πολλά πλεονεκτήματα , χρησιμοποιεί μόνο $1^{ης}$ τάξεως παραγώγους της συνάρτησης f , εξασφαλίζει πάντοτε την ιδιότητα της καθόδου καθότι ισχύει $S_k > 0$ σε κάθε επανάληψη k, έχει υπεργραμμική ταχύτητα σύγκλισης και επιπλέον η σύγκλιση της μεθόδου δεν είναι ευαίσθητη στην ακρίβεια υπολογισμού του μήκους βήματος λ_k ,δηλαδή δεν απαιτούνται μεγάλης ακρίβειας έρευνες γραμμής.

Όμως η ταχύτητα σύγκλισης της μεθόδου πρακτικά μειώνεται όταν η έρευνα γραμμής δεν είναι ακριβής. Βέβαια με κατάλληλη τροποποίηση της σχέσης ενημέρωσης της μήτρας S_{k+1}, αίρεται το πρόβλημα της ευαισθησίας της έρευνας γραμμής , δηλαδή δεν επηρεάζεται η μέθοδος πολύ από ανακρίβειες στην έρευνα γραμμής , ενώ συγχρόνως διατηρούνται τα πλεονεκτήματα της μεθόδου DFP.

4.2 Αλγόριθμος DFP

Βήμα 0: $x_0 \in \mathbb{R}^n$, S_0 συμμετρική,θετικά ορισμένη (για παράδειγμα $S_0 = I$) , $k = 0$.

Βήμα 1: $h_k = -S_k \nabla f(x_k)$

Βήμα 2: $\lambda_k = \arg\min_{\lambda \geq 0} \{ f(x_k + \lambda h_k) \}$

Βήμα 3: $x_{k+1} = x_k + \lambda_k h_k$

$$p_k = x_{k+1} - x_k = \lambda_k h_k$$

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

Βήμα 4: (Προαιρετικό:Αν το k είναι πολλαπλάσιο του n ,τότε θέσε $S_{k+1} = S_0$.Στο Βήμα 4 η μήτρα S_k ξαναπαίρνει την αρχική της τιμή S_0 κάθε n επαναλήψεις.

Αν το k είναι δεν πολλαπλάσιο του n, τότε θέσε $S_{k+1} = S_k + \frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} - \frac{S_k q_k q_k^{\mathrm{T}} S_k}{q_k^{\mathrm{T}} S_k q_k}$) Αυτό ενδεχομένως μπορεί να βελτιώσει τα χαρακτηριστικά γενικής σύγκλισης της DFP .

$$S_{k+1} = S_k + \frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} - \frac{S_k q_k q_k^{\mathrm{T}} S_k}{q_k^{\mathrm{T}} S_k q_k}$$

Βήμα 5: Αν $|| \nabla f(x_{k+1}) || = 0$ σταμάτησε.

Αλλιώς θέσε $k = k+1$ και πήγαινε στο Βήμα 1.

Για τη μέθοδο DFP ισχύει:

$$S_{k+1} q_k = S_k q_k + p_k - \frac{S_k q_k q_k^{\mathrm{T}} S_k q_k}{q_k^{\mathrm{T}} S_k q_k} = p_k$$

επομένως ικανοποιείται η εξίσωση quasi newton.

Οι παρακάτω προτάσεις δείχνουν δύο βασικές ιδιότητες της μεθόδου DFP:

<u>Πρόταση 1</u>: Αν σε μια επανάληψη k η μήτρα S_k είναι θετικά ορισμένη, τότε και η S_{k+1} είναι θετικά ορισμένη.

<u>Πρόταση 2:</u> Αν η f είναι τετραγωνική: $f(x) = a + b^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} C x$ με θετικά ορισμένη την μήτρα C, τότε για την μέθοδο D.F.P έχουμε:

$$p_i^{\mathrm{T}} C p_j = 0 \text{ για } 0 \le i < j \le k$$

$$S_{k+1} C p_i = p_i \text{ για } 0 \le i \le k$$

4.3 Αλγόριθμος BFGS

Υποθέτουμε οτι αντί για τη μήτρα S ,η οποία προσεγγίζει την αντίστροφη της μήτρας δευτέρων παραγώγων της συνάρτησης, υπολογίζουμε τη μήτρα M η οποία συνιστά προσέγγιση της μήτρας δευτέρων παραγώγων[6,][11],[12]. Τότε η μήτρα M πρέπει να ικανοποιεί την δυαδική εξίσωση *quasi newton* δηλαδή $M_{k+1}p_k = q_k$. Αν στην παραπάνω σχέση του βήματος 4 που δείχνει τον τρόπο με τον οποίο ανανεώνεται σε κάθε επανάληψη η μήτρα S αντικαταστήσουμε τη $M = S^{-1}$, παρατηρούμε οτι οι μέθοδοι BFGS και DFP είναι δυαδικές, κάτι το οποίο συμβαίνει πάντοτε όταν ανακαλύπτεται μια μέθοδος *quasi newton* .

Συγκεκριμένα αντιπαραβάλλοντας τον τρόπο με τον οποίο ανανεώνονται οι μήτρες S και M για τις δυο μεθόδους[6],[12] διακρίνεται εύκολα η δυαδικότητα τους:

DFP	$S_{k+1} = S_k + \frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} - \frac{S_k q_k q_k^{\mathrm{T}} S_k}{q_k^{\mathrm{T}} S_k q_k}$	$M_{k+1} = M_k + \frac{q_k q_k^{\mathrm{T}}}{q_k^{\mathrm{T}} p_k}\left(1 + \frac{p_k^{\mathrm{T}} M_k p_k}{q_k^{\mathrm{T}} p_k}\right) - \frac{M_k p_k q_k^{\mathrm{T}} + q_k p_k^{\mathrm{T}} M_k}{q_k^{\mathrm{T}} p_k}$
BFGS	$S_{k+1} = S_k + \frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k}\left(1 + \frac{q_k^{\mathrm{T}} S_k q_k}{p_k^{\mathrm{T}} q_k}\right) - \frac{S_k q_k p_k^{\mathrm{T}} + p_k q_k^{\mathrm{T}} S_k}{p_k^{\mathrm{T}} q_k}$	$M_{k+1} = M_k + \frac{q_k q_k^{\mathrm{T}}}{q_k^{\mathrm{T}} p_k} - \frac{M_k p_k p_k^{\mathrm{T}} M_k}{p_k^{\mathrm{T}} M_k p_k}$

ΠΑΡΑΤΗΡΗΣΗ: θεωρούμε την τετραγωνική συνάρτηση $f(x) = a + b^{\mathrm{T}} x + \frac{1}{2} x^{\mathrm{T}} C x$ και τις τιμές της κλίσης αυτής $g(x_k) = \nabla f(x_k) = b + C x_k \quad k = 0,1,\ldots,n$ στα σημεία $x_0, x_1, \ldots, x_n$. Ας είναι

$$p_k = x_{k+1} - x_k$$
$$q_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$
$$k = 0,1,\ldots,n-1$$

Αν τα διανύσματα $p_0, p_1, \ldots, p_{n-1}$ είναι γραμμικά ανεξάρτητα ,τότε $C = QP^{-1}$ όπου

$Q = [q_0, q_1, \ldots, q_{n-1}](n \times n)$

$P = [p_0, p_1, \ldots, p_{n-1}](n \times n)$

Η παραπάνω παρατήρηση μας λέει οτι η μήτρα $2^{\omega\nu}$ παραγώγων μιας τετραγωνικής συνάρτησης είναι δυνατόν να προσδιοριστεί ακριβώς με μοναδική πληροφορία την κλίση της συνάρτησης σε σημεία του χώρου.

Τα βήματα του αλγορίθμου BFGS είναι ίδια με πριν .Το μόνο που αλλάζει είναι ο τύπος ανανέωσης της μήτρας S_k όπως φαίνεται και στον πίνακα.

Βήμα 0: $x_0 \in \mathbb{R}^n$, S_0 συμμετρική,θετικά ορισμένη (για παράδειγμα $S_0 = I$) , $k = 0$.

Βήμα 1: $h_k = -S_k \nabla f(x_k)$

Βήμα 2: $\lambda_k = \arg\min_{\lambda \geq 0} \{ f(x_k + \lambda h_k) \}$

Βήμα 3: $x_{k+1} = x_k + \lambda_k h_k$

$$p_k = x_{k+1} - x_k = \lambda_k h_k$$

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

Βήμα 4: Ανανέωση της μήτρας S.

$$S_{k+1} = S_k + \frac{p_k p_k^{\mathrm{T}}}{p_k^{\mathrm{T}} q_k} \left(1 + \frac{q_k^{\mathrm{T}} S_k q_k}{p_k^{\mathrm{T}} q_k} \right) - \frac{S_k q_k p_k^{\mathrm{T}} + p_k q_k^{\mathrm{T}} S_k}{p_k^{\mathrm{T}} q_k}$$

Βήμα 5: Αν $|| \nabla f(x_{k+1}) || = 0$ σταμάτησε.

Αλλιώς θέσε $k = k + 1$ και πήγαινε στο Βήμα 1.

Για τη μέθοδο BFGS ισχύει: $S_{k+1} q_k = S_k q_k + p_k \left(1 + \frac{q_k^{\mathrm{T}} S_k q_k}{p_k^{\mathrm{T}} q_k} \right) - \frac{S_k q_k p_k^{\mathrm{T}} q_k + p_k q_k^{\mathrm{T}} S_k q_k}{p_k^{\mathrm{T}} q_k} = p_k$

επομένως ικανοποιείται η εξίσωση quasi newton.

Κάθε επανάληψη της μεθόδου BFGS μπορεί να εκτελεστεί με πολυπλοκότητα $O(n^2)$ αριθμητικές πράξεις[5](επιπλέον προστίθεται το κόστος του υπολογισμού των τιμών της συνάρτησης και της κλίσης της) . Ο αλγόριθμος είναι εύρωστος και έχει υπεργραμμική ταχύτητα σύγκλισης ,κάτι το οποίο είναι αρκετά γρήγορο για τις περισσότερους πρακτικούς σκοπούς. Αν και η μέθοδος του $NEWTON$ συγκλίνει γρηγορότερα(τετραγωνική ταχύτητα σύγκλισης), η πολυπλοκότητα της σε κάθε επανάληψη είναι μεγαλύτερη, επειδή απαιτεί τη λύση ενός γραμμικού συστήματος $O(n^3)$.

Αριθμητικές δοκιμές καταδεικνύουν την υπεροχή της μεθόδου BFGS σε σχέση τόσο με την DFP όσο και με τις υπόλοιπες $QUASI - NEWTON$ μεθόδους και γι'αυτό το λόγο γενικά προτιμάται. Μάλιστα η μέθοδος BFGS επηρεάζεται λιγότερο από λάθη στην έρευνα γραμμής σε σχέση με τη μέθοδο DFP[1],[11],[12] .

ΚΕΦΑΛΑΙΟ 5

Έρευνα γραμμής-Εύρεση μήκους βήματος

5.1 Μέθοδοι εύρεσης μήκους βήματος

Σε κάθε επανάληψη μιας μεθόδου έρευνας γραμμής υπολογίζεται μια κατεύθυνση έρευνας h_k και αποφασίζεται πόσο μακριά θα κινηθεί κατά μήκος αυτής της κατεύθυνσης(μήκος βήματος, λ_k). Όπως ειπώθηκε και σε προηγούμενο κεφάλαιο κάθε επόμενο σημείο x_{k+1} της ακολουθίας εξαρτάται από το προηγούμενο x_k με τον εξής τρόπο : $x_{k+1} = x_k + \lambda_k h_k$. Η επιτυχία μιας έρευνας γραμμής εξαρτάται από την αποτελεσματική επιλογή τόσο της κατεύθυνσης h_k όσο και του μήκους βήματος λ_k . Οι περισσότεροι αλγόριθμοι απαιτούν η h_k να είναι κατεύθυνση καθόδου, δηλαδή $h_k^{\mathrm{T}} \nabla f(x_k) < 0$, διότι η ιδιότητα αυτή εξασφαλίζει ότι οι τιμές της συνάρτησης μειώνονται κατά μήκος της[7],[12].

Όσον αφόρα την εύρεση του μήκους βήματος(step length) επιθυμούμε να επιλέξουμε το λ_k με τέτοιο τρόπο ώστε να μειώνονται οι τιμές της συνάρτησης f και παράλληλα ο χρόνος για την εύρεση του να είναι μικρός. Η ιδανική επιλογή είναι να θεωρηθεί ως μήκος βήματος λ_k αυτό που προκύπτει από την σχέση:

$$f(x_i + \lambda_i h_i) = \min_{\lambda \geq 0} \{ f(x_i + \lambda h_i) : \lambda_i \geq 0 \}$$

Κάτι τέτοιο όμως ,δηλαδή η εύρεση ενός λ_i που ελαχιστοποιεί την $f(x + \lambda h)$ απαιτεί πολλούς υπολογισμούς της συνάρτησης f και πιθανώς και της κλίσης ∇f . Γι'αυτό το λόγο έχουν αναπτυχθεί περισσότερο πρακτικές μέθοδοι προσεγγιστικού υπολογισμού του ελαχίστου μιας μεταβλητής. Οι μέθοδοι ελαχιστοποίησης συναρτήσεων μιας μεταβλητής χωρίζονται σε δυο κατηγορίες , σε αυτές που χρησιμοποιούν μόνο τιμές της συνάρτησης $\varphi(\lambda)$ και αυτές που χρησιμοποιούν τιμές της $\varphi(\lambda)$ και των παραγώγων της $\varphi'(\lambda)$ και $\varphi''(\lambda)$, όπου ισχύει:

$$\varphi(\lambda) = f(x_i + \lambda h_i)$$

$$\varphi'(\lambda) = \frac{d\varphi(\lambda)}{d\lambda} = \nabla f(x_i + \lambda h_i)^{\mathrm{T}} h_i$$

$$\varphi''(\lambda) = \frac{d^2\varphi(\lambda)}{d\lambda^2} = h_i^{\mathrm{T}} \frac{\partial^2 f(x_i + \lambda h_i)}{\partial \lambda^2} h_i$$

Στην πρώτη κατηγορία ανήκουν η έρευνα γραμμής $Fibonacci$ η οποία όμως δεν έχει ιδιαίτερη πρακτική εφαρμογή, η έρευνα της χρυσής τομής ($Goldensection\ \ method$) , η

οποία διαφέρει σε σχέση με την προηγούμενη , ως προς το ότι ο αριθμός N των υπολογισμών της συνάρτησης f μέσα στο αρχικό διάστημα αβεβαιότητας δεν είναι καθορισμένος από πριν και η τετραγωνική προσαρμογή (*quadratic interpolation*).

Στην τετραγωνική παρεμβολή- *quadratic interpolation* διαθέτουμε τις τιμές της συνάρτησης σε τρία σημεία $\lambda_1, \lambda_2, \lambda_3$ δηλαδή $f(\lambda_1), f(\lambda_2), f(\lambda_3)$ και επιθυμούμε να προσαρμόσουμε μια τετραγωνική συνάρτηση $q(\lambda) = \alpha + b\lambda + \lambda^2$ στα σημεία αυτά δηλαδή να ισχύει $q(\lambda_t) = f_i$. Η ζητούμενη αυτή συνάρτηση είναι η $q(\lambda) = f_1 \frac{(\lambda-\lambda_2)(\lambda-\lambda_3)}{(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)} + f_2 \frac{(\lambda-\lambda_1)(\lambda-\lambda_3)}{(\lambda_2-\lambda_1)(\lambda_2-\lambda_3)} + f_3 \frac{(\lambda-\lambda_1)(\lambda-\lambda_2)}{(\lambda_3-\lambda_1)(\lambda_3-\lambda_2)}$.Το νέο σημείο λ_4 ορίζεται σαν το ελάχιστο της $q(\lambda)$ δηλαδή σαν το σημείο όπου $\frac{dq(\lambda_4)}{d\lambda} = 0$. Το σημείο αυτό δίνεται από τον τύπο: $\lambda_4 = \frac{1}{2} \frac{b_{23} f_1 + b_{31} f_2 + b_{12} f_3}{a_{23} f_1 + a_{31} f_2 + a_{12} f_3}$ όπου $a_{ij} = \lambda_i - \lambda_j$ $b_{ij} = \lambda_i^2 - \lambda_j^2$ $i, j = 1,2,3.$ Από τα τέσσερα σημεία $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ τα τρία από αυτά επιλέγονται για την επόμενη τετραγωνική προσαρμογή με τέτοιο τρόπο ώστε να εξασφαλίζεται η σύγκλιση της μεθόδου .

Απαραίτητη προϋπόθεση για την σύγκλιση είναι τα σημεία $\lambda_1, \lambda_2, \lambda_3$ όπου γίνεται η προσαρμογή να ικανοποιούν: $\lambda_1 < \lambda_2 < \lambda_3$ και $\begin{matrix} f(\lambda_2) \le f(\lambda_1) \\ f(\lambda_2) \le f(\lambda_3) \end{matrix}$.Από τα τέσσερα σημεία $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ εκλέγονται τα τρία που ικανοποιούν την παραπάνω συνθήκη. Η μέθοδος συγκλίνει υπεργραμμικά με τάξη σύγκλισης 1,3 και έχει συχνή πρακτική εφαρμογή.

Στην δεύτερη κατηγορία , όπου χρησιμοποιούνται τιμές και της συνάρτησης και των παραγώγων της , ανήκουν η μέθοδος του *NEWTON* , ο κανόνας *Armijo* , ο κανόνας *Wolfe* , ο κανόνας *Goldstein* και η κυβική παρεμβολή(*cubic interpolation*).

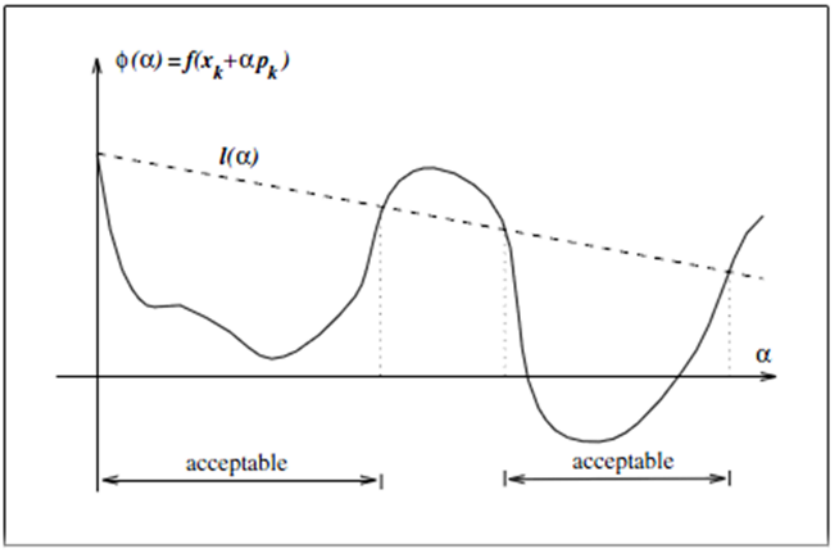
5.2 Κανόνας Armijo

Ο κανόνας *Armijo* [12] ο οποίος μάλιστα χρησιμοποιήθηκε σε ένα από τα προγράμματα που υλοποιήσαμε στο *MATLAB* , έχει την εξής δομή: έστω $f(x_i), \nabla f(x_i), h_i, a \in (0,1), \beta \in (0,1)$. Ο κανόνας *Armijo* δεν προσπαθεί να βρει το ελάχιστο στην κατεύθυνση h_i , απλώς απαιτεί η συνάρτηση f να μειώνεται αρκετά από το σημείο x_i μέχρι το επόμενο $x_{i+1} = x_i + \beta^\eta h_i$. Το ζητούμενο είναι η εύρεση του ελάχιστου μη-αρνητικού ακέραιου η που ικανοποιεί τη σχέση :

$$f(x_i + \beta^\eta h_i) - f(x_i) \le a\beta^\eta \nabla f(x_i)^{\mathrm{T}} h_i$$

Αρχίζουμε με $\eta = 0$ και ελέγχουμε την παραπάνω ανισότητα, αν δεν ικανοποιείται δοκιμάζουμε με $\eta = 1$ κτλ μέχρις ότου η ανισότητα ικανοποιηθεί. Αυτό θα συμβαίνει πάντοτε , δηλαδή ο κανόνας *Armijo* θα συγκλίνει πάντα εφόσον $\nabla f(x_i)^{\mathrm{T}} h_i < 0$. Λόγω

της απλότητας και της εξασφαλισμένης του σύγκλισης ο κανόνας $Armijo$ έχει συχνή πρακτική εφαρμογή.

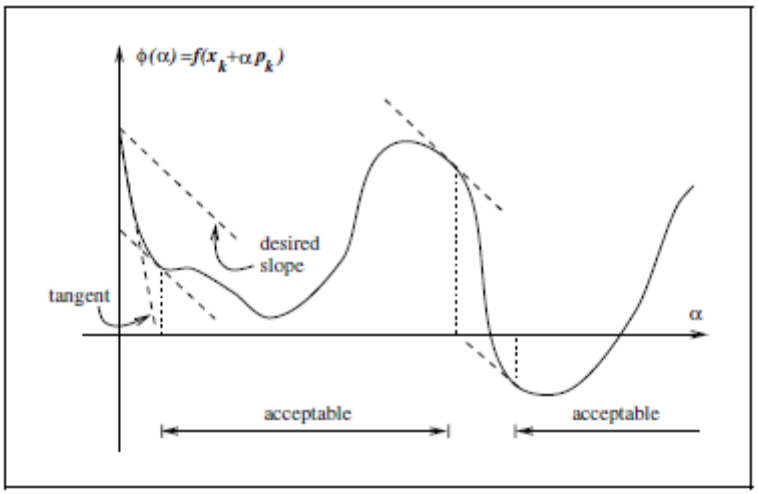


5.3 Κανόνας Wolfe

Ο κανόνας $Wolfe$ είναι ένα σύνολο ανισοτήτων για την εκτέλεση μιας έρευνας γραμμής, αποτελεσματικότερο συγκριτικά με τον κανόνα $Armijo$, διότι περιλαμβάνει και μία επιπλέον συνθήκη[12]. Έστω $\varphi(\lambda) = f(x_i + \lambda h_i)$,τότε $\varphi'(\lambda) = \frac{d\varphi(\lambda)}{d\lambda} = \nabla f(x_i + \lambda h_i)^{\mathrm{T}} h_i$. Ο κανόνας $Wolfe$ απαιτεί το μήκος βήματος λ_i να ικανοποιεί τις εξής συνθήκες :

$i) f(x_i + \lambda_i h_i) \le f(x_i) + m_1 \lambda_i \nabla f(x_i)^{\mathrm{T}} h_i$ (sufficient decrease condition- $Armijo\ rule$)

$ii) \nabla f(x_i + \lambda_i h_i)^{\mathrm{T}} h_i \ge m_2 \nabla f(x_i)^{\mathrm{T}} h_i$ (the curvature condition)

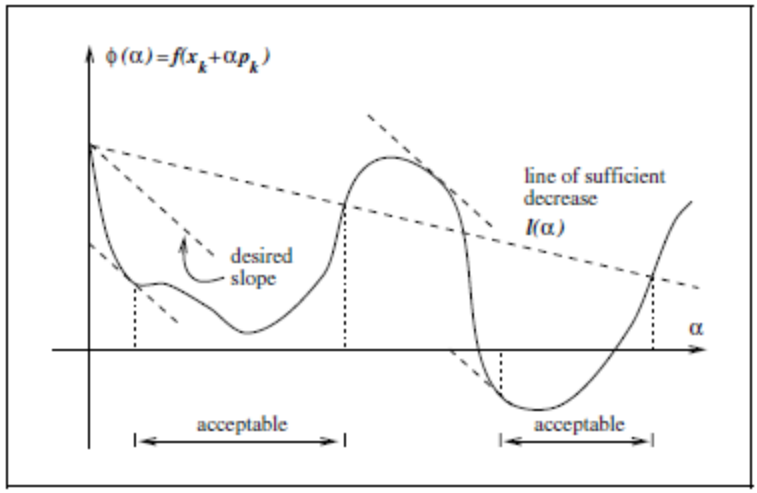


όπου ισχύει $0 < m_1 < m_2 < 1$.

Ένα μήκος βήματος λ_i μπορεί να ικανοποιεί τις συνθήκες *Wolfe* χωρίς απαραίτητα να βρίσκεται κοντά σε σημείο που ελαχιστοποιεί την φ. Μπορούμε να τροποποιήσουμε την συνθήκη ii ώστε να εξασφαλίσουμε ότι το λ_i θα βρίσκεται σε μια περιοχή ενός στάσιμου σημείου της φ. Στις ισχυρές συνθήκες *Wolfe* (strong Wolfe conditions) η δεύτερη συνθήκη τροποποιείται ως εξής:

$$\left|\nabla f(x_i + \lambda_i h_i)^{\mathrm{T}} h_i\right| \le m_2 \left|\nabla f(x_i)^{\mathrm{T}} h_i\right|$$

Η μόνη διαφορά σε σχέση με τις συνθήκες *Wolfe* (*Wolfe conditions*) είναι ότι πλέον δεν επιτρέπουμε στην παράγωγο $\varphi'(\lambda_i)$ να είναι θετική. Με αυτόν τον τρόπο αποκλείονται σημεία τα οποία είναι μακριά από στάσιμα σημεία της φ.



Step lengths satisfying the Wolfe conditions.

5.4 Κυβική προσαρμογή (Cubic interpolation)

Η μέθοδος της κυβικής προσαρμογής(cubic interpolation)[1],[5],[6],[14] υπολογίζει διαδοχικά σε κάθε επανάληψη ένα κατάλληλο διάστημα $[\lambda_1, \lambda_2]$, μέσα στο οποίο η συνάρτηση φ έχει ένα τοπικό ελάχιστο και με αυτόν τον τρόπο επιτυγχάνεται επιτάχυνση της έρευνας γραμμής. Έχουμε ως δεδομένα της τιμές της φ στα λ_1, λ_2 και τιμές των παραγώγων της στα ίδια σημεία δηλαδή :

$$\varphi(\lambda_1) = f(x+\lambda_1 h), \varphi(\lambda_2) = f(x+\lambda_2 h)$$
$$\varphi'(\lambda_1) = \nabla f(x+\lambda_1 h)^{\mathrm{T}} h, \varphi'(\lambda_2) = \nabla f(x+\lambda_2 h)^{\mathrm{T}} h$$

Σκοπός της μεθόδου είναι η προσαρμογή ενός τριτοβάθμιου πολυωνύμου στα δεδομένα αυτά (η κυβική αυτή συνάρτηση υπάρχει πάντα και είναι μοναδική), το οποίο στην συνέχεια ελαχιστοποιείται και το ελάχιστο του δίνει το νέο σημείο $\hat{\lambda}$.

Για το νέο σημείο $\hat{\lambda}$ ισχύει $\hat{\lambda} = \lambda_2 + \tilde{r}(\lambda_1 - \lambda_2)$ όπου $\tilde{r} = \dfrac{\varphi_2' + p - d}{\varphi_1' + \varphi_2' + 2p}$

$d = sgn(\lambda_2 - \lambda_1)\sqrt{D}$, $p = \varphi_1' + \varphi_2' - 3\dfrac{\varphi_2 - \varphi_1}{\lambda_2 - \lambda_1}$ $D = p^2 - \varphi_1'\varphi_2'$.

Η τάξη σύγκλισης της μεθόδου είναι τετραγωνική.

Μπορούν να χρησιμοποιηθούν και συνδυασμοί των παραπάνω μεθόδων προκειμένου να επιτευχθεί βελτίωση της ακρίβειας στην εύρεση του μήκους βήματος λ_i.

Στην υλοποίηση στο *matlab* χρησιμοποιήθηκε συνδυασμός κανόνα *Wolfe* και κυβικής παρεμβολής ως εξής: ξεκινάμε από ένα αρχικό μήκος βήματος $\lambda > 0$ και αρχικοποιούμε $\lambda_L = 0, \lambda_R = 0$.Η μέθοδος τερματίζει όταν η διαφορά $\lambda_R - \lambda_L$ είναι μικρότερη από κάποια πολύ ''μικρή σταθερά που επιλέγουμε και $\lambda_R \neq 0$. Εάν ικανοποιούνται και οι δύο συνθήκες του κανόνα *Wolfe* τότε τερματίζει η διαδικασία εύρεση μήκους βήματος.Αν η 1η από τις 2 συνθήκες δεν ικανοποιείται τότε θέτουμε $\lambda_R = \lambda$ και εφαρμόζουμε κυβική παρεμβολή σύμφωνα με τους τύπους που αναφέρθηκαν παραπάνω και βρίσκουμε το νέο μήκος βήματος $\hat{\lambda}$. Εάν $\lambda_R = 0$ πρέπει να εξασφαλίσουμε ώστε το $\hat{\lambda}$ να βρίσκεται δεξιά του λ_L (extrapolation),όποτε και αντικαθιστούμε το νέο μήκος βήματος $\hat{\lambda}$ με το $\max\{\hat{\lambda}, a\lambda\}$, $a > 1$, εάν κάποιο $\lambda_R > 0$ είναι ήδη διαθέσιμο(interpolation), τότε πρέπει να εξασφαλίσουμε το $\hat{\lambda}$ να βρίσκεται στο διάστημα $[\lambda_L, \lambda_R]$,γι' αυτό το λόγο αντικαθιστούμε το $\hat{\lambda}$ διαδοχικά με $\hat{\lambda} = \min\{\hat{\lambda}, \lambda_R - \theta(\lambda_R - \lambda_L)\}$ και μετά με $\hat{\lambda} = \max\{\hat{\lambda}, \lambda_L + \theta(\lambda_R - \lambda_L)\}$, $\theta \in [0, \frac{1}{2}]$. Εάν η 1η συνθήκη ικανοποιείται αλλά η 2η δεν

ικανοποιείται τότε θέτουμε $\lambda_L = \lambda$ και συνεχίζουμε με τον ίδιο τρόπο όπως και στην προηγούμενη περίπτωση.

Συγκεκριμένα σύμφωνα με τους Lemarechal, Bonnans, Gilbert[6] έχουμε τις εξής περιπτώσεις :

$i)$ αν $f(x_i + \lambda_i h_i) \le f(x_i) + m_1 \lambda_i \nabla f(x_i)^{\mathrm{T}} h_i$ και $\nabla f(x_i + \lambda_i h_i)^{\mathrm{T}} h_i \ge m_2 \nabla f(x_i)^{\mathrm{T}} h_i$ τότε τερμάτισε.

$ii)$ αν $f(x_i + \lambda_i h_i) > f(x_i) + m_1 \lambda_i \nabla f(x_i)^{\mathrm{T}} h_i$ τότε $\lambda_R = \lambda.$

$iii)$ αν $f(x_i + \lambda_i h_i) \le f(x_i) + m_1 \lambda_i \nabla f(x_i)^{\mathrm{T}} h_i$ και $\nabla f(x_i + \lambda_i h_i)^{\mathrm{T}} h_i < m_2 \nabla f(x_i)^{\mathrm{T}} h_i$ τότε $\lambda_L = \lambda.$

$0 < m_1 < m_2 < 1.$

ΚΕΦΑΛΑΙΟ 6

Νέα προτεινόμενη μέθοδος Quasi-Newton

6.1 Περιγραφή της μεθόδου

Η λογική στην οποία στηρίζεται η προτεινόμενη *quasi newton* μέθοδος είναι η εξής:

Έχουμε ότι:

$$q_j = \nabla f(x_{j+1}) - \nabla f(x_j)$$

$$p_j = x_{j+1} - x_j$$

$$q_j = \nabla f(x_{j+1}) - \nabla f(x_j) = \nabla f(x_j + p_j) - \nabla f(x_j) = \int_0^1 \frac{\partial^2 f(x_j + \sigma p_j)}{\partial x^2} d\sigma\, p_j = H_j p_j$$

οπότε:

$$H_j = \int_0^1 \frac{\partial^2 f(x_j + \sigma p_j)}{\partial x^2} d\sigma,\ j = k, k-1, \ldots, k-n+1$$

Τότε

$$Q_k \triangleq \left[q_k, q_{k-1}, \ldots, q_{k-n+1}\right] = \left[\nabla f(x_{k+1}) - \nabla f(x_k), \nabla f(x_k) - \nabla f(x_{k-1}), \ldots, \nabla f(x_{k-n+2}) - \nabla f(x_{k-n+1})\right] =$$

$$= \left[H_k p_k, H_{k-1} p_{k-1}, \ldots, H_{k-n+1} p_{k-n+1}\right]$$

Ας είναι $\bar{H} = \frac{1}{n}\sum_{i=0}^{n-1} H_{k-i}$ και $\Delta_j = H_j - \bar{H},\ j = k, k-1, \ldots, k-n+1$

Ας είναι $B_k = \left[p_k, p_{k-1}, \ldots, p_{k-n+1}\right] = \left[x_{k+1} - x_k, x_k - x_{k-1}, \ldots, x_{k-n+2} - x_{k-n+1}\right]$

Τότε

$$Q_k = \left[H_k p_k, H_{k-1} p_{k-1}, \ldots, H_{k-n+1} p_{k-n+1}\right] = \left[(\bar{H} + \Delta_k) p_k, (\bar{H} + \Delta_{k-1}) p_{k-1}, \ldots, (\bar{H} + \Delta_{k-n+1}) p_{k-n+1}\right] =$$

$$= \left[\bar{H} p_k, \bar{H} p_{k-1}, \ldots, \bar{H} p_{k-n+1}\right] + \left[\Delta_k p_k, \Delta_{k-1} p_{k-1}, \ldots, \Delta_{k-n+1} p_{k-n+1}\right]$$

$$= \bar{H} B_k + \left[\Delta_k p_k, \Delta_{k-1} p_{k-1}, \ldots, \Delta_{k-n+1} p_{k-n+1}\right]$$

Αν υποθέσουμε ότι η ακολουθία συγκλίνει, τότε για j "μεγάλο" θα ισχύει $H_k \cong H_{k-1} \cong H_{k-2} \cong \ldots H_{k-n+1} \cong \bar{H}$ άρα $\Delta_j = H_j - \bar{H} \cong 0$, επομένως ο όρος $\left[\Delta_k p_k, \Delta_{k-1} p_{k-1}, \ldots, \Delta_{k-n+1} p_{k-n+1}\right]$ μπορεί να παραλειφθεί , άρα $Q_k \simeq \bar{H} B_k$ (ικανοποιείται η εξίσωση quasi newton για n προηγούμενες επαναλήψεις).'Αρα τελικά $\bar{H}^{-1} \cong \left[\frac{\partial^2 f(x_{k+1})}{\partial x^2}\right]^{-1} \cong B_k Q_k^{-1}$

Όπως φαίνεται από την παραπάνω ανάλυση αυτό που διαφοροποιεί την δική μας προτεινόμενη μέθοδο σε σχέση με τις άλλες quasi-newton μεθόδους είναι ότι σε κάθε επανάληψη ανανέωσης των μητρών χρησιμοποιείται πληροφορία άμεσα από τις n προηγούμενες επαναλήψεις, όπου n είναι ο αριθμός των μεταβλητών της υπό ελαχιστοποίηση συνάρτησης, ενώ στις υπόλοιπες μεθόδους (DFP,BFGS) χρησιμοποιείται πληροφορία μόνο από την προηγούμενη επανάληψη.

6.2 Αλγόριθμος προτεινόμενης μεθόδου

Για την επίλυση του μη-περιορισμένου προβλήματος προτείνεται η εξής μέθοδος:

Βήμα 0:αρχικές τιμές: $x_0 \in \mathbb{R}^n, W_0 = I_n, B_0 \in R^n, d > 0, \beta \in (0,1), \alpha \in \left(0, \frac{1}{2}\right), 1 >> \varepsilon > 0, k = 0$

Βήμα 1: υπολογισμός κατεύθυνσης έρευνας h_k : επίλυση ως προς s_k των γραμμικών εξισώσεων. $s_k = W_k \nabla f(x_k)$

Περίπτωση 1: $h_k = -B_k s_k$ αν $\nabla f(x_k)^{\mathrm{T}} B_k s_k \geq d \,\|\nabla f(x_k)\|_2^2$

Περίπτωση 2: $h_k = -\nabla f(x_k)$ αν $\nabla f(x_k)^{\mathrm{T}} B_k s_k < d \,\|\nabla f(x_k)\|_2^2$

Βήμα 2: υπολογισμός μήκους βήματος λ_k με τον κανόνα $Armijo$: εύρεση του ελάχιστου μη-αρνητικού ακεραίου η_k που ικανοποιεί τη σχέση:
$f(x_k + \beta^{\eta_k} h_k) - f(x_k) \leq -ad\beta^{\eta_k} \,\|\nabla f(x_k)\|_2^2$
$\lambda_\kappa = \beta^{\eta_k}$

Βήμα 3:ενημέρωση τρέχοντος σημείου x_k και κριτήριο τερματισμού: $x_{k+1} = x_k + \lambda_k h_k$ αν $\|\nabla f(x_{k+1})\|_2^2 < \varepsilon$ τερματισμός.

Βήμα 4:ενημέρωση μητρών $W_k, B_k : i = \left(k \bmod (n)\right) + 1$

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

$$q_i = \nabla f\left(x_{k+1}\right) - \nabla f\left(x_k\right)$$

$$\overline{p}_i = p_i - B_k e_i = x_{k+1} - x_k - B_k e_i$$

$$B_{k+1} = B_k + \overline{p}_i e_i^{\mathrm{T}} = \left[B_k^i\right]_{p_i}$$

$$W_{k+1} = W_k - \frac{W_k q_i e_i^{\mathrm{T}} W_k - e_i e_i^{\mathrm{T}} W_k}{e_i^{\mathrm{T}} W_k q_i}$$

$k = k+1$ και επιστροφή στο Βήμα 1.

(όπου $\left[A_b^i\right]$ συμβολίζει τη μήτρα που προκύπτει από την A με αντικατάσταση της i στήλης της από το διάνυσμα b)

Το πρόγραμμα στο *MATLAB* υλοποιεί τον παραπάνω αλγόριθμο χρησιμοποιώντας τον πίνακα Q_k αντί για τον W_k , όπου ισχύει $W_k = Q_k^{-1}$.

Πιο συγκεκριμένα στο Βήμα 1 η υλοποίηση στο *MATLAB* έχει ως εξής:

$$Q_k s_k = \nabla f(x_k) \Leftrightarrow s_k = Q_k^{-1} \nabla f(x_k)$$

Και στο βήμα 4:

$$\overline{q}_i = q_i - Q_k e_i = \nabla f\left(x_{k+1}\right) - \nabla f\left(x_k\right) - Q_k e_i$$

$$Q_{k+1} = Q_k + \overline{q}_i e_i^{\mathrm{T}} = \left[Q_k^i\right]_{p_i}$$

Η ισοδυναμία στη χρήση των πινάκων Q_k, W_k αποδεικνύεται παρακάτω:

$$W_k \overline{q}_i = W_k q_i - e_i$$

$$W_{k+1} = Q_{k+1}^{-1} = \left[Q_k + \overline{q}_i e_i^{\mathrm{T}}\right]^{-1} = Q_k^{-1} - \frac{Q_k^{-1}{}_k \overline{q}_i e_i^{\mathrm{T}} Q_k^{-1}}{1 + e_i^{\mathrm{T}} Q_k^{-1}{}_k \overline{q}_i} =$$

$$= W_k - \frac{W_k \overline{q}_i e_i^{\mathrm{T}} W_k}{1 + e_i^{\mathrm{T}} W_k \overline{q}_i} = W_k - \frac{(W_k q_i - e_i) e_i^{\mathrm{T}} W_k}{1 + e_i^{\mathrm{T}} (W_k q_i - e_i)}$$

$$= W_k - \frac{(W_k q_i - e_i)\left(e_i^{\mathrm{T}} W_k\right)}{1 + e_i^{\mathrm{T}} W_k q_i - e_i^{\mathrm{T}} e_i} = W_k - \frac{W_k q_i e_i^{\mathrm{T}} W_k - e_i e_i^{\mathrm{T}} W_k}{e_i^{\mathrm{T}} W_k q_i}$$

Ο προγραμματισμός της μεθόδου στο MATLAB γίνεται με τη χρήση του πίνακα Q_k αντί για τον W_k , για λόγους μεγαλύτερης αριθμητικής ακρίβειας, παρόλο που ο αριθμός των πράξεων είναι μεγαλύτερος.

ΚΕΦΑΛΑΙΟ 7

Αριθμητικά παραδείγματα και συγκρίσεις

Στους παρακάτω πίνακες φαίνεται ο αριθμός των επαναλήψεων για τον υπολογισμό του ελαχίστου συναρτήσεων επιλέγοντας διάφορα αρχικά σημεία x_0, εξετάζοντας με αυτόν τον τρόπο την αποτελεσματικότητα συγκεκριμένων μεθόδων $qua\,\mathrm{si}-newton$.Συγκεκριμένα υλοποιούνται έξι προγράμματα στο περιβάλλον του MATLAB , στα οποία τρέχουμε διάφορες συναρτήσεις (test functions).Σε όλες τις συναρτήσεις αναφέρεται η ακρίβεια ($epsilon$) που έχει χρησιμοποιηθεί για τους αλγορίθμους.

Επεξήγηση ονομάτων αλγορίθμων

NEWMETHODARMIJO: νέος προτεινόμενος αλγόριθμος με χρήση κανόνα Armijo για την εύρεση του μήκους βήματος.

BFGS MATLAB: Αλγόριθμος BFGS του MATLAB

DFP MATLAB: Αλγόριθμος DFP του MATLAB

NEWMETHODWOLFECUBIC: νέος προτεινόμενος αλγόριθμος με χρήση κανόνα Wolfe και κυβικής παρεμβολής για την εύρεση του μήκους βήματος.

BFGSWOLFECUBIC: Δική μας υλοποίηση του αλγορίθμου BFGS με χρήση κανόνα Wolfe και κυβικής παρεμβολής για την εύρεση του μήκους βήματος.

DFPWOLFECUBIC: Δική μας υλοποίηση του αλγορίθμου DFP με χρήση κανόνα Wolfe και κυβικής παρεμβολής για την εύρεση του μήκους βήματος.

Για κάθε μέθοδο έχουμε θέσει έναν μέγιστο αριθμό επαναλήψεων.Συγκεκριμένα ισχύουν τα εξής:

NEWMETHODARMIJO: maximum number of iterations=1500.

BFGS MATLAB: maximum number of iterations=1500.

DFP MATLAB: maximum number of iterations=4000.

NEWMETHODWOLFECUBIC: maximum number of iterations=1500.

BFGSWOLFECUBIC: maximum number of iterations=1500.

DFPWOLFECUBIC: maximum number of iterations=4000.

Το σύμβολο – δηλώνει οτι η συνάρτηση δεν συγκλίνει στο επιθυμητό σημείο εξαντλώντας τον ορισμένο αριθμό επαναλήψεων για τις περιπτώσεις των αλγορίθμων NEWMETHODARMIJO, BFGS MATLAB, NEWMETHODWOLFECUBIC, BFGSWOLFECUBIC, DFPWOLFECUBIC ενώ για τον αλγόριθμο DFP MATLAB δείχνει ότι σταματάει να εκτελείται η ρουτίνα του MATLAB επειδή υπερβαίνει έναν ορισμένο αριθμό υπολογισμών της συνάρτησης(4000 function evaluations).

ΠΑΡΑΔΕΙΓΜΑ 1

ROSENBROCK FUNCTION [18] (2 μεταβλητών)

$$f_2(x) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

$$\nabla f_2(x) = \begin{bmatrix} -400x_1\left(x_2 - x_1^2\right) - 2(1 - x_1) \\ 200\left(x_2 - x_1^2\right) \end{bmatrix}$$

$$epsilon = 1e-10$$

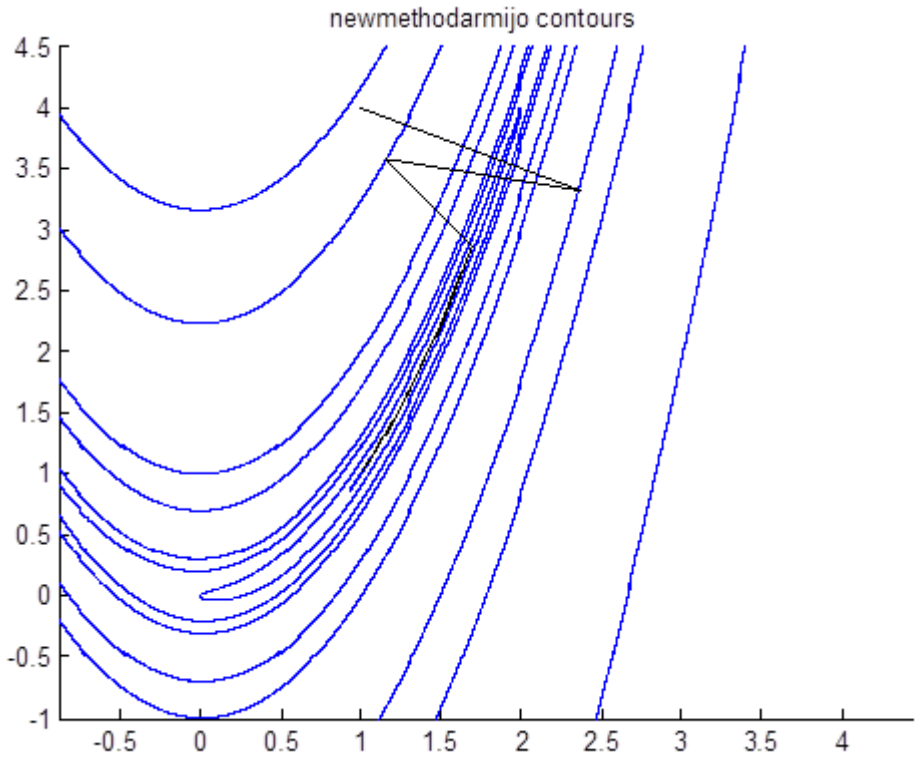
Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0 = [1, 4]$	34	25	468	31	24	66
$x_0 = [-1, 4]$	66	40	1001	41	39	41
$x_0 = [5, -5]$	80	46	995	26	18	234
$x_0 = [-1, 6]$	46	48	82	58	39	38
$x_0 = [-1, -6]$	18	22	240	26	27	142
$x_0 = [2, 16]$	98	45	660	55	41	35
$x_0 = [16, 16]$	116	58	1118	67	54	187
$x_0 = [-12, 5]$	126	53	2451	67	51	196
$x_0 = [-12, -5]$	136	35	332	67	59	74
$x_0 = [-18, -15]$	119	46	786	167	117	192
$x_0 = [22, -1]$	147	43	-	51	43	46
$x_0 = [1, -22]$	44	45	784	26	16	38
$x_0 = [30, 30]$	614	66	-	105	82	113
$x_0 = [30, -30]$	323	70	2302	63	55	73
$x_0 = [-50, 30]$	1096	56	3321	157	102	153
$x_0 = [25, 20]$	494	55	-	82	60	541
$x_0 = [-3, -60]$	47	55	184	15	21	34

$x_0 = [-4,-6]$	49	29	307	31	30	39
$x_0 = [10,0]$	125	28	711	39	32	114
$x_0 = [0,0]$	38	21	21	27	20	20

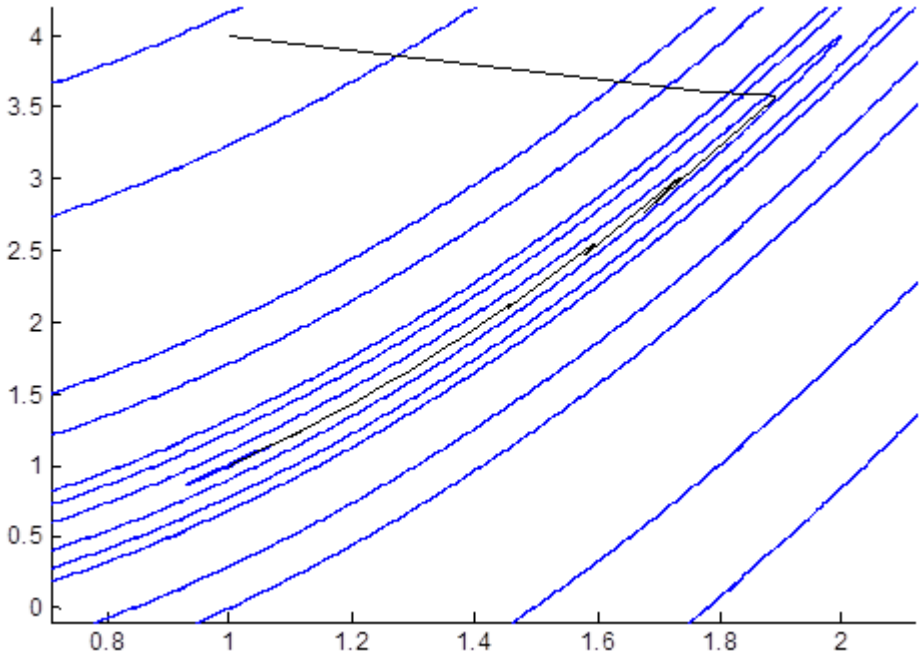
Το επιθυμητό σημείο στο οποίο θέλουμε να συγκλίνουν οι μέθοδοι είναι το $(1,1)$, δηλαδή το ελάχιστο της συνάρτησης *Rosenbrock* δύο μεταβλητών.

Παρακάτω παρατίθενται διάφορες γραφικές παραστάσεις για τις μεθόδους **newmethodarmijo , newmethodwolfecubic , bfgswolfecubic , dfpwolfecubic** .Φαίνονται δηλαδή οι ισοϋψείς της συνάρτησης και η αλληλουχία των σημείων της ακολουθίας που κατασκευάζει ο αλγόριθμος από το αρχικό σημείο μέχρι την επιθυμητή λύση.

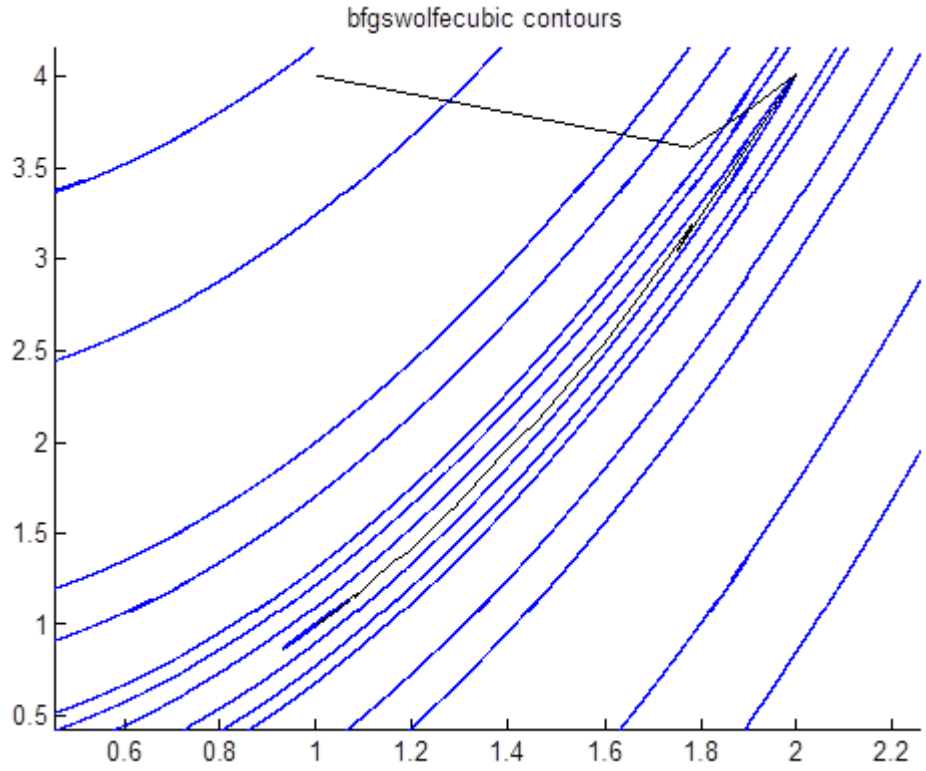
newmethodarmijo με αρχικό σημείο $(1,4)$



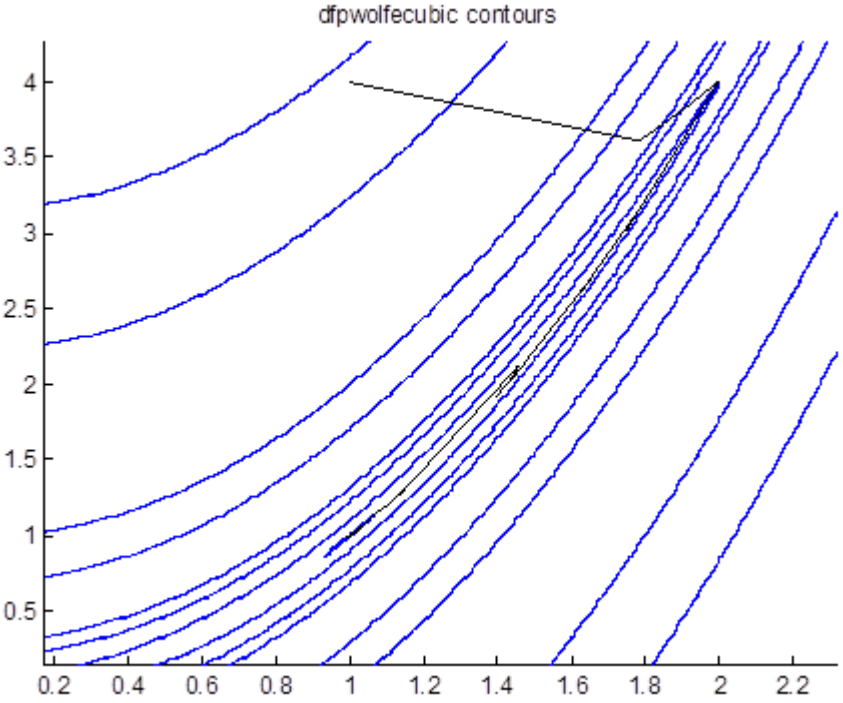
newmethodwolfecubic με αρχικό σημείο $(1,4)$



bfgswolfecubic με αρχικό σημείο $(1,4)$



dfpwolfecubic με αρχικό σημείο $(1,4)$



ΠΑΡΑΔΕΙΓΜΑ 2

HIMMELBLAU'S FUNCTION [18] (2 μεταβλητές)

$$f_2(x) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$$

$$\nabla f_2(x) = \begin{bmatrix} 4x_1^3 + 4x_1x_2 - 42x_1 + 2x_2^2 - 14 \\ 2x_1^2 - 26x_2 + 4x_1x_2 + 4x_2^3 - 22 \end{bmatrix}$$

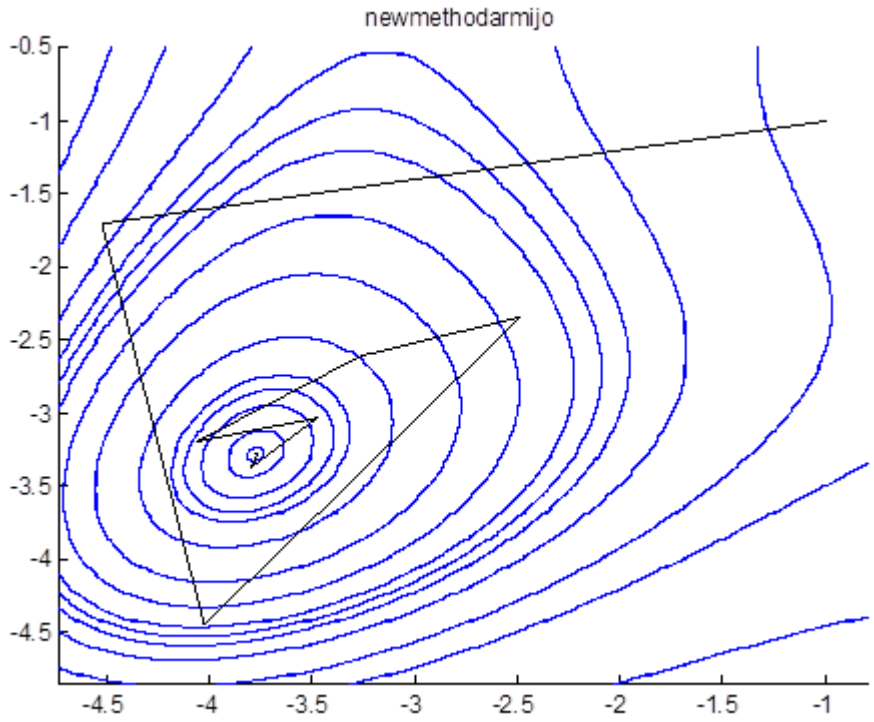
$$epsilon = 1e-10$$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0 = [-1,-1]$	12(c)	9(c)	13(c)	7(c)	9(c)	10(c)
$x_0 = [1,1]$	15(a)	8(a)	9(a)	8(a)	9(a)	9(a)
$x_0 = [10,-10]$	14(b)	17(d)	31(d)	11(d)	10(d)	9(d)
$x_0 = [30,30]$	16(b)	23(a)	250(a)	14(a)	14(a)	20(a)
$x_0 = [6,10]$	13(a)	17(a)	48(a)	8(a)	11(a)	11(a)
$x_0 = [-6,-10]$	15(d)	16(c)	23(c)	11(c)	12(c)	13(c)
$x_0 = [0,0]$	11(d)	10(a)	25(d)	9(a)	9(a)	10(a)
$x_0 = [-4,10]$	9(c)	17(b)	29(b)	9(b)	9(b)	8(b)
$x_0 = [15,17]$	16(b)	21(a)	116(a)	8(a)	17(a)	21(a)
$x_0 = [-15,-17]$	12(d)	19(c)	41(c)	13(c)	11(c)	10(c)
$x_0 = [-0.5,4]$	10(c)	10(b)	9(b)	11(b)	9(b)	9(b)
$x_0 = [-0.5,-4]$	8(b)	11(c)	16(c)	9(c)	10(c)	10(c)

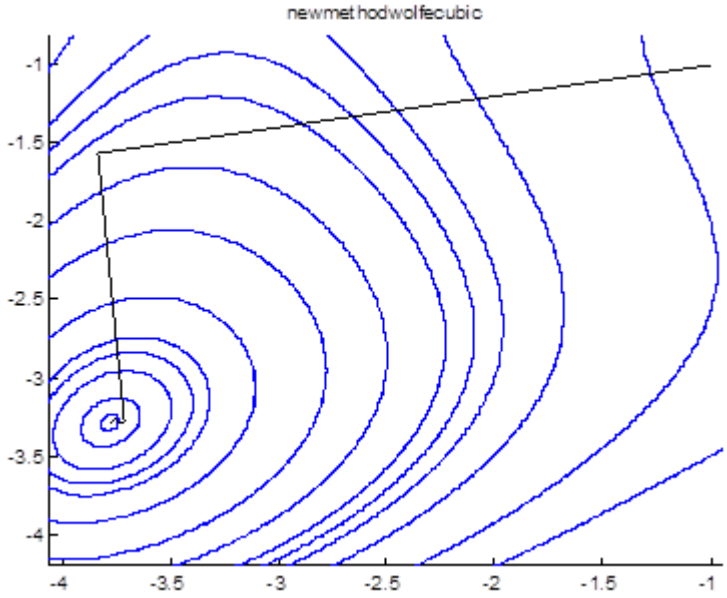
Η συνάρτηση himmelblau έχει τέσσερα τοπικά ελάχιστα, τα $a = f(3.0, 2.0) = 0$, $b = f(-2.805118, 3.131312) = 0$, $c = f(-3.779310, -3.283186) = 0$, $d = f(3.584428, -1.848126) = 0$. Η παρένθεση δίπλα από τον αριθμό των επαναλήψεων δείχνει σε πιο σημείο από τα δύο συγκλίνει ο εκάστοτε αλγόριθμος. Είναι προφανές οτι για να γίνει σύγκριση μεταξύ μεθόδων θα πρέπει να συγκλίνουν στο ίδιο σημείο.

Παρακάτω παρατίθενται και οι γραφικές παραστάσεις για τις μεθόδους **newmethodarmijo , newmethodwolfecubic , bfgswolfecubic , dfpwolfecubic**

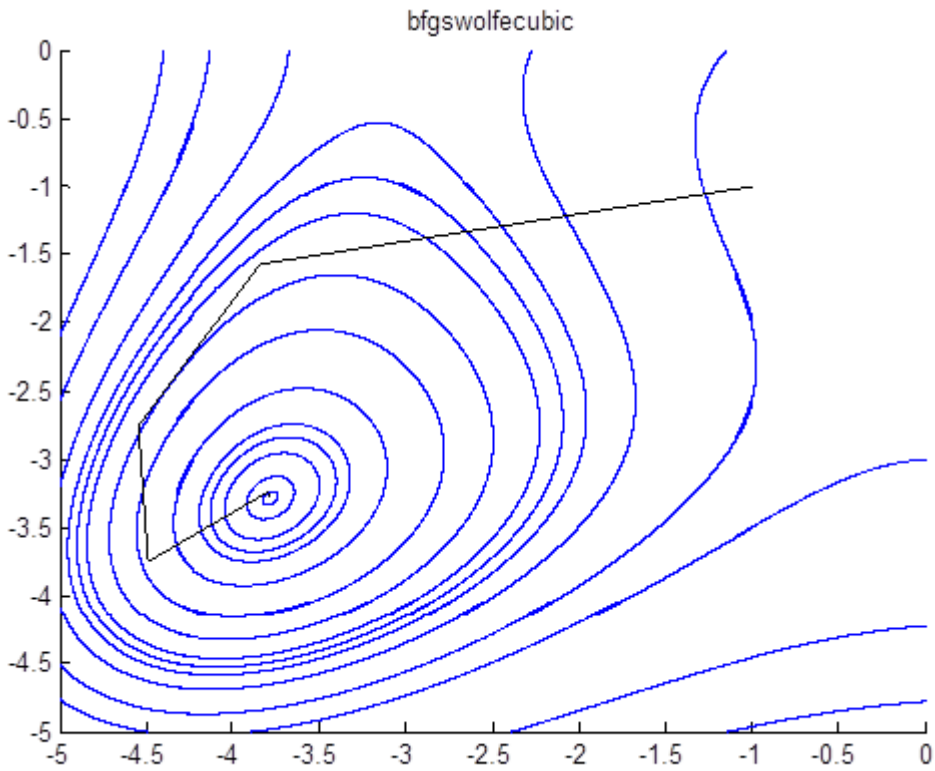
newmethodarmijo με αρχικό σημείο $(-1,-1)$



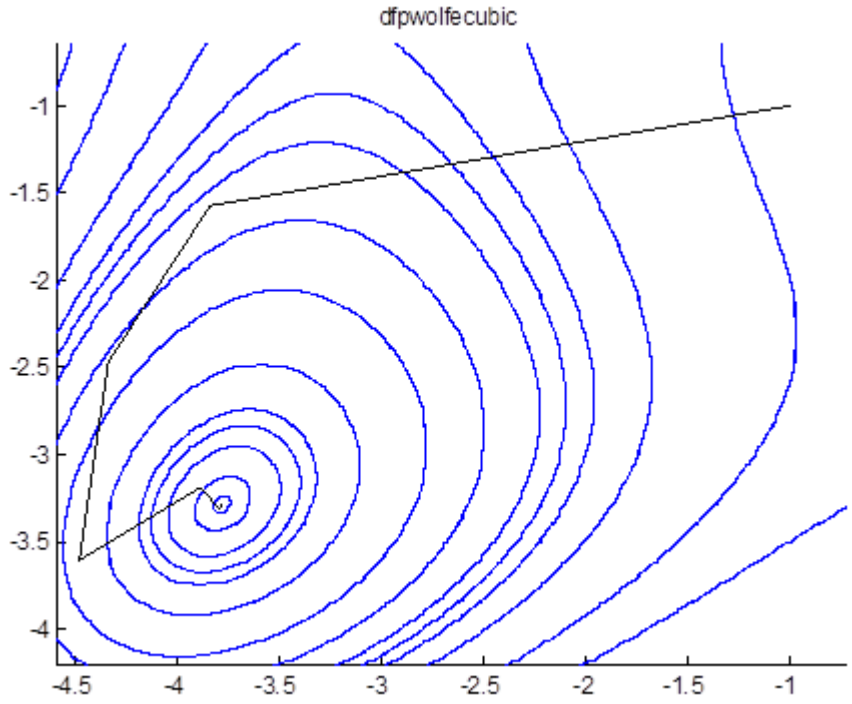
newmethodwolfecubic με αρχικό σημείο $(-1,-1)$



bfgswolfecubic με αρχικό σημείο $(-1,-1)$



dfpwolfecubic με αρχικό σημείο $(-1,-1)$



ΠΑΡΑΔΕΙΓΜΑ 3

HUMP FUNCTION [15] (2 μεταβλητών)

$$f_2(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

$$\nabla f_2(x) = \begin{bmatrix} 8x_1 - 8.4x_1^3 + 2x_1^5 + x_2 \\ x_1 - 8x_2 + 16x_2^3 \end{bmatrix}$$

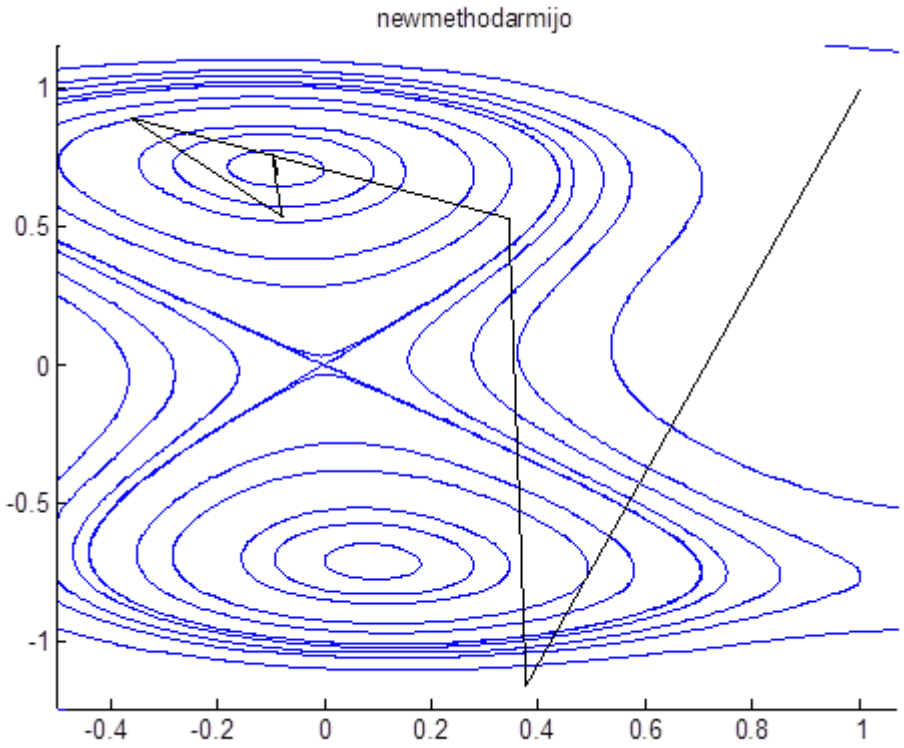
$$epsilon = 1e-10$$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0 = [1,1]$	11(b)	11(a)	20(a)	8(b)	9(b)	9(b)
$x_0 = [-1,-1]$	11(a)	11(b)	20(b)	8(a)	9(a)	9(a)
$x_0 = [1,-3]$	16(b)	21(a)	167(b)	12(a)	9(a)	9(a)
$x_0 = [0,1]$	8(a)	7(b)	7(b)	7(b)	7(b)	7(b)
$x_0 = [-1,0]$	8(a)	7(b)	7(b)	7(b)	7(b)	7(b)

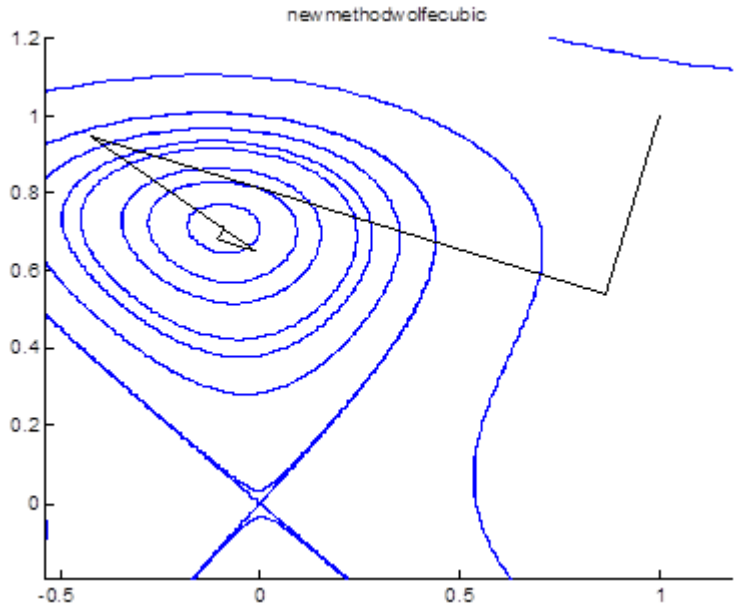
Η συνάρτηση HUMP έχει δύο ελάχιστα, τα οποία είναι ολικά(global). Τα ελάχιστα αυτά είναι στα σημεία $\bar{a} = (0.0898, -0.7126), \bar{b} = (-0.0898, 0.7126)$ και ισχύει $f(\bar{a}) = f(\bar{b}) = 0$. Η παρένθεση δίπλα από τον αριθμό των επαναλήψεων δείχνει σε πιο σημείο από τα δύο συγκλίνει ο εκάστοτε αλγόριθμος. Είναι προφανές οτι για να γίνει σύγκριση μεταξύ μεθόδων θα πρέπει να συγκλίνουν στο ίδιο σημείο.

Παρακάτω παρατίθενται και οι γραφικές παραστάσεις για τις μεθόδους **newmethodarmijo , newmethodwolfecubic , bfgswolfecubic , dfpwolfecubic**

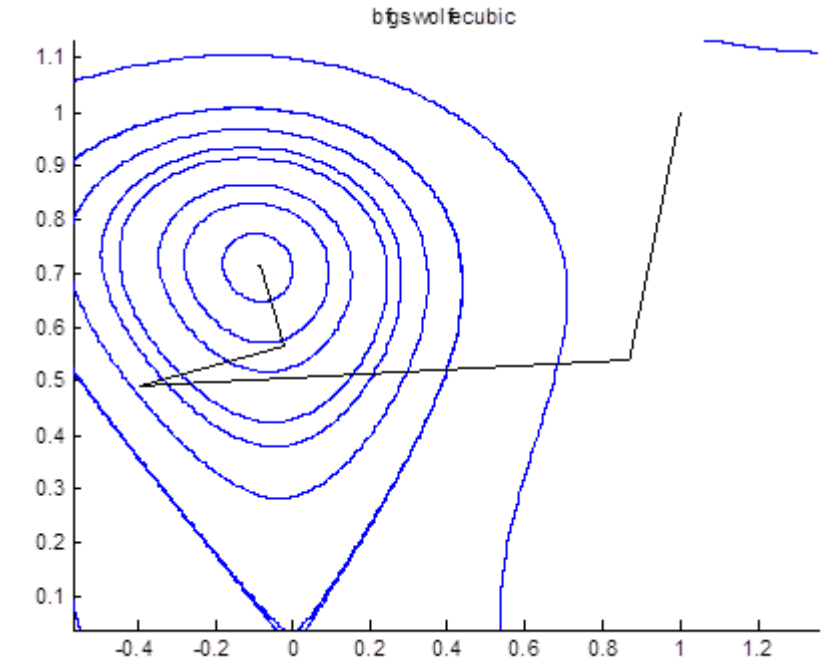
newmethodarmijo με αρχικό σημείο $(1,1)$



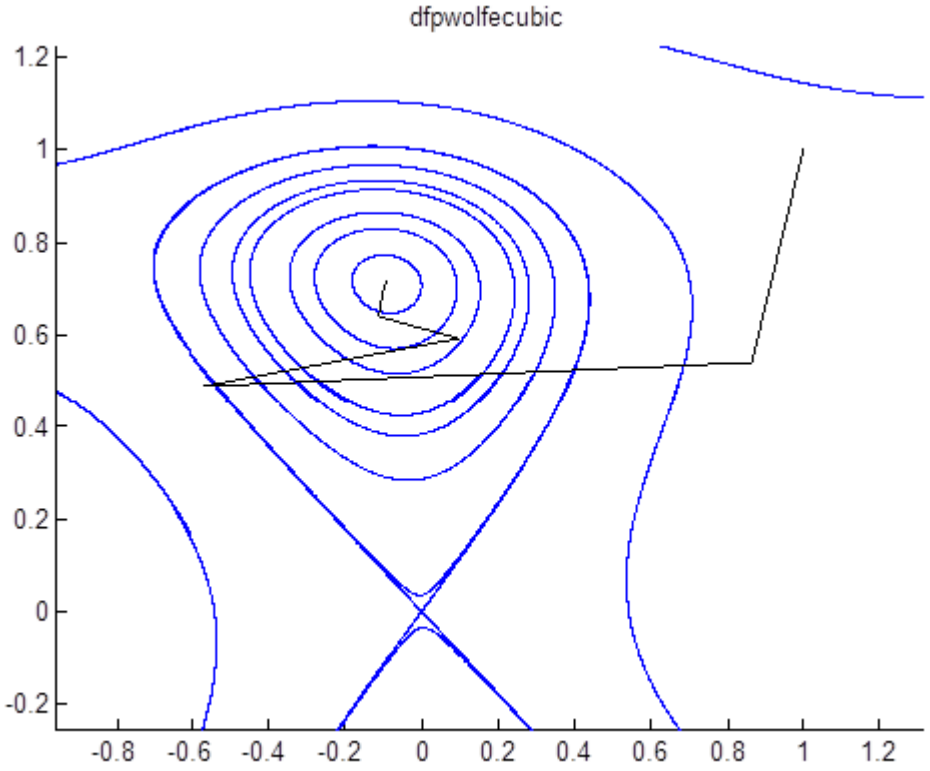
newmethodwolfecubic με αρχικό σημείο $(1,1)$



bfgswolfecubic με αρχικό σημείο $(1,1)$



dfpwolfecubic με αρχικό σημείο $(1,1)$



ΠΑΡΑΔΕΙΓΜΑ 4

BEALE FUNCTION [18] (2 μεταβλητές)

$$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$$

$$\nabla f(x) = \begin{bmatrix} 2(1.5 - x_1 + x_1x_2)(x_2 - 1) + 2(2.5 - x_1 + x_1x_2^2)(x_2^2 - 1) + 2(2.625 - x_1 + x_1x_2^3)(x_2^3 - 1) \\ 2(1.5 - x_1 + x_1x_2)x_1 + 2(2.5 - x_1 + x_1x_2^2)2x_1x_2 + 2(2.625 - x_1 + x_1x_2^3)3x_1x_2^2 \end{bmatrix}$$

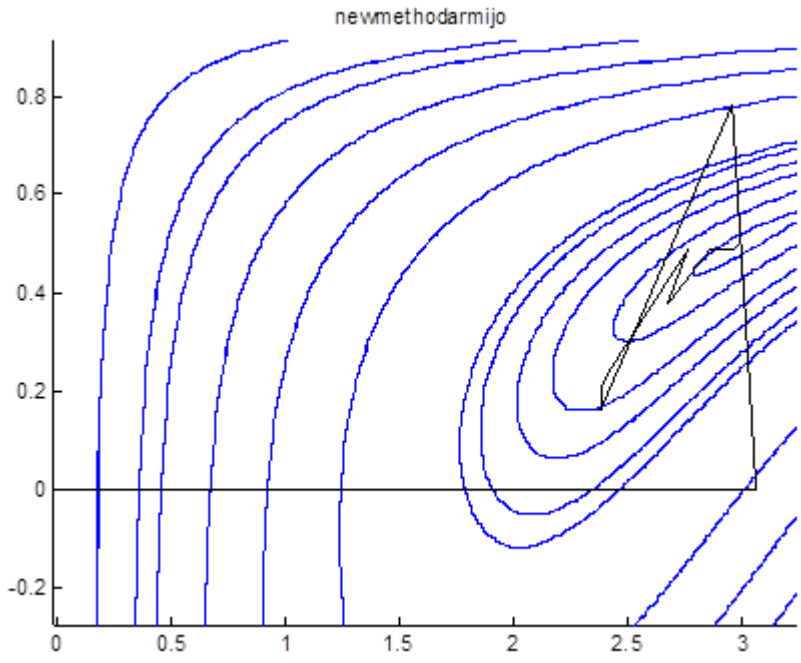
$$epsilon = 1e-10$$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0 = [1,1]$	16	16	19	15	11	14
$x_0 = [-1,4]$	33	-	-	-	-	-
$x_0 = [-2,-2]$	54	22	392	-	21	55
$x_0 = [-2,0]$	19	-	-	20	12	14
$x_0 = [0,0]$	13	13	14	11	9	14
$x_0 = [3,1]$	14	14	180	5	12	21
$x_0 = [2,-2]$	28	22	106	13	22	102
$x_0 = [4.2,1.7]$	19	19	3982	15	15	20
$x_0 = [-2.4,-0.4]$	18	-	-	12	12	16
$x_0 = [4,-2]$	-	24	238	13	18	26
$x_0 = [-1,-2]$	20	20	94	14	16	22
$x_0 = [7,5]$	-	35	-	28	51	-
$x_0 = [10,-3]$	117	-	-	36	45	-

Η συνάρτηση Beale έχει ολικό ελάχιστο στο $x^* = (3,\ 0.5)$, το $f(x^*) = 0$.

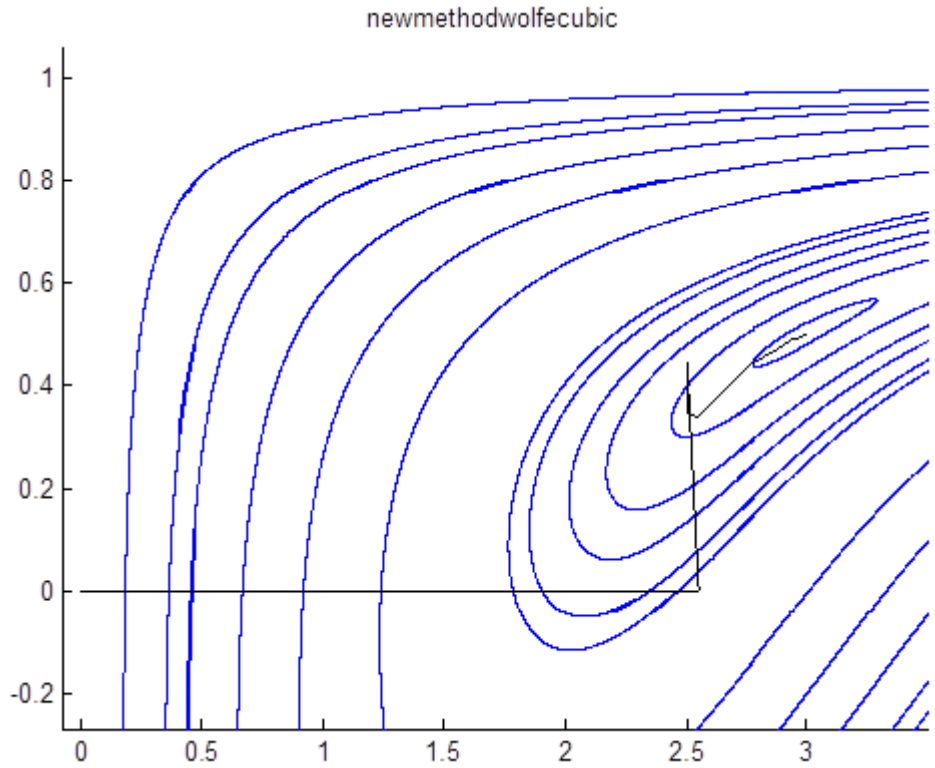
Γραφικές παραστάσεις για τις διάφορες μεθόδους:

newmethodarmijo με αρχικό σημείο $(0,0)$

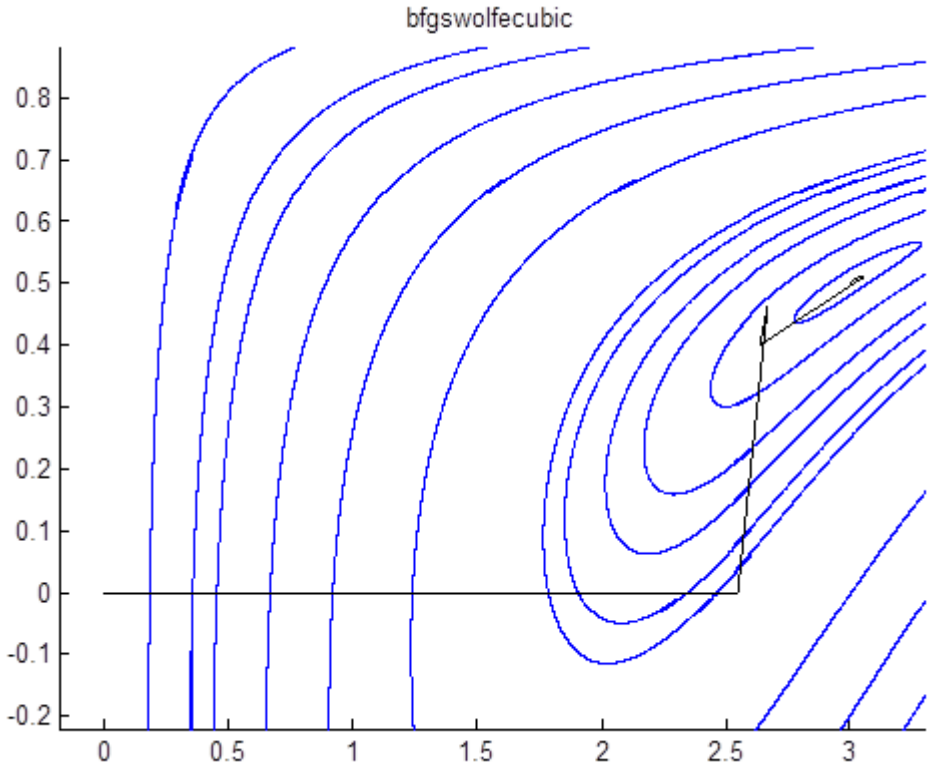


---

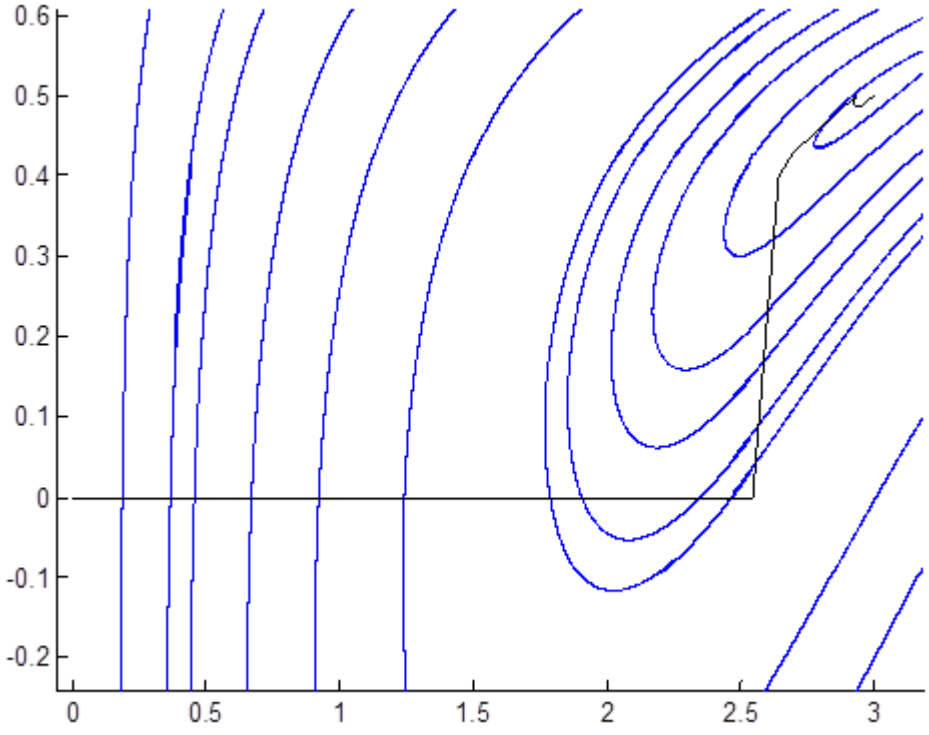
newmethodwolfecubic με αρχικό σημείο $(0,0)$



bfgswolfecubic με αρχικό σημείο $(0,0)$



dfpwolfecubic με αρχικό σημείο $(0,0)$



ΠΑΡΑΔΕΙΓΜΑ 5

FREUDENSTEIN AND ROTH FUNCTION [15] (2 μεταβλητές)

$$f(x)=\left(-13+x_1+((5-x_2)x_2-2)x_2\right)^2+\left(-29+x_1+((x_2+1)x_2-14)x_2\right)^2$$

$$\nabla f(x)=\begin{bmatrix}2(-13+x_1+5x_2^2-x_2^3-2x_2)+2(-29+x_1+x_2^3+x_2^2-14x_2)\\2(-13+x_1+5x_2^2-x_2^3-2x_2)(10x_2-3x_2^2-2)+2(-29+x_1+x_2^3+x_2^2-14x_2)(3x_2^2+2x_2-14)\end{bmatrix}$$

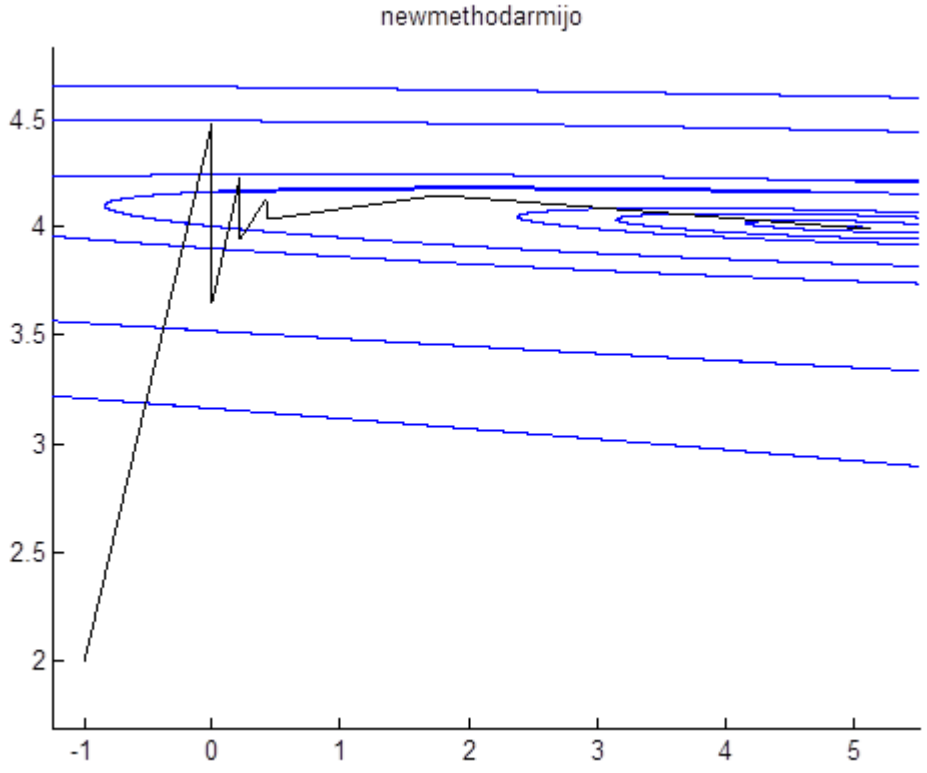
$$epsilon=1e-10$$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0=[-1,4]$	9	12	838	9	6	7
$x_0=[-1,2]$	12	7	8	10	9	8
$x_0=[15,15]$	14	36	-	11	16	21
$x_0=[-12,12]$	16.	33	-	12	14	18
$x_0=[-1,1]$	12.	17.	394.	13.	11.	12.
$x_0=[1,100]$	22	29	-	17	29	37.
$x_0=[20,-20]$	17.	43.	-	14.	22.	34.
$x_0=[-15,-15]$	16	38.	-	17.	20.	29.
$x_0=[-3,-9]$	20.	30.	-	14.	16.	24.
$x_0=[-300,1]$	20.	16.	19.	20.	17.	15.
$x_0=[50,0]$	15.	17.	48.	12.	10.	13.
$x_0=[0,0]$	17	17.	335.	11.	11.	12.
$x_0=[-1,11]$	13	31	-	9	14	17
$x_0=[22,22]$	18.	42	-	13	18	26
$x_0=[7,-25]$	15.	42.	-	15.	22.	34.

Η συνάρτηση έχει ένα γενικό ελάχιστο στο $(5,4)$ το $f(5,4)=0$ και ένα τοπικό ελάχιστο στο $(11.4127....,\ -0.8968....)$ το $f\ (11.4127....,\ -0.8968....)=48.9842$. Η τελεία στο δεξί μέρος των αριθμών δηλώνει οτι η ακολουθία από το συγκεκριμένο σημείο που επιλέξαμε συγκλίνει στο $(11.4127....,\ -0.8968....)$.

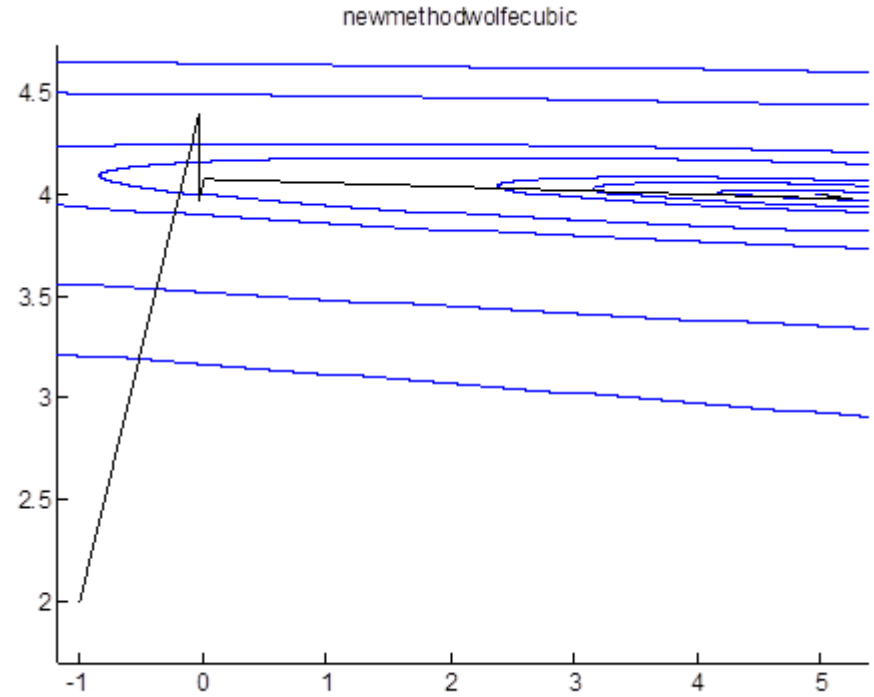
Γραφικές παραστάσεις για τις διάφορες μεθόδους.

newmethodarmijo με αρχικό σημείο $(-1,2)$

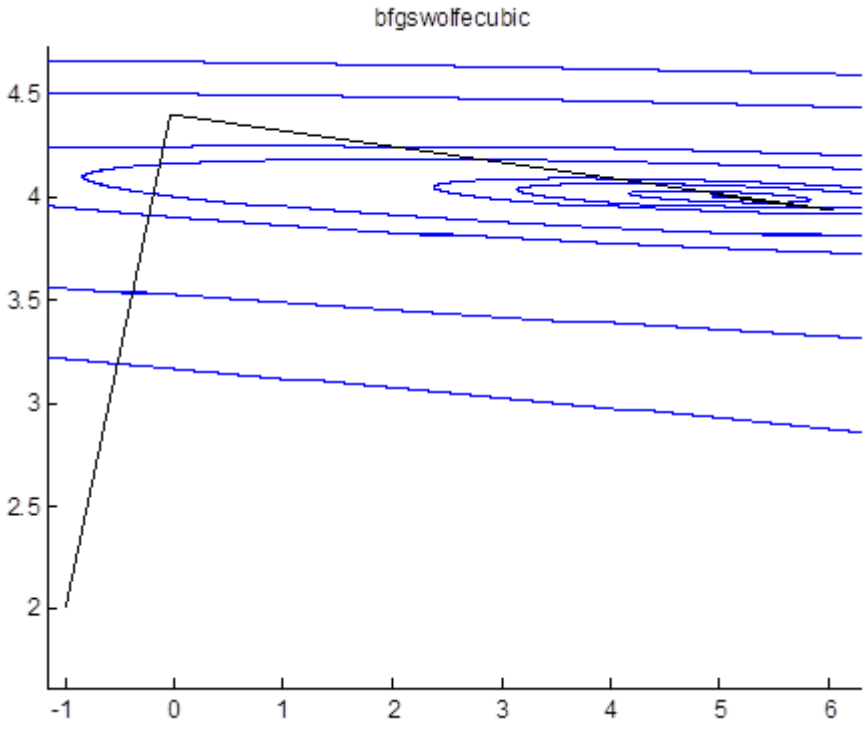


---

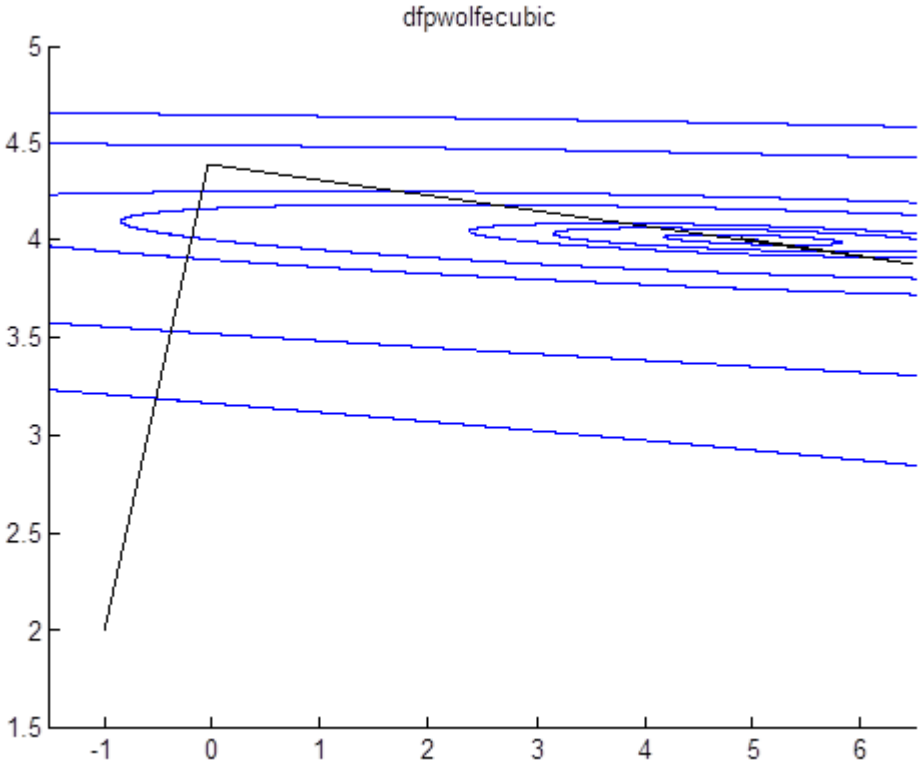
newmethodwolfecubic με αρχικό σημείο $(-1,2)$



bfgswolfecubic με αρχικό σημείο $(-1,2)$



dfpwolfecubic με αρχικό σημείο $(-1,2)$



ΠΑΡΑΔΕΙΓΜΑ 6

EXTENDED ROSENBROCK FUNCTION [18] (4 μεταβλητών)

$$f_4(x)=100\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2+100\left(x_3-x_2^2\right)^2+\left(1-x_2\right)^2+100\left(x_4-x_3^2\right)^2+\left(1-x_3\right)^2$$

$$\nabla f_4(x)=\begin{bmatrix}-400x_1\left(x_2-x_1^2\right)-2\left(1-x_1\right)\\-400x_2\left(x_3-x_2^2\right)-2\left(1-x_2\right)+200\left(x_2-x_1^2\right)\\-400x_3\left(x_4-x_3^2\right)-2\left(1-x_3\right)+200\left(x_3-x_2^2\right)\\200\left(x_4-x_3^2\right)\end{bmatrix}$$

$epsilon=1e-10$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0=[1,2,-1,-1]$	-	45	-	33	23	126
$x_0=[1,2,3,1]$	-	35	-	41	22	860
$x_0=[4,4,3,1]$	55	38	-	42	-	-
$x_0=[2,5,4,1]$	41	42	1634	32	43	-
$x_0=[0,0,1,1]$	77	-	-	53	-	544
$x_0=[2,10,2,1]$	129	49	1660 (κακή ακρίβεια)	30	71	-
$x_0=[2,0,-2,1]$	-	-	-	35	-	-
$x_0=[5,0,30,1]$	89	-	-	82	-	-
$x_0=[1,4,1,2]$	26	41	-	31	33	-
$x_0=[7,3,4,1]$	-	40	-	41	105	277
$x_0=[2,20,8,7]$	260	73	-	58	67	-
$x_0=[4,9,2,0]$	103	-	-	49	79	-

ΠΑΡΑΔΕΙΓΜΑ 7

POWELL FUNCTION [18] (4 μεταβλητές)

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

$$\nabla f(x) = \begin{bmatrix} 2(x_1 + 10x_2) + 40(x_1 - x_4)^3 \\ 20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3 \\ 10(x_3 - x_4) - 8(x_2 - 2x_3)^3 \\ -10(x_3 - x_4) - 40(x_1 - x_4)^3 \end{bmatrix}$$

$$epsilon = 1e - 20$$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0 = [3,-1,0,1]$	39	43	425	40	52	141
$x_0 = [1,1,1,1]$	39	39	501	36	50	149
$x_0 = [5,5,5,5]$	29	90	223	38	50	402
$x_0 = [-1,1,-1,1]$	40	65	1127	36	58	101
$x_0 = [10,10,10,10]$	44	82	464	39	64	105
$x_0 = [-9,1,0,-8]$	32	66	202	39	59	91
$x_0 = [1,2,-1,-2]$	37	89	1157	22	44	94
$x_0 = [0,2,-2,0]$	40	61	3684	24	53	57
$x_0 = [1,0,1,0]$	39	58	236	35	44	59
$x_0 = [-10,-30,-40,10]$	44	107	-	39	94	444
$x_0 = [0,-15,0,15]$	63	81	1483	52	59	69
$x_0 = [0,0,0,15]$	41	93	3946	44	57	192
$x_0 = [-1,-1,-1,-1]$	39	39	501	36	50	149
$x_0 = [22,-2,-2,-9]$	45	100	-	44	48	235

Η συνάρτηση Powell έχει ολικό ελάχιστο(global minimum) στο $(0,0,0,0)$ το $f(0,0,0,0) = 0$

ΠΑΡΑΔΕΙΓΜΑ 8

COLVILLE FUNCTION [15],[17],[18] (4 μεταβλητές)

$$f(x)=100(x_1^2-x_2)^2+(x_1-1)^2+90(x_3^2-x_4)^2+(x_3-1)^2+10.1\left((x_2-1)^2+(x_4-1)^2\right)+19.8(x_2-1)(x_4-1)$$

$$\nabla f(x)=\begin{bmatrix} 400x_1(x_1^2-x_2)+2(x_1-1) \\ -200(x_1^2-x_2)+20.2(x_2-1)+19.8(x_4-1) \\ 360x_3(x_3^2-x_4)+2(x_3-1) \\ -180(x_3^2-x_4)+20.2(x_4-1)+19.8(x_2-1) \end{bmatrix}$$

$epsilon=1e-10$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
$x_0=[2,2,2,2]$	247	30	2680	31	23	114
$[5,-5,5,-5]$	51	95	-	134	35	2730
$[-10,-10,-8,4]$	113	80	-	112	57	72
$[4,1,-1,4]$	98	72	-	60	38	1003
$[0,0,0,0]$	249	27	-	21	17	58
$[-12,-12,-12,-12]$	90	57	-	78	54	109*
$[-1,8,-8,-1]$	93	79	-	115	61	831
$[-9,9,10,-10]$	97	100	-	113	55	61*
$[0,8,0,8]$	81	26	959	26	34	34*
$[-7,-6,12,10]$	125	62	-	54	41	55*
$[30,-30,30,-30]$	239	98	-	65	88	60*
$[-7,-6,12,-10]$	155	60	-	22	39	64*

Η συνάρτηση COLVILLE έχει γενικό ελάχιστο(global minimum) στο σημείο $(1,1,1,1)$ το $f(1,1,1,1)=0$. Το σύμβολο * δίπλα από τον αριθμό των επαναλήψεων δηλώνει οτι οι ο συγκεκριμένος αριθμός επετεύχθη με την εξής τεχνική: κάθε n επαναλήψεις η μήτρα S γινόταν ίση με τη μοναδιαία. Με τρόπο αυτό καταφέραμε βελτίωση της απόδοσης της μεθόδου dfpwolfecubic.

ΠΑΡΑΔΕΙΓΜΑ 9

EXTENDED POWELL FUNCTION [18] (8 μεταβλητές)

$$f(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4+(x_5+10x_6)^2+5(x_7-x_8)^2+(x_6-2x_7)^4+10(x_5-x_8)^4$$

$$\nabla f(x)=\begin{bmatrix} 2(x_1+10x_2)+40(x_1-x_4)^3 \\ 20(x_1+10x_2)+4(x_2-2x_3)^3 \\ 10(x_3-x_4)-8(x_2-2x_3)^3 \\ -10(x_3-x_4)-40(x_1-x_4)^3 \\ 2(x_5+10x_6)+40(x_5-x_8)^3 \\ 20(x_5+10x_6)+4(x_6-2x_7)^3 \\ 10(x_7-x_8)-8(x_6-2x_7)^3 \\ -10(x_7-x_8)-40(x_5-x_8)^3 \end{bmatrix}$$

$epsilon = 1e-06$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[3,−1,0,1, 3,−1,0,1]	76	33	271	55	27	260
[−1,−1,−1,−1, −1,−1,−2,−1]	43	38	1598	70	31	38
[1, 2, 3, 4, 4, 3, 2, 1]	64	57	2994 (περιορισμ. ακρίβεια)	56	30	37
[0,−5,10,−2, 1,−1,−1,2]	44	64	3998 (κακή ακρίβεια)	45	44	127
[10,20,30,−12, −2,1,2,−20]	85	109	-	67	36	943
[−5,−10,6,2, 8,−25,3,0]	52	66	-	55	44	110
[0,1,−1,2, 8,2,31,−20]	81	91	-	62	55	149
[12,10,−9,−12, −30,12,20,−39]	77	119	-	93	64	149
[−1,1,1,1, 1,1,1,1]	51	54	3139	45	34	33
[−1,2,3,−12, 8,5,32,40]	72	92	-	104	64	248
[−2,2,−2,2, 4−5,−6,4]	47	77	2159	54	39	612

Η συνάρτηση Powell έχει ελάχιστο στο $(0,0,0,0,0,0,0,0)$ το $f(0,0,0,0,0,0,0,0)=0$ Το ίδιο ακριβώς ισχύει και στην επέκταση της σε n μεταβλητές ,όπου n πολλαπλάσιο του 4.

ΠΑΡΑΔΕΙΓΜΑ 10

EXTENDED POWELL FUNCTION [18] (12 μεταβλητές)

$$f(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4+(x_5+10x_6)^2+5(x_7-x_8)^2+(x_6-2x_7)^4+10(x_5-x_8)^4+(x_9+10x_{10})^2+5(x_{11}-x_{12})^2+(x_{10}-2x_{11})^4+10(x_9-x_{12})^4$$

$$\nabla f(x)=\begin{bmatrix}2(x_1+10x_2)+40(x_1-x_4)^3\\20(x_1+10x_2)+4(x_2-2x_3)^3\\10(x_3-x_4)-8(x_2-2x_3)^3\\-10(x_3-x_4)-40(x_1-x_4)^3\\2(x_5+10x_6)+40(x_5-x_8)^3\\20(x_5+10x_6)+4(x_6-2x_7)^3\\10(x_7-x_8)-8(x_6-2x_7)^3\\-10(x_7-x_8)-40(x_5-x_8)^3\\2(x_9+10x_{10})+40(x_9-x_{12})^3\\20(x_9+10x_{10})+4(x_{10}-2x_{11})^3\\10(x_{11}-x_{12})-8(x_{10}-2x_{11})^3\\-10(x_{11}-x_{12})-40(x_9-x_{12})^3\end{bmatrix}$$

$epsilon = 1e-10$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[3,−1,0,1,3,−1 ,0,1,3,−1,0,1,]	98	43	434	39	41	69
[1,1,1,1,1,1, 1,1,1,1,1,1]	36	39	246	77	23	78
[−1,0,1,1,0,0, 1,2,3,1,2,−3]	107	154	-	156	91	400
[−4,5,1,2,3,4, 5,6,7,8,9,10]	207	169	-	150	84	148
[−4,5,−1,2,−3,4,−5, 6,−7,8,−9,10]	61	168	-	260	123	-
[10,−12,13,14,15,16,17, 18,−19,−20,21,−22]	95	166	-	212	118	1870
[2,2,2,2,2,2, 2,2,2,2,2,2]	46	47	243	68	34	45

ΠΑΡΑΔΕΙΓΜΑ 11

EXTENDED POWELL FUNCTION [18] (16 μεταβλητές)

$$f(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4+(x_5+10x_6)^2+5(x_7-x_8)^2+(x_6-2x_7)^4+10(x_5-x_8)^4+$$
$$(x_9+10x_{10})^2+5(x_{11}-x_{12})^2+(x_{10}-2x_{11})^4+10(x_9-x_{12})^4+(x_{13}+10x_{14})^2+5(x_{15}-x_{16})^2+(x_{14}-2x_{15})^4+10(x_{13}-x_{16})^4$$

$$\nabla f(x)=\begin{bmatrix}2(x_1+10x_2)+40(x_1-x_4)^3\\20(x_1+10x_2)+4(x_2-2x_3)^3\\10(x_3-x_4)-8(x_2-2x_3)^3\\-10(x_3-x_4)-40(x_1-x_4)^3\\2(x_5+10x_6)+40(x_5-x_8)^3\\20(x_5+10x_6)+4(x_6-2x_7)^3\\10(x_7-x_8)-8(x_6-2x_7)^3\\-10(x_7-x_8)-40(x_5-x_8)^3\\2(x_9+10x_{10})+40(x_9-x_{12})^3\\20(x_9+10x_{10})+4(x_{10}-2x_{11})^3\\10(x_{11}-x_{12})-8(x_{10}-2x_{11})^3\\-10(x_{11}-x_{12})-40(x_9-x_{12})^3\\2(x_{13}+10x_{14})+40(x_{13}-x_{16})^3\\20(x_{13}+10x_{14})+4(x_{14}-2x_{15})^3\\10(x_{15}-x_{16})-8(x_{14}-2x_{15})^3\\-10(x_{15}-x_{16})-40(x_{13}-x_{16})^3\end{bmatrix}$$

$epsilon = 1e-10$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1]	47	39	479	87	27	139
[1,−1,1,−1,1,−1,1,−1, 1,−1,1,−1,1,−1,1,−1]	73	56	1118	35	66	521
[2,4,6,8,10,12,14,16, 18,20,2,2,−2,−2,−2,−2]	157	254	-	233	134	-
[5,4,3,2,1,−1,−2,−3, −4,−5,1,2,2,1,−1,−1]	292	226	-	189	113	722
[1,1,1,−10,−2,−3,−4,−5, 1,2,2,1,−1,−1,15,16]	197	241	-	224	168	1065
[4,3,2,−1,−2,−3,4,5,1, 2,3,−1,−10,−20,−12,−14]	263	216	-	141	120	-
[−20,1,2,3,−1,−2,−3,−4, −5,−6,−7,−8,−9,−10,1,10]	185	251	-	135	135	-
[3,3,1,2,3,1,2,−1,−3, −1,2,−3,2,−1,2,−3]	210	190	-	291	105	1289
[−10,10,2,8,23,6,3,2, 43,0,21,2,3,21,−12,20]	248	243	-	434	154	-

ΠΑΡΑΔΕΙΓΜΑ 12

EXTENDED POWELL FUNCTION [18] (20 μεταβλητές)

$$f(x)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4+(x_5+10x_6)^2+5(x_7-x_8)^2+(x_6-2x_7)^4+10(x_5-x_8)^4+$$
$$(x_9+10x_{10})^2+5(x_{11}-x_{12})^2+(x_{10}-2x_{11})^4+10(x_9-x_{12})^4+(x_{13}+10x_{14})^2+5(x_{15}-x_{16})^2+(x_{14}-2x_{15})^4+10(x_{13}-x_{16})^4+$$
$$(x_{17}+10x_{18})^2+5(x_{19}-x_{20})^2+(x_{18}-2x_{19})^4+10(x_{17}-x_{20})^4$$

$$\nabla f(x)=\begin{bmatrix}
2(x_1+10x_2)+40(x_1-x_4)^3\\
20(x_1+10x_2)+4(x_2-2x_3)^3\\
10(x_3-x_4)-8(x_2-2x_3)^3\\
-10(x_3-x_4)-40(x_1-x_4)^3\\
2(x_5+10x_6)+40(x_5-x_8)^3\\
20(x_5+10x_6)+4(x_6-2x_7)^3\\
10(x_7-x_8)-8(x_6-2x_7)^3\\
-10(x_7-x_8)-40(x_5-x_8)^3\\
2(x_9+10x_{10})+40(x_9-x_{12})^3\\
20(x_9+10x_{10})+4(x_{10}-2x_{11})^3\\
10(x_{11}-x_{12})-8(x_{10}-2x_{11})^3\\
-10(x_{11}-x_{12})-40(x_9-x_{12})^3\\
2(x_{13}+10x_{14})+40(x_{13}-x_{16})^3\\
20(x_{13}+10x_{14})+4(x_{14}-2x_{15})^3\\
10(x_{15}-x_{16})-8(x_{14}-2x_{15})^3\\
-10(x_{15}-x_{16})-40(x_{13}-x_{16})^3\\
2(x_{17}+10x_{18})+40(x_{17}-x_{20})^3\\
20(x_{17}+10x_{18})+4(x_{18}-2x_{19})^3\\
10(x_{19}-x_{20})-8(x_{18}-2x_{19})^3\\
-10(x_{19}-x_{20})-40(x_{17}-x_{20})^3
\end{bmatrix}$$

$epsilon = 1e-10$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1]	75	39	377	181	30	434
[2,−1,2,−1,2,−1,2,−1,2,−1, 2,−1,2,−1,2,−1,2,−1,2,−1,]	60	66	2245	144	33	342
[1,2,3,4,5,6,7,8,9,1, 2,3,4,5,6,7,8,9,1,2]	133	257	-	457	127	-
[−1,−2,−3,−4,−5,−6,−7,−8,−9, −9,−8,−7,−6,−5,−4,−3,−2,−1,0,1]	216	253	-	471	126	-
[2,4,6,8,8,6,4,2,−2,−4,−6, −8,−8,−6,−4,−2,1,2,3,−2]	145	165	-	324	126	-

ΠΑΡΑΔΕΙΓΜΑ 13

EXTENDED ROSENBROCK FUNCTION [18] (8 μεταβλητών)

$$f_8(x)=100\left(x_2-x_1^2\right)^2+(1-x_1)^2+100\left(x_3-x_2^2\right)^2+(1-x_2)^2+100\left(x_4-x_3^2\right)^2+(1-x_3)^2+$$
$$100\left(x_5-x_4^2\right)^2+(1-x_4)^2+100\left(x_6-x_5^2\right)^2+(1-x_5)^2+100\left(x_7-x_6^2\right)^2+(1-x_6)^2+100\left(x_8-x_7^2\right)^2+(1-x_7)^2$$

$$\nabla f_8(x)=\begin{bmatrix}-400x_1\left(x_2-x_1^2\right)-2(1-x_1)\\ -400x_2\left(x_3-x_2^2\right)-2(1-x_2)+200\left(x_2-x_1^2\right)\\ -400x_3\left(x_4-x_3^2\right)-2(1-x_3)+200\left(x_3-x_2^2\right)\\ -400x_4\left(x_5-x_4^2\right)-2(1-x_4)+200\left(x_4-x_3^2\right)\\ -400x_5\left(x_6-x_5^2\right)-2(1-x_5)+200\left(x_5-x_4^2\right)\\ -400x_6\left(x_7-x_6^2\right)-2(1-x_6)+200\left(x_6-x_5^2\right)\\ -400x_7\left(x_8-x_7^2\right)-2(1-x_7)+200\left(x_7-x_6^2\right)\\ 200\left(x_8-x_7^2\right)\end{bmatrix}$$

$epsilon=1e-07$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[1,2,3,1,2,3,1,2]	-	51	-	45	29	227
[1,1,1,2,2,2,3,3]	107	51	-	61	31	256
[0,1,0,1,0,1,0,1]	-	-	-	118	30	44
[1.2,2,1,2,3,4,2,2]	90	59	3393	62	47	571
[3,1,1,3,3,1,2,4]	197	54	713(κακή ακρίβεια)	43	40	327

Η συνάρτηση Rosenbrock έχει ελάχιστο στο $(1,1,1,1,1,1,1,1)$ το $f(1,1,1,1,1,1,1,1)=0$ Το ίδιο ακριβώς ισχύει και στην επέκταση της σε n μεταβλητές .

ΠΑΡΑΔΕΙΓΜΑ 14

EXTENDED ROSENBROCK FUNCTION [18] (11 μεταβλητών)

$$f_{11}(x)=100\left(x_2-x_1^2\right)^2+(1-x_1)^2+100\left(x_3-x_2^2\right)^2+(1-x_2)^2+100\left(x_4-x_3^2\right)^2+(1-x_3)^2+$$
$$100\left(x_5-x_4^2\right)^2+(1-x_4)^2+100\left(x_6-x_5^2\right)^2+(1-x_5)^2+100\left(x_7-x_6^2\right)^2+(1-x_6)^2+$$
$$100\left(x_8-x_7^2\right)^2+(1-x_7)^2+100\left(x_9-x_8^2\right)^2+(1-x_8)^2+100\left(x_{10}-x_9^2\right)^2+(1-x_9)^2+100\left(x_{11}-x_{10}^2\right)^2+(1-x_{10})^2$$

$$\nabla f_{11}(x)=\begin{bmatrix}-400x_1\left(x_2-x_1^2\right)-2(1-x_1)\\-400x_2\left(x_3-x_2^2\right)-2(1-x_2)+200\left(x_2-x_1^2\right)\\-400x_3\left(x_4-x_3^2\right)-2(1-x_3)+200\left(x_3-x_2^2\right)\\-400x_4\left(x_5-x_4^2\right)-2(1-x_4)+200\left(x_4-x_3^2\right)\\-400x_5\left(x_6-x_5^2\right)-2(1-x_5)+200\left(x_5-x_4^2\right)\\-400x_6\left(x_7-x_6^2\right)-2(1-x_6)+200\left(x_6-x_5^2\right)\\-400x_7\left(x_8-x_7^2\right)-2(1-x_7)+200\left(x_7-x_6^2\right)\\-400x_8\left(x_9-x_8^2\right)-2(1-x_8)+200\left(x_8-x_7^2\right)\\-400x_9\left(x_{10}-x_9^2\right)-2(1-x_9)+200\left(x_9-x_8^2\right)\\-400x_{10}\left(x_{11}-x_{10}^2\right)-2(1-x_{10})+200\left(x_{10}-x_9^2\right)\\200\left(x_{11}-x_{10}^2\right)\end{bmatrix}$$

$epsilon=1e-07$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[1,2,1,2,1, 2,1,2,1,1,1]	-	38	3999	53	41	67
[3,4,1,1,2,3, 1,1,2,2,2]	111	71	-	73	50	601
[1.5,1,2,3,2, 1,2,3,1,0,1]	223	61	-	80	32	283
[0.6,2,1,3,2, 1,1,1,1,2,2]	92	60	-	45	34	69
[0.5,1.2,1,2,3, 2,1,0.7,1.4,2,1]	33	56	-	30	39	57

ΠΑΡΑΔΕΙΓΜΑ 15

EXTENDED ROSENBROCK FUNCTION [18] (20 μεταβλητών)

$$f_{20}(x)=100\left(x_2-x_1^2\right)^2+\left(1-x_1\right)^2+100\left(x_3-x_2^2\right)^2+\left(1-x_2\right)^2+100\left(x_4-x_3^2\right)^2+\left(1-x_3\right)^2$$
$$100\left(x_5-x_4^2\right)^2+\left(1-x_4\right)^2+100\left(x_6-x_5^2\right)^2+\left(1-x_5\right)^2+100\left(x_7-x_6^2\right)^2+\left(1-x_6\right)^2+$$
$$100\left(x_8-x_7^2\right)^2+\left(1-x_7\right)^2+100\left(x_9-x_8^2\right)^2+\left(1-x_8\right)^2+100\left(x_{10}-x_9^2\right)^2+\left(1-x_9\right)^2+$$
$$100\left(x_{11}-x_{10}^2\right)^2+\left(1-x_{10}\right)^2+100\left(x_{12}-x_{11}^2\right)^2+\left(1-x_{11}\right)^2+$$
$$100\left(x_{13}-x_{12}^2\right)^2+\left(1-x_{12}\right)^2+100\left(x_{14}-x_{13}^2\right)^2+\left(1-x_{13}\right)^2+100\left(x_{15}-x_{14}^2\right)^2+\left(1-x_{14}\right)^2+$$
$$100\left(x_{16}-x_{15}^2\right)^2+\left(1-x_{15}\right)^2+100\left(x_{17}-x_{16}^2\right)^2+\left(1-x_{16}\right)^2+100\left(x_{18}-x_{17}^2\right)^2+\left(1-x_{17}\right)^2+$$
$$100\left(x_{19}-x_{18}^2\right)^2+\left(1-x_{18}\right)^2+100\left(x_{20}-x_{19}^2\right)^2+\left(1-x_{19}\right)^2$$

$$\nabla f_{20}(x)=\begin{bmatrix}-400x_1\left(x_2-x_1^2\right)-2\left(1-x_1\right)\\-400x_2\left(x_3-x_2^2\right)-2\left(1-x_2\right)+200\left(x_2-x_1^2\right)\\-400x_3\left(x_4-x_3^2\right)-2\left(1-x_3\right)+200\left(x_3-x_2^2\right)\\-400x_4\left(x_5-x_4^2\right)-2\left(1-x_4\right)+200\left(x_4-x_3^2\right)\\-400x_5\left(x_6-x_5^2\right)-2\left(1-x_5\right)+200\left(x_5-x_4^2\right)\\-400x_6\left(x_7-x_6^2\right)-2\left(1-x_6\right)+200\left(x_6-x_5^2\right)\\-400x_7\left(x_8-x_7^2\right)-2\left(1-x_7\right)+200\left(x_7-x_6^2\right)\\-400x_8\left(x_9-x_8^2\right)-2\left(1-x_8\right)+200\left(x_8-x_7^2\right)\\-400x_9\left(x_{10}-x_9^2\right)-2\left(1-x_9\right)+200\left(x_9-x_8^2\right)\\\text{.....}\\\text{.....}\\200\left(x_{20}-x_{19}^2\right)\end{bmatrix}$$

$epsilon=1e-07$

Αρχικό Σημείο x_0	NEWMETHOD ARMIJO	BFGS MATLAB	DFP MATLAB	NEWMETHOD WOLFECUBIC	BFGS WOLFECUBIC	DFP WOLFECUBIC
[1.5,1,2,3,2,1,2,3,1,0, 1,2,1,2,1,2,2,1,2,1]	171	83	-	83	68	241
[3,2,1,1,1,1,2,3,4,1, 3,2,1,1,2,1,2,3,1,2]	66	83	-	69	58	337
[4,3,2,1,1,1,2,3,4,4, 5,4,3,2,1,1,2,3,4,5]	-	122	-	167	75	1406
[0,1,2,3,1,2,3,0,3,2, 1,2,3,2,1,0,1,2,1,2]	750	90	-	58	53	205
[3,2,2,3,1,1,2,2,2,2, 5,4,3,2,1,3,3,3,3,3]	453	125	-	110	85	983
[4,4,4,4,3,2,1,1,2,3, 4,5,4,3,2,1,1,1,1,1]	-	69	239	39	69	-

ΚΕΦΑΛΑΙΟ 8

Συμπεράσματα

Η νέα μέθοδος που προτάθηκε και προγραμματίστηκε συνιστά μια μέθοδο $quasi-newton$ αφού ικανοποιεί τη χαρακτηριστική εξίσωση που ικανοποιούν όλες οι μέθοδοι της κατηγορίας, αλλά αυτό που την διαφοροποιεί σε σχέση με τις άλλες κλασσικές μεθόδους $quasi-newton$ BFGS και DFP είναι η ιδιότητα της να ενσωματώνει άμεσα σε κάθε επανάληψη πληροφορία από τις n προηγούμενες επαναλήψεις, όπου n ο αριθμός των μεταβλητών της υπό ελαχιστοποίηση συνάρτησης ενώ οι μέθοδοι BFGS και DFP σε κάθε επανάληψη τους χρησιμοποιούν πληροφορία-δεδομένα μόνο της προηγούμενης επανάληψης.

Έγινε αριθμητική υλοποίηση της μεθόδου με δύο διαφορετικούς τρόπους , δηλαδή προγραμματίστηκε η μέθοδος πρώτα με κανόνα Armijo και στη συνέχεια με συνδυασμό κανόνα Wolfe και κυβικής παρεμβολής για την εύρεση του μήκους βήματος. Επιπλέον, υλοποιήθηκαν οι μέθοδοι BFGS και DFP με χρήση κανόνα Wolfe και κυβικής παρεμβολής. Πραγματοποιήθηκε σύγκριση μεταξύ των μεθόδων σε πολλά παραδείγματα που αντλήθηκαν από την βιβλιογραφία με τέτοιο τρόπο ώστε να έχουμε ένα μεγάλο εύρος αριθμού μεταβλητών, από δύο έως είκοσι.

Ενδιαφέρουσα είναι η επίδραση της παραμέτρου d στις επαναλήψεις που απαιτούνται για τη σύγκλιση της μεθόδου μας. Παρατηρείται ότι η επιλογή μεγάλων τιμών για την παράμετρο d έχει σαν αποτέλεσμα την αύξηση του αριθμού των επαναλήψεων για την σύγκλιση της, αφού γίνεται χρήση κυρίως της κατεύθυνσης της κλίσης αντί της quasi-newton κατεύθυνσης.Ιδανική τιμή για την παράμετρο d φαίνεται να είναι η τιμή d=0.00001.Τιμές μικρότερες από αυτές δείχνουν να έχουν μηδενική επίδραση στα αποτελέσματα.

Είναι εμφανές από τα αριθμητικά αποτελέσματα των παραδειγμάτων ότι η νέα μέθοδος είναι αξιόπιστη, αποτελεσματική και βρίσκει ελάχιστα χωρίς προβλήματα με λίγες περιπτώσεις στις οποίες αδυνατεί να τα βρει ενώ έχει υπερβεί ένα όριο επαναλήψεων. Σε σύγκριση με τις μεθόδους BFGS και DFP παρατηρούμε ότι η προτεινόμενη μέθοδος είναι πολύ καλύτερη από τη DFP ενώ σε σχέση με την BFGS , η δεύτερη επιτυγχάνει σύγκλιση σε ίδιες ή και λιγότερες επαναλήψεις. Ανακεφαλαιώνοντας μπορούμε να πούμε οτι συνιστά μια αξιόπιστη μέθοδο με αριθμητικές επιδόσεις ανάμεσα στην DFP και την BFGS.

Αξίζει τέλος να αναφερθεί οτι οι δικές μας υλοποιήσεις των αλγορίθμων BFGS και DFP είναι αισθητά καλύτερες από του matlab, ειδικά όσον αφορά την μέθοδο DFP , της οποίας ο έτοιμος κώδικας στο matlab δείχνει αρκετά αναποτελεσματικός. Επιπλέον ο κώδικας τους είναι ιδιαίτερα πολύπλοκος, δυσνόητος και δύσκολα τροποποιήσιμος, κάτι το οποίο καθίστα αδύνατη τη μεταβολή των παραμέτρων του κώδικα για την επίλυση διαφορετικών προβλημάτων.

Παράρτημα

Κώδικας MATLAB και Επεξηγήσεις

ΑΛΓΟΡΙΘΜΟΣ DFP MATLAB

Ο παρακάτω κώδικας υλοποιεί τον αλγόριθμο DFP χρησιμοποιώντας την ενσωματωμένη ρουτίνα fminunc του $MATLAB$, η οποία προσπαθεί να βρει ένα ελάχιστο μια βαθμωτής συνάρτησης n μεταβλητών, ξεκινώντας από μια αρχική εκτίμηση.Στον παρακάτω κώδικα ορίζουμε ως f τη συνάρτηση που επιθυμούμε να ελαχιστοποιήσουμε και ως g τη κλίση ($g = \nabla f$) της συνάρτησης f. Αξίζει να αναφερθεί ότι η συνάρτηση fminunc του $MATLAB$ χρησιμοποιεί ένα συνδυασμό τετραγωνικής(quadratic interpolation) και κυβικής προσαρμογής(cubic interpolation) για την εύρεση του μήκους βήματος. Η συνάρτηση του $ROSENBROCK$ έχει χρησιμοποιηθεί στο παρακάτω παράδειγμα.

```
function trialDFP
x0=[1;4];
options=optimset('GradObj','on','LargeScale','off','HessUpda
te','dfp','MaxFunEvals',1500,'MaxIter',1000);
[x,fval,exitflag,output]=fminunc(@fgrad,x0,options)
function [f,g]=fgrad(x)
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
g=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
```

ΑΛΓΟΡΙΘΜΟΣ BFGS MATLAB

Ο παρακάτω κώδικας υλοποιεί τον αλγόριθμο BFGS χρησιμοποιώντας την ενσωματωμένη ρουτίνα fminunc του $MATLAB$, η οποία προσπαθεί να βρει ένα ελάχιστο μια βαθμωτής συνάρτησης n μεταβλητών, ξεκινώντας από μια αρχική εκτίμηση. Στον παρακάτω κώδικα ορίζουμε ως f τη συνάρτηση που επιθυμούμε να ελαχιστοποιήσουμε και ως g τη κλίση ($g = \nabla f$) της συνάρτησης f. Η συνάρτηση του $ROSENBROCK$ έχει χρησιμοποιηθεί στο παρακάτω παράδειγμα.

```
function trialBFGS
x0=[1;4];
options=optimset('GradObj','on','LargeScale','off');
[x,fval,exitflag,output]=fminunc(@fgrad,x0,options)
function [f,g]=fgrad(x)
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
g=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
```

ΠΡΟΤΕΙΝΟΜΕΝΟΣ QUASI-NEWTON ΑΛΓΟΡΙΘΜΟΣ

ΜΕ ΧΡΗΣΗ ΚΑΝΟΝΑ ARMIJO

Ο παρακάτω κώδικας υλοποιεί τον προτεινόμενο quasi-newton αλγόριθμο και αποτελείται από δύο υποπρογράμματα: το 1^{ο} υποπρόγραμμα έχοντας ως δεδομένα το αρχικό σημείο $\overline{x}$, και τις απαραίτητες παραμέτρους – μήτρες , καλεί το βασικό υποπρόγραμμα(quasi) με ορίσματα τις παραμέτρους αυτές , το οποίο υλοποιεί τον αλγόριθμο. Το υποπρόγραμμα quasi για την εύρεση του μήκους βήματος λ_i χρησιμοποιεί τον κανόνα $Armijo$.

```
function [out]=trial
x=[-5;10];
Q=eye(2,2);
B=eye(2,2);
alpha=0.3;
beta=0.7;
d=0.00001;
epsilon=1e-6;
[out]=quasi(x,Q,B,d,alpha,beta,epsilon)
```

```
function [out]=quasi(x,Q,B,d,alpha,beta,epsilon)
% x must be a column vector
%nn=size(x);
%n=nn(1);
fold=f(x);
grf=gradf(x);
grm=grf'*grf;
k=0;
out=[k x']
while grm>epsilon
%  STEP 1
    s=Q\grf;
    h=B*s;
    fbs=grf'*h;
    if fbs<d*grm; h=grf; end
    h=-h;
    fbs=grf'*h;
%  STEP 2
    ita=0;
    lamda=1;
    fnew=f(x+h);
    delta=fnew-fold+alpha*d*grm;
    while delta>0
        lamda=lamda*beta;
        fnew=f(x+lamda*h);
        delta=fnew-fold+alpha*d*lamda*grm;
    end
%  STEP 3
```

```
    xnew=x+lamda*h;
%  STEP 4
    p=xnew-x;
    grfnew=gradf(xnew);
    dir=grfnew'*h;
    q=grfnew-grf;
    i=round(mod(k,n)+1);
    BB=B(1:n,1:(i-1));
    BBB=B(1:n,(i+1):n);
    B=[BB p BBB];
    QQ=Q(1:n,1:(i-1));
    QQQ=Q(1:n,(i+1):n);
    Q=[QQ q QQQ];
    k=k+1;
    x=xnew;
    grf=grfnew;
    grm=grf'*grf;
    fold=fnew;
    [k fbs dir x']
    out=[out;[k x']];
end
function [r]=f(x)
r=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
function [r]=gradf(x)
r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
```

ΠΡΟΤΕΙΝΟΜΕΝΟΣ QUASI-NEWTON ΑΛΓΟΡΙΘΜΟΣ ΜΕ ΧΡΗΣΗ ΚΑΝΟΝΑ WOLFE ΚΑΙ ΚΥΒΙΚΗΣ ΠΡΟΣΑΡΜΟΓΗΣ

Ο παρακάτω κώδικας υλοποιεί τον προτεινόμενο quasi-newton αλγόριθμο και αποτελείται από δύο υποπρογράμματα ακριβώς όπως και το προηγούμενο πρόγραμμα. Η εύρεση του μήκους βήματος λ_i γίνεται με συνδυασμό κανόνα Wolfe(Wolfe's rule) και κυβικής παρεμβολής(cubic interpolation) με τον τρόπο που περιγράψαμε. Υλοποιώντας στο matlab με τον ίδιο τρόπο την εύρεση του μήκους βήματος για τις μεθόδους BFGS και DFP μπορούμε να αποφανθούμε για την αποτελεσματικότητα του προτεινόμενου αλγορίθμου συγκριτικά με τους άλλους, επί ίσοις όροις.

```
function [out]=trialWC
x=[-3;-1;1;1];
Q=eye(4,4);
B=eye(4,4);
alpha=0.3;
beta=0.7;
d=0.00001;
epsilon=1e-6;
[out]=quasiWC(x,Q,B,d,alpha,beta,epsilon)
```

```
function [out]=quasiWC(x,Q,B,d,alpha,beta,epsilon)
% x must be a column vector
nn=size(x);
n=nn(1);
eps1=1e-4;
aa=1.7;
theta=0.2;
fold=f(x);
grf=gradf(x);
grm=grf'*grf;
k=0;
out=[k x']
while grm>epsilon
%  STEP 1
s=Q\grf;
    h=B*s;
    fbs=grf'*h;
    if fbs<d*grm; h=grf; end
    h=-h;
    fbs=grf'*h;
%  STEP 2
%  Lemarechal, Wolf, cubic
    lamda0=0; f0=fold; fbs0=fbs;
    lamdaL=0; fL=fold; fbsL=fbs;
    lamdaR=0; fR=fold; fbsR=fbs;
    lamdaold=0; fbsold=fbs;
    lamda=1;
    xnew=x+lamda*h;
    fnew=f(xnew);
    grfnew=gradf(xnew);
    fbsnew=grfnew'*h;
    delta1=fnew-f0-alpha*lamda*fbs0;
    delta2=fbsnew-beta*fbs0;
    while ((lamdaR-lamdaL)>=eps1)||(lamdaR==0)
        if delta1<=0
            if delta2>=0
                break;
            else
                lamdaL=lamda; fL=fnew; fbsL=fbsnew;
            end
        else
            lamdaR=lamda; fR=fnew; fbsR=fbsnew;
        end
        pp=fbsnew+fbsold-3*(fnew-fold)/(lamda-lamdaold);
        DD=pp*pp-fbsnew*fbsold;
        DD=max(0,DD);
        dd=sqrt(DD);
        rr=(fbsnew-pp+dd)/(fbsnew-fbsold+2*dd);
        lamdanew=lamda+rr*(lamdaold-lamda);
        if lamdaR==0
            lamdanew=max(lamdanew,aa*lamda);
```

```
        else
            lamdanew=min(lamdanew,lamdaR-theta*(lamdaR-lamdaL));
            lamdanew=max(lamdanew,lamdaL+theta*(lamdaR-lamdaL));
        end
        lamdaold=lamda; fold=fnew; fbsold=fbsnew;
        lamda=lamdanew;
        xnew=x+lamda*h;
        fnew=f(xnew);
        grfnew=gradf(xnew);
        fbsnew=grfnew'*h;
        delta1=fnew-f0-alpha*lamda*fbs0;
        delta2=fbsnew-beta*fbs0;
    end
%  STEP 3
    xnew=x+lamda*h;
%  STEP 4
    p=xnew-x;
    grfnew=gradf(xnew);
    dir=grfnew'*h;
    q=grfnew-grf;
    i=round(mod(k,n)+1);
    BB=B(1:n,1:(i-1));
    BBB=B(1:n,(i+1):n);
    B=[BB p BBB];
    help=Q;
    QQ=Q(1:n,1:(i-1));
    QQQ=Q(1:n,(i+1):n);
    Q=[QQ q QQQ];
    if det(Q)==0; Q=eye(4,4); end
    k=k+1;
    if k>5000, break; end;
    x=xnew;
    grf=grfnew;
    grm=grf'*grf;
    fold=fnew;
    [k fbsold h' x']
    out=[out;[k x']];
end
function [r]=f(x)
%r=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
 r=100*(x(2)-x(1)^2)^2+(1-x(1))^2+100*(x(4)-x(3)^2)^2+(1-
x(3))^2+10*(x(2)+x(4)-2)^2;

function [r]=gradf(x)
% r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
  r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-
x(1)^2)+20*(x(2)+x(4)-2);-400*x(3)*(x(4)-x(3)^2)-2*(1-
x(3));200*(x(4)-x(3)^2)+20*(x(2)+x(4)-2)];
```

ΑΛΓΟΡΙΘΜΟΣ BFGS ΜΕ ΧΡΗΣΗ ΚΑΝΟΝΑ WOLFE ΚΑΙ ΚΥΒΙΚΗΣ ΠΡΟΣΑΡΜΟΓΗΣ

Ο παρακάτω κώδικας υλοποιεί τον αλγόριθμο BFGS και αποτελείται από δύο υποπρογράμματα. Η εύρεση του μήκους βήματος λ_i γίνεται με συνδυασμό κανόνα Wolfe(Wolfe's rule) και κυβικής παρεμβολής(cubic interpolation) .

```
function [out]=trialBFGSWC
x=[-100;200];
Q=eye(2,2);
B=eye(2,2);
alpha=0.3;
beta=0.7;
d=0.00001;
epsilon=1e-6;
[out]=quasiBFGSWC(x,Q,B,d,alpha,beta,epsilon)
```

```
function [out]=quasiBFGSWC(x,Q,B,d,alpha,beta,epsilon)
% x must be a column vector
nn=size(x);
n=nn(1);
eps1=1e-4;
aa=1.7;
theta=0.2;
fold=f(x);
grf=gradf(x);
grm=grf'*grf;
k=0;
out=[k x']
while grm>epsilon
%  STEP 1
    h=-Q*grf;
    fbs=grf'*h;
%  STEP 2
%  Lemarechal, Wolf, cubic
    lamda0=0; f0=fold; fbs0=fbs;
    lamdaL=0; fL=fold; fbsL=fbs;
    lamdaR=0; fR=fold; fbsR=fbs;
    lamdaold=0; fbsold=fbs;
    lamda=1;
    xnew=x+lamda*h;
    fnew=f(xnew);
    grfnew=gradf(xnew);
    fbsnew=grfnew'*h;
    delta1=fnew-f0-alpha*lamda*fbs0;
    delta2=fbsnew-beta*fbs0;
    while ((lamdaR-lamdaL)>=eps1)||(lamdaR==0)
        if delta1<=0
            if delta2>=0
```

```
                break;
            else
                lamdaL=lamda; fL=fnew; fbsL=fbsnew;
            end
        else
            lamdaR=lamda; fR=fnew; fbsR=fbsnew;
        end
        pp=fbsnew+fbsold-3*(fnew-fold)/(lamda-lamdaold);
        DD=pp*pp-fbsnew*fbsold;
        DD=max(0,DD);
        dd=sqrt(DD);
        rr=(fbsnew-pp+dd)/(fbsnew-fbsold+2*dd);
        lamdanew=lamda+rr*(lamdaold-lamda);
        if lamdaR==0
            lamdanew=max(lamdanew,aa*lamda);
        else
            lamdanew=min(lamdanew,lamdaR-theta*(lamdaR-
lamdaL));
            lamdanew=max(lamdanew,lamdaL+theta*(lamdaR-
lamdaL));
        end
        lamdaold=lamda; fold=fnew; fbsold=fbsnew;
        lamda=lamdanew;
        xnew=x+lamda*h;
        fnew=f(xnew);
        grfnew=gradf(xnew);
        fbsnew=grfnew'*h;
        delta1=fnew-f0-alpha*lamda*fbs0;
        delta2=fbsnew-beta*fbs0;
    end
%  STEP 3
    xnew=x+lamda*h;
%  STEP 4
    p=xnew-x;
    grfnew=gradf(xnew);
    dir=grfnew'*h;
    q=grfnew-grf;
%    if round(mod(k,(n+1)))==n;
%        Q=eye(n,n);
%    else
        qq=Q*q;
        QQ=qq*p';
        pq=1/(p'*q);
        qqq=q'*qq;
        Q=Q-pq*(QQ+QQ')+((1+pq*qqq)*pq)*(p*p');
%    end
    k=k+1;
    if k>5000, break; end;
    x=xnew;
    grf=grfnew;
    grm=grf'*grf;
    fold=fnew;
    [k fbsold h' x']
```

```
    out=[out;[k x']];
end
function [r]=f(x)
%r=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
    r=100*(x(2)-x(1)^2)^2+(1-x(1))^2+100*(x(4)-x(3)^2)^2+(1-
x(3))^2+10*(x(2)+x(4)-2)^2;

function [r]=gradf(x)
% r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
  r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-
x(1)^2)+20*(x(2)+x(4)-2);-400*x(3)*(x(4)-x(3)^2)-2*(1-
x(3));200*(x(4)-x(3)^2)+20*(x(2)+x(4)-2)];
```

ΑΛΓΟΡΙΘΜΟΣ DFP ΜΕ ΧΡΗΣΗ ΚΑΝΟΝΑ WOLFE ΚΑΙ ΚΥΒΙΚΗΣ ΠΡΟΣΑΡΜΟΓΗΣ

Ο παρακάτω κώδικας υλοποιεί τον αλγόριθμο DFP και αποτελείται από δύο υποπρογράμματα. Η εύρεση του μήκους βήματος λ_i γίνεται με συνδυασμό κανόνα Wolfe(Wolfe's rule) και κυβικής παρεμβολής(cubic interpolation) .

```
function [out]=trialDFPWC
x=[-3;-10;-5;-10];
Q=eye(4,4);
B=eye(4,4);
alpha=0.3;
beta=0.7;
d=0.00001;
epsilon=1e-6;
[out]=quasiDFPWC(x,Q,B,d,alpha,beta,epsilon)
```

```
function [out]=quasiDFPWC(x,Q,B,d,alpha,beta,epsilon)
% x must be a column vector
nn=size(x);
n=nn(1);
eps1=1e-4;
aa=1.7;
theta=0.2;
fold=f(x);
grf=gradf(x);
grm=grf'*grf;
k=0;
out=[k x']
while grm>epsilon
%  STEP 1
```

```
    h=-Q*grf;
    fbs=grf'*h;
%  STEP 2
%  Lemarechal, Wolf, cubic
    lamda0=0; f0=fold; fbs0=fbs;
    lamdaL=0; fL=fold; fbsL=fbs;
    lamdaR=0; fR=fold; fbsR=fbs;
    lamdaold=0; fbsold=fbs;
    lamda=1;
    xnew=x+lamda*h;
    fnew=f(xnew);
    grfnew=gradf(xnew);
    fbsnew=grfnew'*h;
    delta1=fnew-f0-alpha*lamda*fbs0;
    delta2=fbsnew-beta*fbs0;
    while ((lamdaR-lamdaL)>=eps1)||(lamdaR==0)
        if delta1<=0
            if delta2>=0
                break;
            else
                lamdaL=lamda; fL=fnew; fbsL=fbsnew;
            end
        else
            lamdaR=lamda; fR=fnew; fbsR=fbsnew;
        end
        pp=fbsnew+fbsold-3*(fnew-fold)/(lamda-lamdaold);
        DD=pp*pp-fbsnew*fbsold;
        DD=max(0,DD);
        dd=sqrt(DD);
        rr=(fbsnew-pp+dd)/(fbsnew-fbsold+2*dd);
        lamdanew=lamda+rr*(lamdaold-lamda);
        if lamdaR==0
            lamdanew=max(lamdanew,aa*lamda);
        else
            lamdanew=min(lamdanew,lamdaR-theta*(lamdaR-
lamdaL));
            lamdanew=max(lamdanew,lamdaL+theta*(lamdaR-
lamdaL));
        end
        lamdaold=lamda; fold=fnew; fbsold=fbsnew;
        lamda=lamdanew;
        xnew=x+lamda*h;
        fnew=f(xnew);
        grfnew=gradf(xnew);
        fbsnew=grfnew'*h;
        delta1=fnew-f0-alpha*lamda*fbs0;
        delta2=fbsnew-beta*fbs0;
    end
%  STEP 3
    xnew=x+lamda*h;
%  STEP 4
    p=xnew-x;
    grfnew=gradf(xnew);
```

```
    dir=grfnew'*h;
    q=grfnew-grf;
%    if round(mod(k,(n+1)))==n;
%        Q=eye(n,n);
%    else
       qq=Q*q;
       pq=1/(p'*q);
       qqq=1/(q'*qq);
       Q=Q-qqq*(qq*qq')+pq*(p*p');
%    end
    k=k+1;
    if k>5000, break; end;
    x=xnew;
    grf=grfnew;
    grm=grf'*grf;
    fold=fnew;
    [k fbsold h' x']
    out=[out;[k x']];
end
function [r]=f(x)
%r=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
 r=100*(x(2)-x(1)^2)^2+(1-x(1))^2+100*(x(4)-x(3)^2)^2+(1-
x(3))^2+10*(x(2)+x(4)-2)^2;

function [r]=gradf(x)
% r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
  r=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-
x(1)^2)+20*(x(2)+x(4)-2);-400*x(3)*(x(4)-x(3)^2)-2*(1-
x(3));200*(x(4)-x(3)^2)+20*(x(2)+x(4)-2)];
```

ΙΣΟΫΨΕΙΣ ΣΥΝΑΡΤΗΣΕΩΝ

(CONTOURS OF THE TEST FUNCTIONS)

Ο παρακάτω κώδικας υλοποιεί τις ισοϋψείς των υπό εξέταση συναρτήσεων(test functions) και ανάλογα με την quasi-newton μέθοδο που εκτελούμε δείχνει την αλληλουχία των σημείων της ακολουθίας που κατασκευάζει η μέθοδος.

Προφανώς ο κώδικας είναι ενιαίος για όλες τις μεθόδους και το μόνο που διαφοροποιείται είναι το όνομα στην κλήση της συνάρτησης και η συνάρτηση που καλείται από το υποπρόγραμμα.

```
function contquasi(min1,max1,n1,min2,max2,n2)
hold on
step1=(max1-min1)/n1;
x1=min1:step1:max1;
step2=(max2-min2)/n2;
x2=min2:step2:max2;
```

```
nn1=size(x1);
nn1=nn1(2);
nn2=size(x2);
nn2=nn2(2);
for i=1:nn1
    x(1)=x1(i);
    for k=1:nn2
        x(2)=x2(k);
        Z(i,k)=fRosenbrock(x);
%        Z1(i,k)=fd1exm2(x,y);
%        Z2(i,k)=fd2exm2(x,y);
    end
end
v=[0 0.005 1 5 10 50 100 500 1000 5000];
vder=[0 0.001];
Z=Z';
%Z1=Z1';
%Z2=Z2';
contour(x1,x2,Z,v);
%contour(x1,x2,Z1,vder,'-r');
%contour(x1,x2,Z2,vder,'-g');
[out]=trial;
plot(out(:,2),out(:,3),'-r');
hold off
```

Κώδικας συνάρτησης που καλείται από το πρόγραμμα contquasi:

```
function [r]=fRosenbrock(x)
r=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
```

Βιβλιογραφία-Αναφορές

[1] Ν. Μαράτος, Τεχνικές Βελτιστοποίησης, ΕΜΠ, Αθήνα, 1990

[2] M. Avriel, Nonlinear Programming, Analysis and Methods, Dover, 2003

[3] M. Bazarra, C. Shetty, Nonlinear Programming, Theory and Algorithms, Third Edition, John Wiley and Sons, 2006

[4]A. Belegundu,T. Chandrupatla ,Optimization concepts and applications in engineering. New York: Cambridge University Press,2011.

[5] D. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, Massachusetts, 1995

[6] Bonnans, Gilbert, Lemarechal, Sagastizabal, Numerical Optimization, Second Edition, Springer, 2006

[7]S. Boyd,L. Vandenberghe, Convex optimization. Cambridge, UK: Cambridge University Press,2004

[8]J. Dennis,R. Schnabel ,Numerical methods for unconstrained optimization and nonlinear equations. Philadelphia: SIAM,1996

[9] R. Fletcher, Practical Methods of Optimization, Second Edition, John Wiley and Sons, 1987

[10] S Jacoby, J.Kowalik,J. Pizzo ,Iterative methods for nonlinear optimization problems. Englewood Cliffs, N.J: Prentice-Hall, 1772.

[11] D. Luenberger, Linear and Nonlinear Programming, Third Edition, Springer, 2008

[12] J. Nocedal.,S.Wright Numerical optimization. New York: Springer,1999

[13]E. Polak, Optimization-Algorithms and Consistent Approximations, Springer, 1997

[14]E. Polak, Computational Methods in Optimization, An Unified Approach, Academic Press, 1971

[15]S. Rao, Engineering optimization: Theory and practice. New York: Wiley, 1996

[16]P. Venkataraman, Applied Optimization with Matlab Programming, John Wiley and Sons, 2002

[17]K. Schittkowski, Test Examples for Nonlinear Programming Codes, http://www.ai7.uni-bayreuth.de/ , 2009

[18]J. More, B. Garbow, K. Hillstrom. ,Testing Unconstrained Optimization Software, ACM Transactions on Mathematical Software, Vol 7, No. 1, March , Pages 77-41,1981.

[19]Ν. Καδιανάκης, Σ. Καρανάσιος, Α. Φελούρης, Ανάλυση ΙΙ - Συναρτήσεις πολλών μεταβλητών, Έκδοση 7η, Αθήνα 2007

[20]Optimization Toolbox Matlab

[21]MathematicalOptimization http://en.wikipedia.org/wiki/Mathematical_optimization