



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

## Επίλυση συστημάτων μη γραμμικών εξισώσεων κυρτών διαφορών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Χ. Βαρελάς

Επιβλέπων : Νικόλαος Γ. Μαράτος  
Καθηγητής

Αθήνα, Σεπτέμβριος, 2014





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

## Επίλυση συστημάτων μη γραμμικών εξισώσεων κυρτών διαφορών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Χ. Βαρελάς

Επιβλέπων : Νικόλαος Γ. Μαράτος  
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....  
Νικόλαος Μαράτος  
Καθηγητής

.....  
Κωνσταντίνος Βουρνάς  
Καθηγητής

.....  
Κωνσταντίνος Τζαφέστας  
Επίκουρος Καθηγητής

Αθήνα, Σεπτέμβριος 2014

.....  
Κωνσταντίνος Χ. Βαρελάς

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Κωνσταντίνος Βαρελάς

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Ο σκοπός της συγκεκριμένης διπλωματικής εργασίας ήταν η επίλυση συστημάτων μη γραμμικών εξισώσεων κυρτών διαφορών. Το σύστημα εξισώσεων ανήχθη σε ένα ισοδύναμο πρόβλημα ελαχιστοποίησης κοίλης συνάρτησης με κυρτούς περιορισμούς. Το νέο πρόβλημα αντιμετωπίστηκε με έναν επαναληπτικό αλγόριθμο εξωτερικής προσέγγισης, ο οποίος εγγυάται τη σύγκλιση σε ολικό ελάχιστο. Η υλοποίηση του αλγορίθμου έγινε στο περιβάλλον Matlab, και έγινε εφαρμογή του σε διάφορα παραδείγματα. Καταγράφηκαν και αξιολογήθηκαν τα αποτελέσματα και επαληθεύτηκε η ορθή λειτουργία της μεθόδου σε περιπτώσεις ύπαρξης μιας ή και περισσότερων λύσεων. Τέλος, έγινε μια αξιολόγηση της απόδοσης και του υπολογιστικού κόστους του αλγορίθμου.

## Λεξεις κλειδιά

συναρτήσεις κυρτών διαφορών, κοίλη ελαχιστοποίηση, ολική ελαχιστοποίηση, μέθοδοι εξωτερικής προσέγγισης, κοίλη συνάρτηση, κυρτό σύνολο, κυρτά πολύεδρα, μη γραμμικοί περιορισμοί

## **Abstract**

The goal of this thesis was to solve systems of equations of nonlinear, d.c. functions i.e. functions which can be expressed as differences of convex functions. The system was transformed to an equivalent problem of concave minimization with convex constraints. The new problem was solved with a global optimization, outer approximation method, which guarantees convergence to a global minimum. The implementation of the algorithm was made using the software package Matlab , and it was applied to several examples. The results were evaluated and the proper use of this method was tested both in cases with one and more solutions. Finally, an evaluation of the computational cost of the algorithm was made.

## **Key words**

d.c. functions, concave minimization, global optimization, outer approximation methods, concave function, convex set, convex polyhedra, nonlinear constraints

## **Ευχαριστίες**

Η διπλωματική εργασία εκπονήθηκε την περίοδο Μαρτίου 2014 έως και Σεπτεμβρίου 2014, υπό την επίβλεψη του κ. Νικόλαου Μαράτου, καθηγητή της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, στον οποίο θα ήθελα να εκφράσω θερμά τις ευχαριστίες μου για την ανάθεση της εργασίας, τη βοήθεια και καθοδήγηση που μου παρείχε και την άψογη συνεργασία του σε όλη την περίοδο εκπόνησης της διπλωματικής εργασίας. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Γιάννη Σπηλιώτη για τις πολύτιμες συμβουλές του σχετικά με τη συνέχεια των σπουδών μου. Τέλος, ευχαριστώ την οικογένειά μου για τη στήριξη που μου έδειξαν σε όλα τα χρόνια των σπουδών μου.

# Περιεχόμενα

## 1 Κεφάλαιο 1-Εισαγωγή

1.1 Το πρόβλημα της κοίλης ελαχιστοποίησης.....	9
1.2 Μέθοδοι εξωτερικής προσέγγισης.....	12
1.3 D.C. συναρτήσεις και ιδιότητες.....	14
1.4 Ανάλυση σε διαφορές κυρτών συναρτήσεων.....	15
1.5 Κριτήρια κυρτότητας.....	17
1.6 Πράξεις που διατηρούν την κυρτότητα.....	17

## 2 Διατύπωση του προβλήματος και επίλυση

2.1 Το πρόβλημα και το ισοδύναμό του.....	19
2.2 Αλγόριθμος.....	21
2.3 Προσαρμογή του προβλήματος.....	24

## 3 Παραδείγματα - Συμπεράσματα

3.1 Παράδειγμα 1.....	26
3.2 Παράδειγμα 2.....	34
3.3 Παράδειγμα 3.....	39
3.4 Παράδειγμα 4.....	43
3.5 Παράδειγμα 5.....	47
3.6 Συμπεράσματα.....	52

<b>Παράρτημα Κώδικας MATLAB.....</b>	<b>54</b>
--------------------------------------	-----------

<b>Βιβλιογραφία.....</b>	<b>79</b>
--------------------------	-----------



# Κεφάλαιο 1

## Εισαγωγή

Το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η επίλυση συστημάτων μη γραμμικών εξισώσεων κυρτών διαφορών (D.C. συναρτήσεων). Το σύνολο συναρτήσεων κυρτών διαφορών περιλαμβάνει συναρτήσεις  $f$  που μπορούν να γραφούν στη μορφή:

$$f = g - h, \quad f, g, h : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$$

όπου οι συναρτήσεις  $g$  και  $h$  είναι κυρτές. Το πρόβλημα επίλυσης ενός τέτοιου συστήματος εξισώσεων ανάγεται σε πρόβλημα κοίλης ελαχιστοποίησης, δηλαδή ελαχιστοποίησης μιας κοίλης συνάρτησης με κυρτούς περιορισμούς. Αυτό επιτυγχάνεται επεκτείνοντας τη μέθοδο που αναπτύσσεται στη διπλωματική εργασία του Σ. Κιναλή [8]. Η μέθοδος αυτή παρουσιάζεται στο κεφάλαιο 2. Το καινούργιο πρόβλημα κοίλης ελαχιστοποίησης επιλύεται με αλγόριθμο ολικής ελαχιστοποίησης, και πιο συγκεκριμένα με τον αλγόριθμο εξωτερικής προσέγγισης της K.L.Hoffman [9].

## 1.1 Το πρόβλημα της κοίλης ελαχιστοποίησης

Η γενική διατύπωση προβλημάτων ολικής βελτιστοποίησης είναι η παρακάτω [1] :

*Δοθέντος ενός μη κενού, κλειστού συνόλου  $D \subset \mathbb{R}^n$  και δοθείσης μιας συνεχούς συνάρτησης  $f : A \rightarrow \mathbb{R}$ , όπου  $D \subset A \subset \mathbb{R}^n$  βρες τουλάχιστον ένα σημείο  $x^* \in D$  που ικανοποιεί  $f(x^*) \leq f(x) \forall x \in D$  ή δείξε πως τέτοιο σημείο δεν υπάρχει.*

Τα προβλήματα αυτής της κατηγορίας παρουσιάζουν μεγάλες δυσκολίες στην επίλυσή τους. Αυτό οφείλεται στο ότι αναζητείται σημείο ολικού ελαχίστου. Σημειώνεται πως δεν υπάρχει τοπικό κριτήριο που να εγγυάται πως ένα τοπικό ελάχιστο είναι και ολικό. Συνεπώς οι συμβατικές μέθοδοι βελτιστοποίησης που κάνουν χρήση κλίσεων ή παραγώγων δεν είναι πάντα ικανές να εντοπίσουν ή να αναγνωρίσουν ένα ολικό ελάχιστο. Φυσικά, υπάρχουν κατηγορίες προβλημάτων που έχουν εγγενώς την ιδιότητα κάθε τοπικό ελάχιστο να είναι και ολικό. Χαρακτηριστικό παράδειγμα είναι η ελαχιστοποίηση κυρτής συνάρτησης με κυρτούς

περιορισμούς (δηλαδή ελαχιστοποίηση πάνω σε κυρτό σύνολο). Σε αυτές τις περιπτώσεις μια συμβατική μέθοδος βελτιστοποίησης θα ήταν επιτυχής. Ωστόσο, σε πολλά προβλήματα η ιδιότητα αυτή δεν υπάρχει. Μια σημαντική κατηγορία τέτοιων προβλημάτων, τα οποία στην ουσία αποτελούν και το κυρίως αντικείμενο της συγκεκριμένης εργασίας, είναι τα προβλήματα κοίλης ελαχιστοποίησης, τα οποία εν γένει διατυπώνονται ως εξής:

*Ελαχιστοποίησε την  $f(x)$  με περιορισμό  $x \in D$ , όπου το  $D \subset \mathbb{R}^n$  είναι μη κενό, κλειστό και κυρτό, ενώ η  $f: A \rightarrow \mathbb{R}$  είναι κοίλη στο  $A \supset D$ .*

Στα προβλήματα κοίλης ελαχιστοποίησης ένα σημείο τοπικού ελαχίστου δεν είναι απαραίτητα και σημείο ολικού ελαχίστου. Υπάρχουν όμως οι παρακάτω 4 βασικές ιδιότητες, κρίσιμες για την κατασκευή μεθόδων επίλυσής τους:

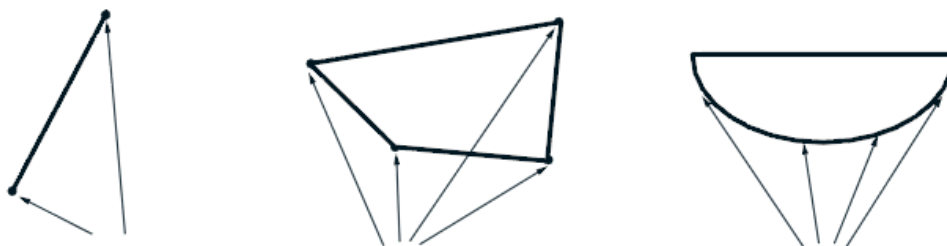
### Ιδιότητα 1

Το επιτρεπτό σύνολο, δηλαδή το σύνολο στο οποίο αναζητείται το ολικό ελάχιστο, είναι κυρτό.

### Ιδιότητα 2

Η δεύτερη και πιο σημαντική ιδιότητα φαίνεται στο θεώρημα που ακολουθεί. Πρώτα όμως ας δώσουμε τον ορισμό του ακραίου σημείου (extreme point) ενός κυρτού συνόλου:

*Ορισμός: Ένα σημείο  $x$  ονομάζεται ακραίο σημείο ενός κυρτού συνόλου  $D \subset \mathbb{R}^n$  εάν  $x \in D$  και το  $x$  δεν μπορεί να εκφραστεί ως κυρτός συνδυασμός δύο σημείων  $y$  και  $z$ , όπου  $y, z \in D$  και  $x \neq y, x \neq z$ .*



Σχήμα 1.1.1: Ακραία σημεία κυρτών συνόλων

*Θεώρημα: Έστω  $f: D \rightarrow \mathbb{R}$  μία κοίλη συνάρτηση, όπου το  $D$  είναι μη κενό, συμπαγές και κυρτό υποσύνολο του  $\mathbb{R}^n$ . Τότε η  $f$  παρουσιάζει ελάχιστο σε κάποιο (ή κάποια) από τα ακραία σημεία του  $D$ .*

Ιδιότητα 3:

Το επιτρεπτό σύνολο  $D$  είναι η τομή των κλειστών ημιχώρων που το περιέχουν.

Ιδιότητα 4:

Η προς ελαχιστοποίηση συνάρτηση  $f$  (αντικειμενική συνάρτηση) είναι παντού συνεχής. Επιπλέον είναι συνάρτηση Lipschitz, δηλαδή υπάρχει πραγματικός αριθμός  $\alpha \geq 0$  ώστε  $|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \alpha \|\mathbf{x}_1 - \mathbf{x}_2\|, \forall \mathbf{x}_1, \mathbf{x}_2 \in D$ . Ακόμη, για κάθε σημείο  $\mathbf{x} \in \mathbb{R}^n$  υπάρχει ένα διάνυσμα  $\mathbf{p}(\mathbf{x})$  τέτοιο ώστε  $\forall \mathbf{y} \in \mathbb{R}^n$  να ισχύει

$$f(\mathbf{y}) - f(\mathbf{x}) \leq \mathbf{p}(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$$

Πολλές κατηγορίες προβλημάτων που αφορούν την επιστήμη, την τεχνολογία και την οικονομία μπορούν να μοντελοποιηθούν ως προβλήματα κοίλης ελαχιστοποίησης. Τέσσερις από τις πιο σημαντικές κατηγορίες τέτοιων προβλημάτων είναι οι ακόλουθες [11] :

Οικονομίες κλίμακας:

Συχνά εμφανίζεται η ανάγκη επιλογής των επιπέδων  $x_i, i = 1 \dots n$  για  $n$  δραστηριότητες, οι οποίες παράγουν ανεξάρτητα κόστη  $f_i(x_i)$ , υπό τους περιορισμούς  $g_j(\mathbf{x}) \leq 0, j = 1 \dots m$ . Οι περιορισμοί είναι κυρτοί, και ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος υπό αυτούς τους περιορισμούς, δηλαδή να ελαχιστοποιηθεί η αντικειμενική συνάρτηση  $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ .

Ακέραιος προγραμματισμός:

Το πρόβλημα του ακέραιου γραμμικού προγραμματισμού είναι το παρακάτω:

$$(IP) : \min \mathbf{c}^T \mathbf{x}$$

$$\text{με περιορισμό } A\mathbf{x} \leq \mathbf{b}, x_i = 0 \text{ ή } x_i = 1, i = 1 \dots n$$

με δεδομένα  $A \in \mathbb{R}^{m \times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ . Αποδεικνύεται πως το (IP), για αρκετά μεγάλο θετικό αριθμό  $M$ , είναι ισοδύναμο με το εξής:

$$(CP1) : \min f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + M\mathbf{x}^T(\mathbf{e} - \mathbf{x})$$

$$\text{με περιορισμό } A\mathbf{x} \leq \mathbf{b}, 0 \leq x_i \leq 1, i = 1 \dots n$$

Η συνάρτηση  $f(\mathbf{x})$  είναι κοίλη (ως άθροισμα μιας γραμμικής συνάρτησης και μιας αρνητικά ορισμένης τετραγωνικής μορφής), συνεπώς το (CP1) είναι πρόβλημα κοίλης ελαχιστοποίησης.

Πρόβλημα γραμμικής συμπληρωματικότητας (linear complementarity):

Πρόκειται για το πρόβλημα εύρεσης  $\mathbf{x} \in \mathbb{R}^n$  έτσι ώστε:

$$(LCP) : Q\mathbf{x} + \mathbf{g} \geq 0, x_i \geq 0, \mathbf{x}^T(Q\mathbf{x} + \mathbf{g}) = 0$$

όπου τα  $Q \in \mathbb{R}^{n \times n}$ ,  $g \in \mathbb{R}^n$  είναι δοθέντα. Αποδεικνύεται πως το (LCP) είναι ισοδύναμο με το :

$$(CP2) : \min f(x) = \sum_{i=1}^n (x_i - \max\{0, x_i - (Qx + g)_i\})$$

$$\text{με περιορισμό } (Qx + g)_i \geq 0, x_i \geq 0, i = 1 \dots n$$

Όπως και πριν, η αντικειμενική συνάρτηση  $f(x)$  είναι κοίλη, συνεπώς το (CP2) είναι πρόβλημα κοίλης ελαχιστοποίησης.

Ελαχιστοποίηση της διαφοράς δύο κυρτών συναρτήσεων:

Πρόκειται για το πρόβλημα:

$$(D) : \min(f_1(x^1) - f_2(x^2))$$

$$\text{με περιορισμό } x^1, x^2 \in D$$

όπου  $x^1 \in \mathbb{R}^{n_1}$ ,  $x^2 \in \mathbb{R}^{n_2}$ ,  $f_1: \mathbb{R}^{n_1} \rightarrow \mathbb{R}$ ,  $f_2: \mathbb{R}^{n_2} \rightarrow \mathbb{R}$ , οι  $f_1$  και  $f_2$  είναι κυρτές συναρτήσεις και το  $D$  είναι κυρτό συμπαγές σύνολο. Όπως και πριν, υπάρχει ισοδύναμο πρόβλημα κοίλης ελαχιστοποίησης, το οποίο εδώ προκύπτει εισάγοντας μια νέα μεταβλητή  $t$  :

$$(CP3) : \min (t - f_2(x^2))$$

$$\text{με περιορισμό } f_1(x^1) \leq t, x^1, x^2 \in D.$$

Η αντικειμενική συνάρτηση  $t - f_2(x^2)$  είναι κοίλη και το επιτρεπτό σύνολο κυρτό.

Υπάρχουν τρεις βασικές αλγοριθμικές προσεγγίσεις για την κοίλη ελαχιστοποίηση. Αρχικά έχουμε μεθόδους απαρίθμησης, οι οποίες όμως εφαρμόζονται μόνο σε περιπτώσεις όπου το επιτρεπτό σύνολο είναι κυρτό πολυέδρο, και συγκρίνουν τις τιμές της αντικειμενικής συνάρτησης στις κορυφές του πολυέδρου. Στη συνέχεια, υπάρχουν οι λεγόμενες διαδοχικές μέθοδοι διαχωρισμού (branch and bound) . Τέλος, έχουμε τις μεθόδους εξωτερικής προσέγγισης. Στην εργασία αυτή ασχολούμαστε με την τελευταία κατηγορία.

## 1.2 Μέθοδοι εξωτερικής προσέγγισης

Έστω το πρόβλημα ολικής βελτιστοποίησης (P):

$$\text{‘ελαχιστοποίησε την } f(x)$$

$$\text{με περιορισμό } x \in D \text{ ‘}$$

όπου η  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  είναι συνεχής και το  $D \subset \mathbb{R}^n$  είναι κλειστό. Η ιδέα των μεθόδων εξωτερικής προσέγγισης για την επίλυση του (P) είναι η αντικατάστασή του με μια σειρά προβλημάτων ( $Q_k$ ):

“ελαχιστοποίησε την  $f(x)$

με περιορισμό  $x \in D_k$  “

όπου  $\mathbb{R}^n \supset D_1 \supset D_2 \supset \dots \supset D$  και  $\min_{x \in D_k} f(x) \xrightarrow{k \rightarrow \infty} \min_{x \in D} f(x)$ . Στις περισσότερες υλοποιήσεις μεθόδων εξωτερικής προσέγγισης, τα σύνολα  $D_k$  είναι κυρτά πολύεδρα που περιγράφονται από ανισωτικούς περιορισμούς της μορφής:  $l_i^k(x) \leq 0, i=1, \dots, m$ . Η διαδικασία η οποία ακολουθείται είναι η εξής:

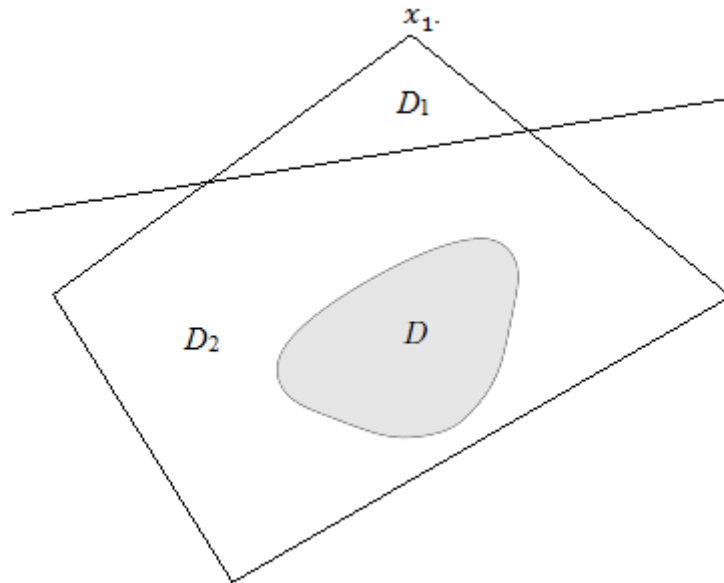
Ξεκινάμε με ένα αρχικό πολύεδρο  $D_1 \supset D$  και ελαχιστοποιούμε την  $f$  στο  $D_1$ . Βάσει της ιδιότητας 2 της προηγούμενης παραγράφου, αυτό γίνεται απλά συγκρίνοντας την τιμή της  $f$  στις κορυφές του  $D_1$ , αφού μόνο αυτές αποτελούν ακραία σημεία του  $D_1$ . Έστω πως η λύση είναι η  $x_1$ . Προφανώς, αφού  $D_1 \supset D$ , η τιμή  $f(x_1)$  είναι ένα κάτω φράγμα για τη λύση του προβλήματος (P) και αν  $x_1 \in D$ , τότε το σημείο  $x_1$  αποτελεί λύση του (P).

Εάν  $x_1 \notin D$ , βάσει γνωστού θεωρήματος παρόμοιου με την ιδιότητα 3 της προηγούμενης παραγράφου, μπορεί πάντα να βρεθεί ένα υπερεπίπεδο που να διαχωρίζει το  $x_1$  από το  $D$ . Δηλαδή μπορούν να βρεθούν ένα διάνυσμα  $p_1 \in \mathbb{R}^n$  και ένας αριθμός  $a_1 \in \mathbb{R}$  ώστε:

$$p_1^T x_1 + a_1 > 0 \text{ και}$$

$$p_1^T x + a_1 \leq 0 \forall x \in D$$

Η τελευταία ανισότητα προστίθεται στους περιορισμούς που ορίζουν το  $D_1$  και επομένως ορίζεται ένα νέο πολύεδρο  $D_2$  ώστε  $D_1 \supset D_2 \supset D$ . Το  $D_2$  αντικαθιστά το  $D_1$  και η διαδικασία επαναλαμβάνεται.



Σχήμα 1.2.1 : Η ευθεία διαχωρίζει το  $x_1$  από το  $D$  και ορίζει το  $D_2$

Στη συγκεκριμένη εργασία, όπως θα φανεί αναλυτικότερα στην παρουσίαση του αλγορίθμου (κεφάλαιο 2) , τα προβλήματα  $(Q_k)$  αντιστοιχούν σε ισοδύναμα προβλήματα ελαχιστοποίησης μιας γραμμικής συνάρτησης, και τα υπερεπίπεδα που ορίζουν τα νέα πολύεδρα  $D_k$  είναι εφαπτόμενα στο  $D$ .

### 1.3 D.C. Συναρτήσεις και ιδιότητες

Οι συναρτήσεις κυρτών διαφορών, ή αλλιώς D.C. συναρτήσεις, όπως αναφέρθηκε, είναι συναρτήσεις  $f$  της μορφής :

$$f = g - h , f, g, h : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$$

,όπου οι  $g, h$  είναι κυρτές στο  $D$ . Κάποιες από τις βασικές ιδιότητες αυτής της κατηγορίας συναρτήσεων φαίνονται στο παρακάτω θεώρημα [1][2]:

Θεώρημα : Έστω  $f, f_i : D \rightarrow \mathbb{R}$  συναρτήσεις κυρτών διαφορών,  $i=1 \dots m$ . Τότε οι παρακάτω είναι επίσης συναρτήσεις κυρτών διαφορών

- $\sum_{i=1}^m \lambda_i f_i$  , για οποιουσδήποτε πραγματικούς  $\lambda_i, i=1 \dots m$ .
- $\max_{i=1 \dots m} f_i$  και  $\min_{i=1 \dots m} f_i$
- $|f|, f^+ = \max\{0, f\}, f^- = \min\{0, f\}$
- $\prod_{i=1}^m f_i$

Εκτός από τις παραπάνω ιδιότητες, ιδιαίτερη σημασία έχει το εύρος της συγκεκριμένης κατηγορίας συναρτήσεων, το οποίο γίνεται φανερό από τα παρακάτω.

Θεώρημα: Κάθε τοπικά D.C. συνάρτηση είναι D.C. [3]

Στο παραπάνω θεώρημα, μια συνάρτηση  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  καλείται τοπικά D.C. συνάρτηση εάν  $\forall \mathbf{x}^0 \in \mathbb{R}^n$  υπάρχει περιοχή  $N = N(\mathbf{x}^0, \varepsilon)$  του  $\mathbf{x}^0$  και κυρτές συναρτήσεις  $p, q$  έτσι ώστε  $f(\mathbf{x}) = p(\mathbf{x}) - q(\mathbf{x}) \forall \mathbf{x} \in N$ . Ως αποτέλεσμα αυτού του θεωρήματος, έχουμε τα παρακάτω τρία λήμματα:

Λήμμα 1: Κάθε συνάρτηση  $f \in C^2$  είναι D.C. συνάρτηση.

Λήμμα 2: Για κάθε πραγματική συνάρτηση  $f$  συνεχή σε ένα συμπαγές κυρτό σύνολο  $D \subset \mathbb{R}^n$ , υπάρχει μια ακολουθία συναρτήσεων  $f_i$  που είναι D.C. και συγκλίνει ομοιόμορφα στην  $f$  στο  $D$ .

Λήμμα 3: Μια συνεχής τμηματικά-γραμμική συνάρτηση  $f$  ορισμένη σε ένα διάστημα  $I$  είναι D.C. στο  $I$ .

Με βάση αυτά τα συμπεράσματα γίνεται φανερό το εύρος της κατηγορίας των D.C. συναρτήσεων, και επομένως και η σημασία ενός αλγορίθμου που επιτυγχάνει την επίλυση συστημάτων εξισώσεων κυρτών διαφορών.

## 1.4 Ανάλυση σε διαφορές κυρτών συναρτήσεων

Οι προηγούμενες προτάσεις είναι ιδιαίτερης σημασίας γιατί υποδεικνύουν ποιές συναρτήσεις είναι συναρτήσεις κυρτών διαφορών, όμως πρακτικά δεν απαντούν στο ερώτημα πώς μπορεί μια συνάρτηση να αναλυθεί ως διαφορά κυρτών συναρτήσεων. Προφανώς το πρόβλημα δεν είναι τετριμμένο και σε κάθε περίπτωση θα πρέπει να γίνει διαφορετική προσπάθεια, ωστόσο υπάρχουν κάποιες γενικές εφαρμογές που αφορούν συνηθισμένα προβλήματα και παρουσιάζονται στη συνέχεια.

1. Μια περίπτωση συνάρτησης που μπορεί να γραφεί τετριμμένα ως διαφορά δύο άλλων κυρτών είναι η τετραγωνική μορφή  $\mathbf{x}^T Q \mathbf{x}$ , όπου η  $Q$  είναι πραγματική, συμμετρική, αόριστη μήτρα. Η τετραγωνική μορφή μπορεί να γραφεί ως:  $\mathbf{x}^T Q \mathbf{x} = \mathbf{x}^T (Q + \|Q\| \cdot I) \mathbf{x} - \|Q\| \cdot \|\mathbf{x}\|^2$ , η οποία αποτελεί μια D.C. αναπαράσταση για οποιαδήποτε νόρμα πίνακα. Το συμπέρασμα αυτό οφείλεται στον T.Phong[4].
2. Επίσης για την περίπτωση αόριστης τετραγωνικής μορφής, υπάρχει ο εναλλακτικός τρόπος αναπαράστασης με το μετασχηματισμό ιδιοδιανυσμάτων:  $\mathbf{x}^T Q \mathbf{x} = \sum_{i:\lambda_i>0} \lambda_i \cdot y_i^2 + \sum_{i:\lambda_i<0} \lambda_i \cdot y_i^2$ .
3. Έστω ένα σημείο  $M \in \mathbb{R}^n$ . Αν συμβολίσουμε με  $d_M(\mathbf{x})$  την ευκλείδεια απόσταση του  $M$  από το  $\mathbf{x} \in \mathbb{R}^n$ , τότε το τετράγωνο της συνάρτησης  $d_M$  αναλύεται ως:  $d_M^2(\mathbf{x}) = \|\mathbf{x}\|^2 - (\|\mathbf{x}\|^2 - d_M^2(\mathbf{x}))$ , αφού μπορεί να αποδειχθεί πως η  $\|\mathbf{x}\|^2 - d_M^2(\mathbf{x})$  είναι κυρτή συνάρτηση του  $x$ . [5]
4. Τέλος, έχει προταθεί D.C. αναπαράσταση για οποιαδήποτε συνάρτηση  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  η οποία ανήκει στην κλάση  $C^2$  [2], [6]. Η αναπαράσταση αυτή στηρίζεται στην ιδιότητα πως εάν  $f \in C^2$ , τότε η συνάρτηση

$$L(\mathbf{x}) = f(\mathbf{x}) + a \cdot \sum_{i=1}^n (x_i^L - x_i)(x_i^U - x_i)$$

όπου  $a \geq \max\{\max_i (0, -\frac{1}{2}\lambda_i^f)\}$  με  $\lambda_i^f$  τις ιδιοτιμές της Hessian μήτρας, είναι κυρτή. Συνεπώς μια D.C.αναπαράσταση για την  $f$  είναι η παρακάτω:

$$f(\mathbf{x}) = L(\mathbf{x}) - a \cdot \sum_{i=1}^n (x_i^L - x_i)(x_i^U - x_i)$$

$$a \geq \max\{\max_i (0, -\frac{1}{2}\lambda_i^f)\}$$



## 1.5 Κριτήρια κυρτότητας

Ως γνωστόν, μια συνάρτηση  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  είναι κυρτή εάν το πεδίο ορισμού της  $D_f$  είναι κυρτό σύνολο και επίσης  $\forall \mathbf{x}, \mathbf{y} \in D_f, \forall \theta \in [0,1]$  ισχύει:

$$f(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta)f(\mathbf{y})$$

Συχνά συμπεραίνουμε την κυρτότητα μιας συνάρτησης με βάση την παραπάνω ανισότητα, η οποία ορίζει τις κυρτές συναρτήσεις. Όμως υπάρχουν και άλλα κριτήρια που εγκυώνται την κυρτότητα μιας συνάρτησης. Αυτά είναι τα ακόλουθα:

### 1. Κριτήριο πρώτης τάξης

Έστω πως η  $f: D \rightarrow \mathbb{R}$  είναι διαφορίσιμη στο ανοικτό σύνολο  $D$ . Τότε η  $f$  είναι κυρτή αν και μόνο αν το πεδίο ορισμού  $D$  είναι κυρτό και  $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$   $\forall \mathbf{x}, \mathbf{y} \in D_f$ . Η  $f$  είναι αυστηρά κυρτή αν και μόνο αν το πεδίο ορισμού  $D$  είναι κυρτό και  $f(\mathbf{y}) > f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$   $\forall \mathbf{x}, \mathbf{y} \in D_f, \mathbf{x} \neq \mathbf{y}$

### 2. Κριτήριο δεύτερης τάξης

Έστω πως η  $f: D \rightarrow \mathbb{R}$  είναι δύο φορές διαφορίσιμη στο ανοικτό σύνολο  $D$ . Τότε η  $f$  είναι κυρτή αν και μόνο αν το πεδίο ορισμού  $D$  είναι κυρτό και η Hessian μήτρα είναι θετικά ημιορισμένη, δηλαδή  $\forall \mathbf{x} \in D_f, \nabla^2 f(\mathbf{x}) \geq 0$ . Εάν ισχύει  $\nabla^2 f(\mathbf{x}) > 0$   $\forall \mathbf{x} \in D_f$ , τότε η  $f$  είναι αυστηρά κυρτή (το αντίστροφο δεν ισχύει).

## 1.6 Πράξεις που διατηρούν την κυρτότητα

Συχνά είναι πιο εύκολο να συμπεράνει κανείς την κυρτότητα ή μη μιας συνάρτησης με βάση άλλες συναρτήσεις που εκ των προτέρων είναι γνωστό πως είναι κυρτές. Για το λόγο αυτό έχει σημασία να είναι γνωστές οι πράξεις μεταξύ συναρτήσεων που διατηρούν την κυρτότητα. Οι σημαντικότερες από αυτές, αρχικά οι πιο απλές και στη συνέχεια πιο εξειδικευμένες, παρουσιάζονται σε αυτή την παράγραφο [7].

### 1. Γραμμικός συνδυασμός κυρτών συναρτήσεων με μη αρνητικούς συντελεστές

Εάν οι συναρτήσεις  $f_i, i = 1 \dots m$  είναι κυρτές και οι συντελεστές  $w_i, i = 1 \dots m$  είναι μη αρνητικοί, τότε η συνάρτηση  $\sum_{i=1}^m w_i f_i$  είναι κυρτή. Ειδικότερα, αν οι συντελεστές  $w_i, i = 1 \dots m$  είναι θετικοί και οι συναρτήσεις  $f_i, i = 1 \dots m$  είναι αυστηρά κυρτές, τότε η  $\sum_{i=1}^m w_i f_i$  είναι αυστηρά κυρτή.

Οι ιδιότητες αυτές γενικεύονται για σειρές συναρτήσεων και ολοκληρώματα. Για παράδειγμα, αν η  $f(x, y)$  είναι κυρτή ως προς τη μεταβλητή  $x$ ,  $\forall y \in A$  και ισχύει  $w(y) \geq 0 \quad \forall y \in A$ , τότε η συνάρτηση  $g(x) = \int_A w(y)f(x, y)dy$  είναι κυρτή.

2. Σύνθεση με ομοπαράλληλικό (affine) μετασχηματισμό

Έστω  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $A \in \mathbb{R}^{n \times m}$ ,  $b \in \mathbb{R}^n$ . Ορίζουμε  $g: \mathbb{R}^m \rightarrow \mathbb{R}$  ως  $g(\mathbf{x}) = f(A\mathbf{x} + \mathbf{b})$  με πεδίο ορισμού  $D_g = \{\mathbf{x} | A\mathbf{x} + \mathbf{b} \in D_f\}$ . Τότε, εάν η  $f$  είναι κυρτή, είναι και η  $g$ .

3. (Κατά σημείο) μέγιστο και supremum

Εάν οι  $f_1, f_2$  είναι κυρτές συναρτήσεις, τότε η  $f(\mathbf{x}) = \max\{f_1(\mathbf{x}), f_2(\mathbf{x})\}$  με πεδίο ορισμού  $D_f = D_{f_1} \cap D_{f_2}$  είναι επίσης κυρτή. Η ιδιότητα αυτή επεκτείνεται και στο (κατά σημείο) supremum σε ένα άπειρο σύνολο κυρτών συναρτήσεων. Εάν  $\forall y \in A$  η  $f(x, y)$  είναι κυρτή ως προς  $x$ , τότε η συνάρτηση  $g(x) = \sup_{y \in A} f(x, y)$  με πεδίο ορισμού  $D_g = \{x | (x, y) \in D_f \quad \forall y \in A, \sup_{y \in A} f(x, y) < \infty\}$  είναι κυρτή.

4. Σύνθεση:

Έστω  $h: \mathbb{R} \rightarrow \mathbb{R}$  και  $g: \mathbb{R}^n \rightarrow \mathbb{R}$ . Υποθέτουμε πως οι δύο αυτές συναρτήσεις είναι δύο φορές παραγωγίσιμες και έχουν πεδία ορισμού  $D_h = D_g = \mathbb{R}$ . Με αυτές τις υποθέσεις, ισχύουν τα παρακάτω

- Η  $h \circ g$  είναι κυρτή εάν η  $h$  είναι κυρτή και αύξουσα και η  $g$  είναι κυρτή
- Η  $h \circ g$  είναι κυρτή εάν η  $h$  είναι κυρτή και φθίνουσα και η  $g$  είναι κοίλη

Σημειώνεται πως παρόμοιες προτάσεις ισχύουν και για σύνθεση διανυσματικών συναρτήσεων. Επίσης υπάρχουν και διαφορετικά κριτήρια που δεν προϋποθέτουν την ύπαρξη δεύτερης τάξης παραγώγων, αλλά έχουν άλλες προϋποθέσεις[7].

5. Ελαχιστοποίηση

Αν η  $f$  είναι κυρτή ως προς  $(x, y)$  και  $C$  είναι ένα κυρτό, μη κενό σύνολο, τότε η συνάρτηση  $g(x) = \inf_{y \in C} f(x, y)$  είναι κυρτή, δεδομένου ότι  $\forall x \quad g(x) > -\infty$ . Η απόδειξη της πρότασης βασίζεται στην ανισότητα Jensen [7].

6. Συνάρτηση προοπτικής (perspective function)

Αν  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ , τότε η συνάρτηση προοπτικής ορίζεται ως :

$$g: \mathbb{R}^{n+1} \rightarrow \mathbb{R}, g(\mathbf{x}, t) = tf\left(\frac{\mathbf{x}}{t}\right)$$

και έχει πεδίο ορισμού:  $D_g = \{(\mathbf{x}, t) | \frac{\mathbf{x}}{t} \in D_f, t > 0\}$ . Ισχύει πως εάν η  $f$  είναι κυρτή, τότε είναι και η  $g$ .

# Κεφάλαιο 2

## Διατύπωση του προβλήματος και επίλυση

Το κεφάλαιο αυτό περιλαμβάνει τη διατύπωση και τη μετατροπή του προβλήματος μη γραμμικών συστημάτων εξισώσεων κυρτών διαφορών σε ισοδύναμο πρόβλημα κοίλης ελαχιστοποίησης και τον αλγόριθμο επίλυσης. Τέλος, αναφέρονται κάποιες δυσκολίες που εμπόδιζαν την άμεση εφαρμογή του αλγορίθμου, και τους τρόπους που αυτές αντιμετωπίστηκαν.

### 2.1 Το πρόβλημα και το ισοδύναμό του

Έστω ότι έχουμε το σύστημα  $n$  εξισώσεων με  $n$  αγνώστους:

$$f_i(\mathbf{x}) = 0, \quad f_i: D \subset \mathbb{R}^n \rightarrow \mathbb{R}, \quad i = 1 \dots n$$

όπου οι συναρτήσεις  $f_i$  είναι D.C., δηλαδή  $\exists g_i, h_i: D \subset \mathbb{R}^n \rightarrow \mathbb{R}$  κυρτές συναρτήσεις ώστε:

$$f_i = g_i - h_i, \quad i = 1 \dots n$$

Το πρόβλημα μετασχηματίζεται με τον τρόπο που αναφέρεται στη διπλωματική εργασία του Σ. Κιναλή [8], δηλαδή ως εξής:

$$\begin{aligned} f_i(\mathbf{x}) = 0 &\Leftrightarrow \\ \Leftrightarrow g_i(\mathbf{x}) = h_i(\mathbf{x}) = y_i &\Leftrightarrow \\ \Leftrightarrow \begin{cases} g_i(\mathbf{x}) - y_i = 0 \\ h_i(\mathbf{x}) - y_i = 0 \end{cases}, i = 1 \dots n & \end{aligned}$$

Ορίζοντας τις συναρτήσεις  $F_i: \mathbb{R}^{2n} \rightarrow \mathbb{R}$  ως εξής:

$$\begin{cases} F_i(\mathbf{x}, \mathbf{y}) = g_i(\mathbf{x}) - y_i \\ F_{n+i}(\mathbf{x}, \mathbf{y}) = h_i(\mathbf{x}) - y_i \end{cases}, i = 1 \dots n$$

το πρόβλημα εκφράζεται ως εξής:

$$F_i(\mathbf{x}, \mathbf{y}) = 0, \quad i = 1 \dots 2n$$

## ΚΕΦΑΛΑΙΟ 2-ΔΙΑΤΥΠΩΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΕΠΙΛΥΣΗ

Με τον τρόπο αυτό , έχουμε πλέον  $2n$  εξισώσεις με  $2n$  αγνώστους. Ωστόσο, το πλεονέκτημα είναι πως τώρα οι συναρτήσεις  $F_i$  ,  $i = 1 \dots 2n$  είναι κυρτές. Αυτό προκύπτει εύκολα από τον ορισμό της κυρτής συνάρτησης:  $\forall \theta \in [0,1]$  , αφού οι  $g_i(\mathbf{x})$  είναι κυρτές, ισχύει

$$\begin{aligned} F_i(\theta \mathbf{x} + (1 - \theta)\mathbf{x}, \theta \mathbf{y} + (1 - \theta)\mathbf{y}) &= g_i(\theta \mathbf{x} + (1 - \theta)\mathbf{x}) - \theta y_i - (1 - \theta)y_i \\ \Rightarrow F_i(\theta \mathbf{x} + (1 - \theta)\mathbf{x}, \theta \mathbf{y} + (1 - \theta)\mathbf{y}) &\leq \theta g_i(\mathbf{x}) + (1 - \theta)g_i(\mathbf{x}) - \theta y_i - (1 - \theta)y_i \\ \Rightarrow F_i(\theta \mathbf{x} + (1 - \theta)\mathbf{x}, \theta \mathbf{y} + (1 - \theta)\mathbf{y}) &\leq \theta(g_i(\mathbf{x}) - y_i) + (1 - \theta)(g_i(\mathbf{x}) - y_i) \\ \Rightarrow F_i(\theta \mathbf{x} + (1 - \theta)\mathbf{x}, \theta \mathbf{y} + (1 - \theta)\mathbf{y}) &\leq \theta F_i(\mathbf{x}, \mathbf{y}) + (1 - \theta)F_i(\mathbf{x}, \mathbf{y}), i = 1 \dots n \end{aligned}$$

Ομοια, αφού οι  $h_i(\mathbf{x})$  είναι κυρτές, αποδεικνύεται πως οι  $F_{n+i}(\mathbf{x}, \mathbf{y}) = h_i(\mathbf{x}) - y_i$  ,  $i = 1 \dots n$  είναι επίσης κυρτές.

Το δεύτερο βήμα είναι να γραφεί το πρόβλημα ως πρόβλημα βελτιστοποίησης. Αυτό γίνεται με τον παρακάτω τρόπο: έστω η συνάρτηση

$$L : \mathbb{R}^{2n} \rightarrow \mathbb{R}, L(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{2n} F_i(\mathbf{x}, \mathbf{y})$$

και το σύνολο  $G = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2n} : F_i(\mathbf{x}, \mathbf{y}) \leq 0, i = 1 \dots 2n\}$ . Τότε τα ζεύγη  $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2n}$  που ικανοποιούν το σύστημα εξισώσεων, ικανοποιούν και το πρόβλημα:

$$\max_{(\mathbf{x}, \mathbf{y})} \{L(\mathbf{x}, \mathbf{y}) : F_i(\mathbf{x}, \mathbf{y}) \leq 0, i = 1 \dots 2n\}$$

Συνεπώς, τα ζεύγη αυτά θα ικανοποιούν και το πρόβλημα :

$$\min_{(\mathbf{x}, \mathbf{y})} \{-L(\mathbf{x}, \mathbf{y}) : F_i(\mathbf{x}, \mathbf{y}) \leq 0, i = 1 \dots 2n\}$$

Το τελευταίο είναι ένα πρόβλημα κοίλης ελαχιστοποίησης. Αυτό ισχύει γιατί εφόσον η συνάρτηση  $L$  είναι κυρτή συνάρτηση (ως άθροισμα κυρτών συναρτήσεων) , τότε η  $-L$  είναι κοίλη συνάρτηση, ενώ οι συναρτήσεις των ανισωτικών περιορισμών είναι κυρτές (συνεπώς και το επιτρεπτό σύνολο θα είναι κυρτό). Έστω πως η λύση του προβλήματος κοίλης ελαχιστοποίησης είναι η  $(\mathbf{x}^*, \mathbf{y}^*)$ . Λόγω των περιορισμών θα έχουμε:  $F_i(\mathbf{x}^*, \mathbf{y}^*) \leq 0$  . Εάν επομένως ισχύει:  $L(\mathbf{x}^*, \mathbf{y}^*) = \sum_{i=1}^{2n} F_i(\mathbf{x}^*, \mathbf{y}^*) = 0$ , θα έχουμε:

$$F_i(\mathbf{x}^*, \mathbf{y}^*) = 0, i = 1 \dots 2n$$

Άρα πράγματι η λύση του προβλήματος κοίλης ελαχιστοποίησης θα είναι και λύση του συστήματος εξισώσεων.

## 2.2 Αλγόριθμος

Το τελικό πρόβλημα κοίλης ελαχιστοποίησης ( $P$ ) με κυρτούς περιορισμούς, δηλαδή το:

$$\min_{\mathbf{x}} \{f(\mathbf{x}): g_i(\mathbf{x}) \leq 0, i = 1 \dots n\}$$

όπου η  $f$  είναι κοίλη και οι  $g_i$  είναι κυρτές, επιλύεται με τον αλγόριθμο της Karla L. Hoffman [9]. Ο αλγόριθμος προϋποθέτει τη συνέχεια των κλίσεων των  $g_i$  και είναι εγγυημένη η σύγκλιση σε ολικό ελάχιστο. Ας υποθέσουμε πως υπάρχει σημείο  $\mathbf{p}$  που βρίσκεται αυστηρά στο εσωτερικό του επιθυμητού συνόλου, δηλαδή:  $\mathbf{p} \in \{\mathbf{x}: g_i(\mathbf{x}) < 0, i = 1 \dots n\}$ . Έστω το σύνολο  $S = \{\mathbf{x}: g_i(\mathbf{x}) \leq 0, i = 1 \dots n\}$ . Ας υποθέσουμε επίσης πως το σύνολο  $S$  είναι μη κενό και συμπαγές. Επιλέγουμε αριθμό  $\varepsilon > 0$  αρκετά μικρό. Με αυτές τις παραδοχές, τα βήματα του αλγορίθμου είναι τα παρακάτω:

- *Βήμα 0*  
Βρες ένα σημείο  $\mathbf{p}$  αυστηρά στο εσωτερικό του  $S$ , δηλαδή  $\mathbf{p} \in \{\mathbf{x}: g_i(\mathbf{x}) < 0, i = 1 \dots n\}$ .  
Θέσε  $u^0 = f(\mathbf{p})$  (το καλύτερο άνω φράγμα μέχρι στιγμής).  
Βρες ένα πολύεδρο  $S^0 \supset S$  έτσι ώστε όλες οι κορυφές του  $S^0$  να είναι γνωστές, και το οποίο μπορεί να περιγραφεί με τον περιορισμό  $A\mathbf{x} \leq \mathbf{b}$ .  
Θέσε  $V^0$  το σύνολο των κορυφών του  $S^0$ .  
Θέσε  $k = 1$   
Πήγαινε στο βήμα 1.
- *Βήμα k* (Αυτό το βήμα εισάγεται με ένα γνωστό πολύεδρο  $S^{k-1} \supset S$ , γνωστό το σύνολο  $V^{k-1}$  των κορυφών του  $S^{k-1}$ , και ένα γνωστό τρέχον άνω φράγμα  $u^{k-1}$  για το πρόβλημα ( $P$ )).

Λύσε το πρόβλημα ( $P^k$ ):

$$\min_{\mathbf{x}} \{f(\mathbf{x}): \mathbf{x} \in S^{k-1}\}$$

επιλέγοντας το  $\min\{f(\mathbf{v}): \mathbf{v} \in V^{k-1}\}$

Έστω πως η λύση του ( $P^k$ ) είναι η  $\mathbf{x}^k$ . Λύσε το πρόβλημα βελτιστοποίησης μιας διάστασης ( $Q^k$ ):

$$\min_{\lambda} \{\lambda: 0 \leq \lambda \leq 1, \mathbf{x}^k + \lambda(\mathbf{p} - \mathbf{x}^k) \in S\}$$

Έστω πως η λύση του προηγούμενου είναι η  $\lambda_k$ . Αν  $\lambda_k = 0$ , τότε  $\mathbf{x}^k \in S$  και επομένως είναι λύση του προβλήματος. Σταμάτα.

Αν  $\lambda_k \neq 0$ , θέσε  $\mathbf{z}^k = \mathbf{x}^k + \lambda_k(\mathbf{p} - \mathbf{x}^k)$ . Εάν  $f(\mathbf{z}^k) < u^{k-1}$  θέσε  $u^k = f(\mathbf{z}^k)$ . Διαφορετικά θέσε  $u^k = u^{k-1}$ . Εάν  $\|\mathbf{z}^k - \mathbf{x}^k\| < \varepsilon$  (ή εναλλακτικά εάν  $u^k - f(\mathbf{x}^k) < \varepsilon$ ), τότε σταμάτα. Το σημείο  $\mathbf{x}^k$  είναι η λύση. Αν όχι, τότε βρες όλους τους ενεργούς περιορισμούς στο  $\mathbf{z}^k$  και όρισε:

$$J^k = \{i: g_i(\mathbf{z}^k) = 0, i = 1 \dots n\}$$

Επίλεξε οποιοδήποτε  $j^k \in J^k$ . Στους περιορισμούς του  $S^{k-1}$  πρόσθεσε τον περιορισμό:

$$G_{j^k}^k(\mathbf{x}) = g_{j^k}(\mathbf{z}^k) + [\nabla g_{j^k}(\mathbf{z}^k)]^T (\mathbf{x} - \mathbf{z}^k) \leq 0$$

Στην παραπάνω σχέση, η ισότητα των δύο μελών αναπαριστά ένα εφαπτόμενο στο  $S$  υπερεπίπεδο, το οποίο διαχωρίζει το σημείο  $\mathbf{x}^k$  από το  $S^k$  που θα οριστεί μετά. Θέσε:

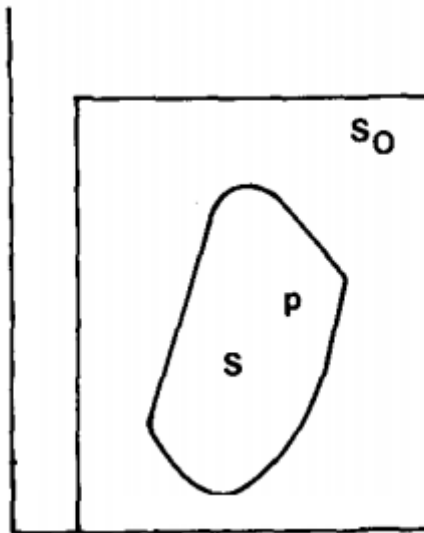
$$S^k = S^{k-1} \cap \{\mathbf{x}: G_{j^k}^k(\mathbf{x}) \leq 0\}$$

$V^k$ : ακραία σημεία του  $S^k$

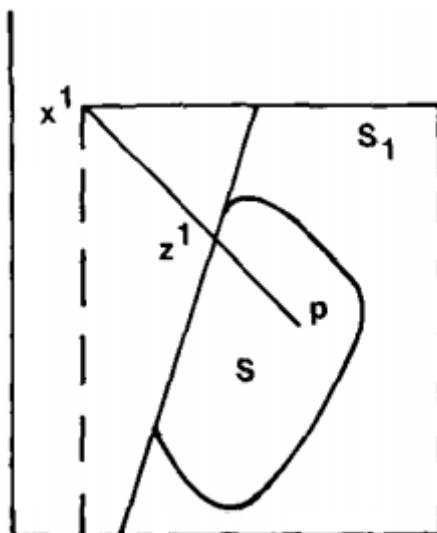
Θέσε  $k = k + 1$ .

Πήγαινε στο βήμα  $k$ .

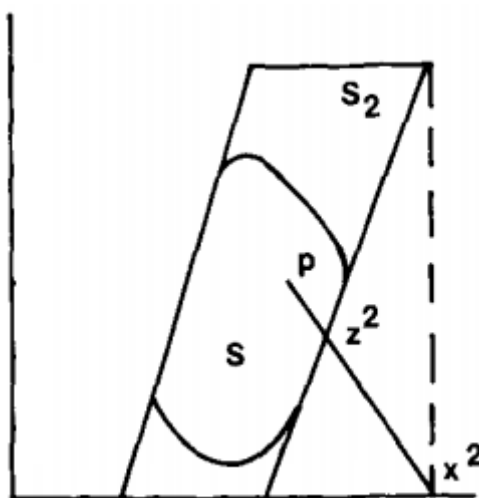
Η δημιουργία των διαδοχικών πολυέδρων γίνεται ευκολότερα κατανοητή εποπτικά στα παρακάτω σχήματα:



Σχήμα 2.2.1: Ξεκινάμε με ένα αρχικό πολύεδρο  $S_0$  που περικλείει το επιθυμητό σύνολο  $S$ , και με ένα σημείο  $\mathbf{p}$  αυστηρά στο εσωτερικό του  $S$



Σχήμα 2.2.2: Βρίσκουμε το σημείο ελαχίστου  $x^1$  της αντικειμενικής συνάρτησης πάνω στο  $S_0$  το οποίο είναι μια από τις κορυφές του  $S_0$  και βρίσκουμε το σημείο  $z^1$  το οποίο βρίσκεται στο σύνορο του  $S$  και στην ευθεία που ενώνει τα  $p$  και  $x^1$ . Η εφαπτόμενη του  $S$  στο  $z^1$  προσθέτει έναν επιπλέον γραμμικό περιορισμό, και έτσι προκύπτει το νέο πολύεδρο  $S_1$ .



Σχήμα 2.2.3: Βρίσκουμε το σημείο ελαχίστου  $x^2$  της αντικειμενικής συνάρτησης πάνω στο  $S_1$  το οποίο είναι μια από τις κορυφές του  $S_1$  και βρίσκουμε το σημείο  $z^2$  το οποίο βρίσκεται στο σύνορο του  $S$  και στην ευθεία που ενώνει τα  $p$  και  $x^2$ . Η εφαπτόμενη του  $S$  στο  $z^2$  προσθέτει έναν επιπλέον γραμμικό περιορισμό, και έτσι προκύπτει το νέο πολύεδρο  $S_2$ . Η διαδικασία αυτή επαναλαμβάνεται με όλο και πιο «σφιχτά» πολύεδρα, μέχρι τη σύγκλιση σε σημείο ελαχίστου της αντικειμενικής συνάρτησης πάνω στο  $S$ .

## 2.3 Προσαρμογή του προβλήματος

Υπενθυμίζεται πως, όπως αναφέρεται στην παράγραφο 2.1 , το πρόβλημα των εξισώσεων ανάγεται στο πρόβλημα:

$$\min_{(x,y)} \{-L(x,y) : F_i(x,y) \leq 0, i = 1 \dots 2n\}$$

όπου

$$L(x,y) = \sum_{i=1}^{2n} F_i(x,y)$$

και

$$\begin{cases} F_i(x,y) = g_i(x) - y_i, i = 1 \dots n \\ F_{n+i}(x,y) = h_i(x) - y_i, i = 1 \dots n \end{cases}$$

Στην προηγούμενη παράγραφο αναφέρθηκε πως το επιθυμητό σύνολο  $S$  υποτίθεται πως είναι κλειστό και φραγμένο. Στο πρόβλημά μας όμως, το  $S$  ορίζεται ως εξής:

$$S = \{(x,y) : g_i(x) - y_i \leq 0, h_i(x) - y_i, i = 1 \dots 2n\}$$

Εδώ το  $S$  δεν είναι φραγμένο. Αυτό μπορεί να φανεί ως εξής: έστω ένα  $x_0$ . Τότε  $\forall y > y_0$  όπου  $y_0 = \max_{i=1 \dots n} [\max \{g_i(x_0), h_i(x_0)\}]$ , το ζευγάρι  $(x_0, y)$  είναι τέτοιο ώστε να ικανοποιούνται οι συνθήκες που ορίζουν το  $S$ . Συνεπώς δεν είναι φραγμένο σύνολο. Το πρόβλημα αυτό αντιμετωπίστηκε με επιβολή άνω και κάτω ορίων στις μεταβλητές. Συγκεκριμένα, ακολουθήθηκε η εξής διαδικασία: για τις μεταβλητές  $x_i$  επιβλήθηκαν αυθαίρετα διαστήματα  $[l_i, u_i]$  στα οποία θα ανήκουν οι τιμές των  $x_i$ . Έστω πως αυτό το σύνολο καλείται  $X$ , δηλαδή:

$$X = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i\}$$

Τα αντίστοιχα κάτω φράγματα για τις μεταβλητές  $y_i$ , έστω  $ly_i$ , βρέθηκαν ως εξής:

$$ly_i = \min_{x,y_i} \{y_i : x \in X, g_i(x) - y_i \leq 0, h_i(x) - y_i \leq 0\}$$

Τα αντίστοιχα άνω φράγματα για τις μεταβλητές  $y_i$ , έστω  $uy_i$ , βρέθηκαν ως εξής:

$$uy_i = \max_x [\max\{g_i(x) : x \in X\}, \max\{h_i(x) : x \in X\}]$$

Σημειώνεται σε αυτό το σημείο πως για την εύρεση των άνω ορίων των μεταβλητών  $y_i$ , δεν είναι απαραίτητο να λυθεί πρόβλημα μεγιστοποίησης στο  $X$  αλλά απλά εύρεση των μέγιστων τιμών των  $g_i, h_i$  με έλεγχο των τιμών τους στις κορυφές του συνόλου  $X$ . Αυτό είναι αληθές λόγω της κυρτότητας αυτών των συναρτήσεων. Ας



συμβολίσουμε με  $Y$  το σύνολο όπου θα ανήκουν οι  $y_i$ . Έχουμε ορίσει επομένως ένα σύνολο  $S_0 = X \times Y$  το οποίο είναι πολύεδρο που περικλείει το  $S$ , όπου το  $S$  πλέον δεν είναι το σύνολο όλων  $(\mathbf{x}, \mathbf{y})$  που ικανοποιούν τους περιορισμούς  $F_i(\mathbf{x}, \mathbf{y}) \leq 0$ , αλλά το σύνολο:  $S = \{(\mathbf{x}, \mathbf{y}): \mathbf{x} \in X, F_i(\mathbf{x}, \mathbf{y}) \leq 0, i = 1 \dots 2n\}$ . Συνεπώς, μπορεί να εφαρμοστεί ο αλγόριθμος, όμως θα βρει μόνο λύσεις του προβλήματος που ανήκουν στο σύνολο  $X$ .

Εκτός από την υπόθεση πως το σύνολο  $S$  είναι μη κενό και συμπαγές, υπάρχει επιπλέον η υπόθεση ύπαρξης ενός σημείου  $\mathbf{p}$  αυστηρά στο εσωτερικό του  $S$ . Το εσωτερικό σημείο βρέθηκε επιλύοντας το πρόβλημα:

$$\min_{\mathbf{x}, \mathbf{y}, \varepsilon} \{\varepsilon: g_i(\mathbf{x}) - y_i \leq \varepsilon, h_i(\mathbf{x}) - y_i \leq \varepsilon, \mathbf{x} \in X, \mathbf{y} \in Y\}$$

Έστω πως η λύση του παραπάνω είναι η  $(\mathbf{x}^*, \mathbf{y}^*, \varepsilon^*)$ . Τότε το εσωτερικό σημείο  $\mathbf{p}$  θα είναι το  $\mathbf{p} = (\mathbf{x}^*, \mathbf{y}^*)$ , το οποίο όμως δεν είναι απαραίτητα αυστηρά στο εσωτερικό του  $S$ . Για το λόγο αυτό μετά την εύρεση του  $\mathbf{p}$ , στην υλοποίηση του αλγορίθμου διευρύνονται τα άνω και κάτω όρια του συνόλου  $X$  συνεπώς και του  $Y$ , ώστε το  $\mathbf{p}$  να είναι αυστηρά στο εσωτερικό του  $S$ .

## Κεφάλαιο 3

### Παραδείγματα - Συμπεράσματα

#### 3.1

Το πρώτο παράδειγμα συστήματος όπου εφαρμόστηκε ο αλγόριθμος είναι το παρακάτω σύστημα δύο εξισώσεων:

$$x_1^2 - 2x_2^2 = 0$$

$$4x_1^2 - x_2^2 = 5$$

Το πρώτο βήμα είναι να γραφούν οι εξισώσεις ως εξισώσεις με D.C. όρους :

$$\begin{cases} x_1^2 - 2x_2^2 = 0 \\ 4x_1^2 - x_2^2 = 5 \end{cases} \Leftrightarrow \Leftrightarrow \begin{cases} 2x_1^2 + x_2^2 - (x_1^2 + 3x_2^2) = 0 \\ 5x_1^2 + x_2^2 - (x_1^2 + 2x_2^2 + 5) = 0 \end{cases}$$

Ορίζοντας τις κυρτές συναρτήσεις:

$$\begin{cases} g_1(\mathbf{x}) = 2x_1^2 + x_2^2 \\ g_2(\mathbf{x}) = 5x_1^2 + x_2^2 \\ h_1(\mathbf{x}) = x_1^2 + 3x_2^2 \\ h_2(\mathbf{x}) = x_1^2 + 2x_2^2 + 5 \end{cases}$$

το σύστημα γράφεται:

$$\begin{cases} g_1(\mathbf{x}) - h_1(\mathbf{x}) = 0 \\ g_2(\mathbf{x}) - h_2(\mathbf{x}) = 0 \end{cases}$$

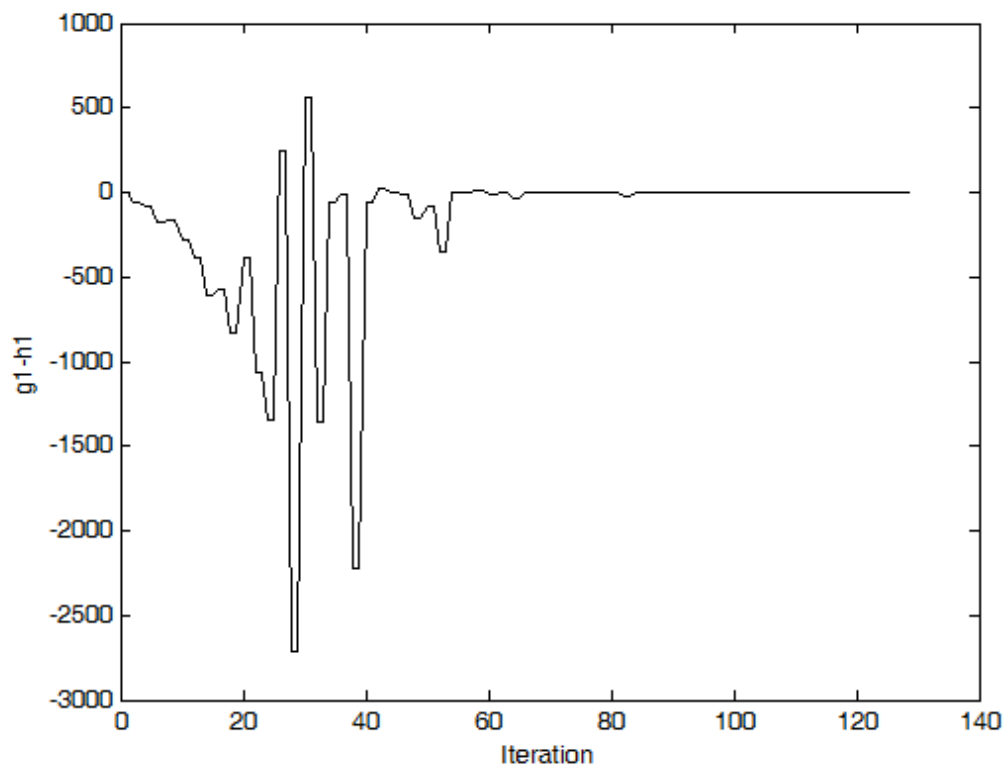
Ορίζουμε τις συναρτήσεις τεσσάρων μεταβλητών:

$$\begin{cases} F_1(\mathbf{x}, \mathbf{y}) = g_1(\mathbf{x}) - y_1 \\ F_2(\mathbf{x}, \mathbf{y}) = g_2(\mathbf{x}) - y_2 \\ F_3(\mathbf{x}, \mathbf{y}) = h_1(\mathbf{x}) - y_1 \\ F_4(\mathbf{x}, \mathbf{y}) = h_2(\mathbf{x}) - y_2 \end{cases}$$

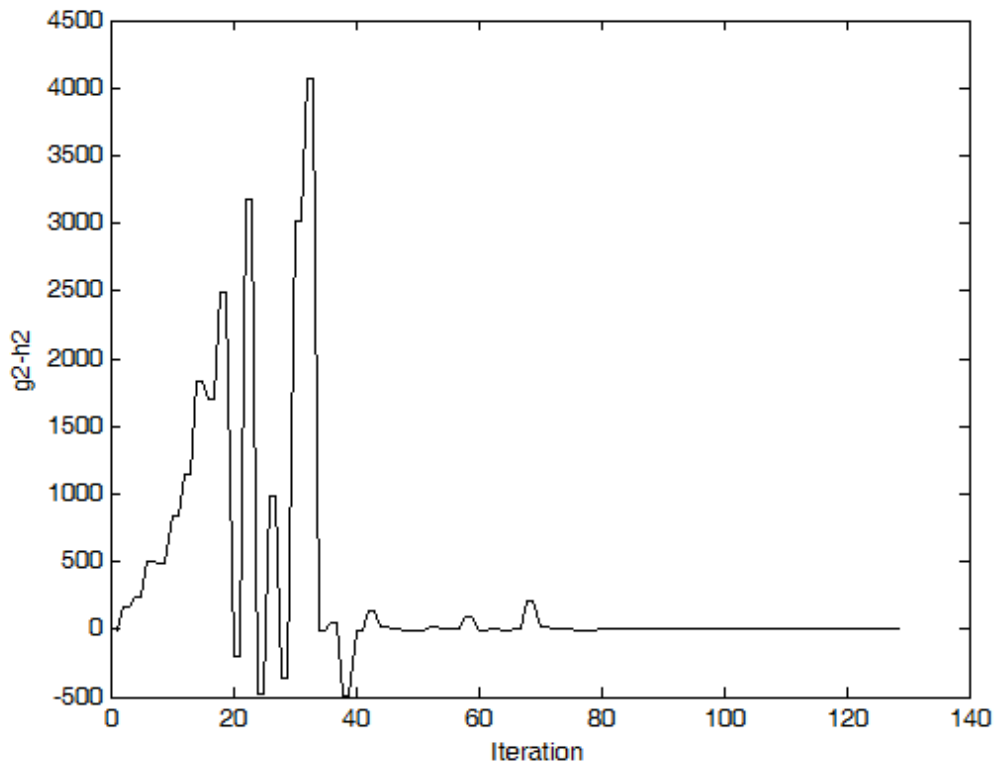
Στο συγκεκριμένο παράδειγμα το σύνολο  $X$  ορίστηκε το  $[-50,50] \times [-50,50]$ . Το αντίστοιχο σύνολο  $Y$  είναι το  $[0,10000] \times [5,15000]$ . Με βάση αυτά το πρόβλημα ελαχιστοποίησης είναι το :

$$\min_{(x,y) \in X \times Y} \{2(y_1 + y_2) - g_1(x) - g_2(x) - h_1(x) - h_2(x) , F_i(x, y) \leq 0, i = 1 \dots 4\}$$

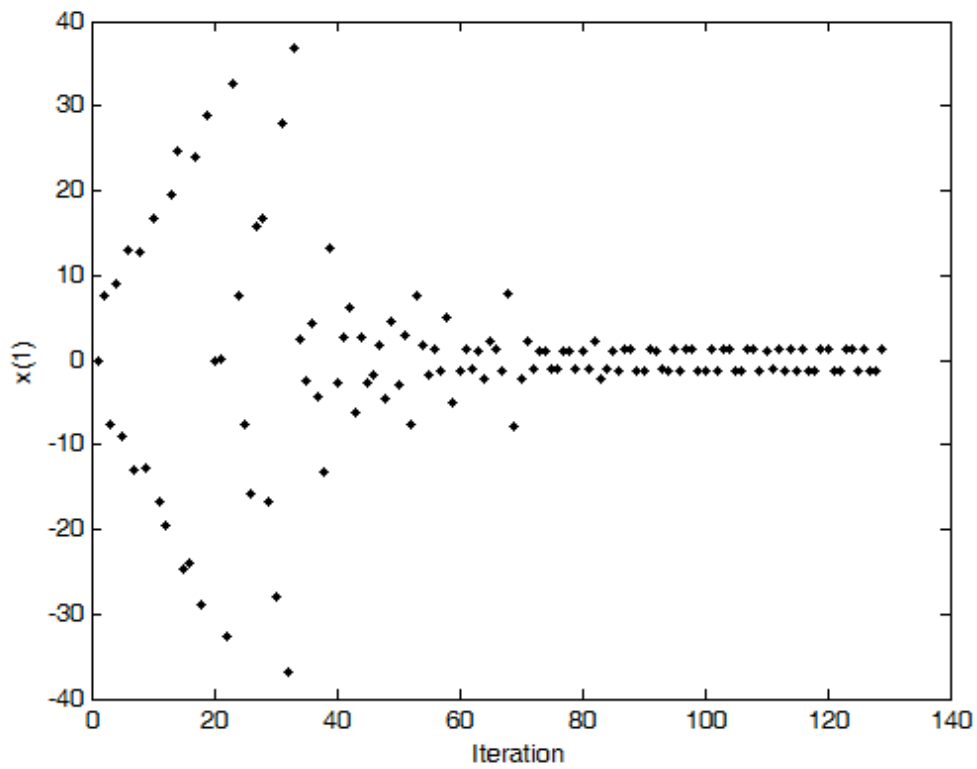
Τα αποτελέσματα είναι τα ακόλουθα:



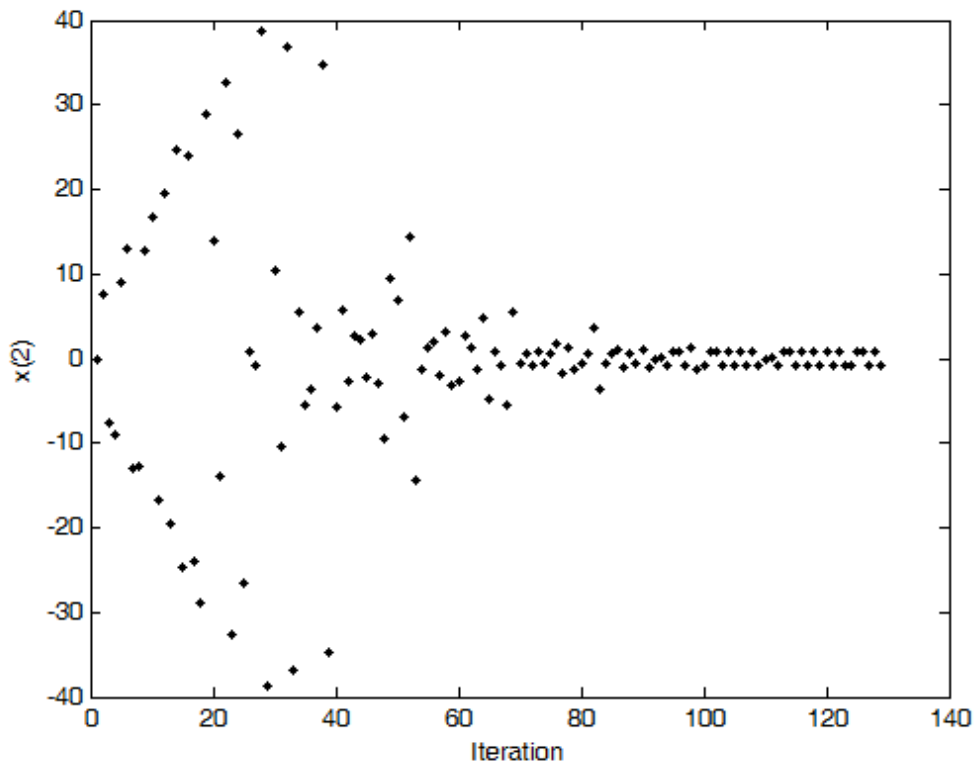
Σχήμα 3.1.1 Οι τιμές της πρώτης εξίσωσης του συστήματος



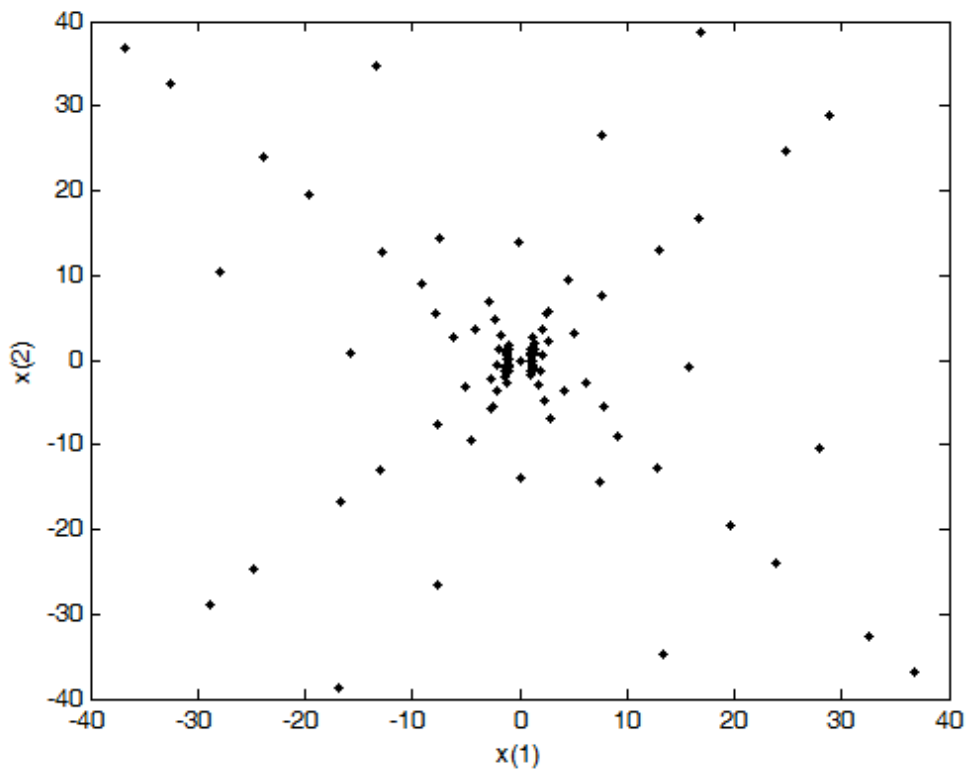
Σχήμα 3.1.2 Οι τιμές της δεύτερης εξίσωσης του συστήματος



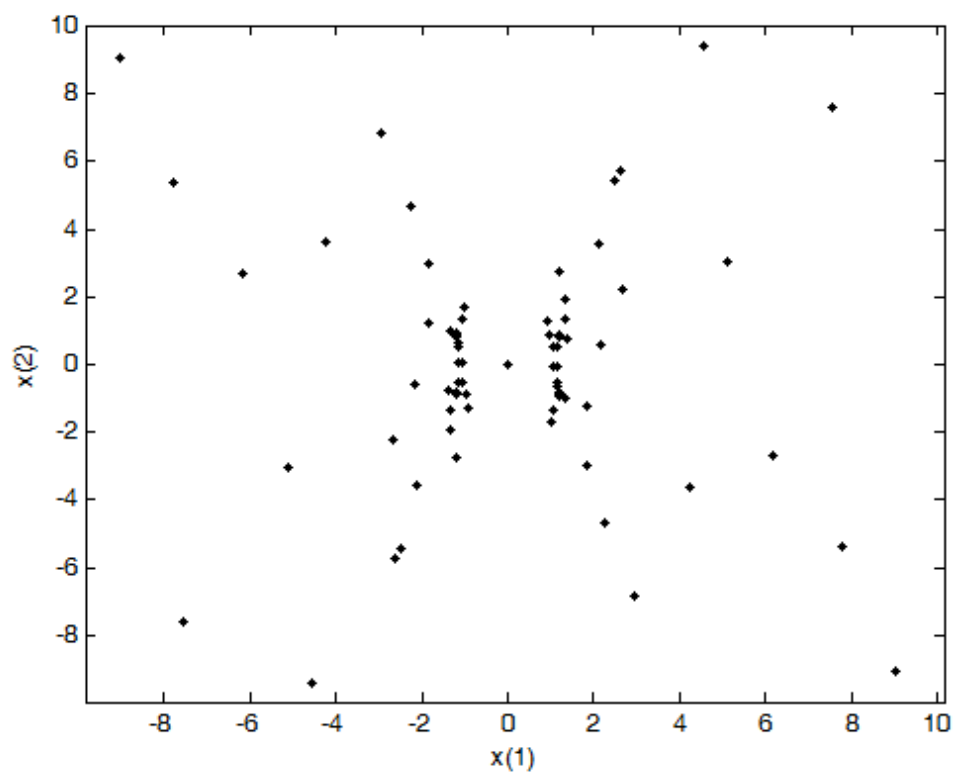
Σχήμα 3.1.3 Οι τιμές της μεταβλητής  $x_1$



Σχήμα 3.1.4 Οι τιμές της μεταβλητής  $x_2$



Σχήμα 3.1.5 Οι μεταβλητές  $x_1$  και  $x_2$



Σχήμα 3.1.6 Οι μεταβλητές  $x_1$  και  $x_2$  κατά τις τελευταίες επαναλήψεις

Επανάληψη	$x_1$	$x_2$	$g_1 - h_1$	$g_2 - h_2$
1	0,00000	0,00000	0,00000	-5,00000
2	7,57616	7,57620	-57,39928	167,19420
3	-7,57619	-7,57615	-57,39758	167,19640
4	9,04977	-9,04991	-81,90357	240,69231
5	-9,04998	9,04984	-81,89693	240,70918
6	13,06550	13,06553	-170,70875	507,12077
7	-13,06565	-13,06562	-170,70963	507,13437
8	12,85461	-12,85474	-165,24778	490,71964
9	-12,85464	12,85451	-165,23505	490,72886
10	16,69705	16,69707	-278,79318	831,37325
11	-16,69699	-16,69697	-278,78775	831,36955
12	-19,57136	19,57125	-383,02977	1144,11889
13	19,57119	-19,57130	-383,03986	1144,09002
14	24,75628	24,75630	-612,87524	1833,61864
15	-24,75639	-24,75637	-612,87672	1833,63724
16	-23,88092	23,88083	-570,28950	1705,89947

ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

17	23,88097	-23,88106	-570,30962	1705,89788
18	-28,81101	-28,81099	-830,07206	2485,22311
19	28,81115	28,81117	-830,08446	2485,24646
20	-0,16304	13,96290	-389,89859	-199,85625
21	0,16285	-13,96292	-389,89995	-199,85715
22	-32,55376	32,55370	-1059,73934	3174,24622
23	32,55364	-32,55370	-1059,74763	3174,21470
24	7,62096	26,44426	-1340,51846	-471,98255
25	-7,62109	-26,44426	-1340,51670	-471,97462
26	-15,75823	0,76080	247,16421	987,70857
27	15,75810	-0,76087	247,16001	987,69251
28	16,83419	38,68863	-2710,23024	-368,25066
29	-16,83424	-38,68865	-2710,23218	-368,24578
30	-27,99950	10,51208	562,96410	3020,38386
31	27,99943	-10,51212	562,95838	3020,36668
32	-36,83818	36,83814	-1357,04489	4066,15873
33	36,83829	-36,83833	-1357,06626	4066,17470
34	2,50086	5,43562	-52,83766	-9,52873
35	-2,50090	-5,43560	-52,83694	-9,52771
36	4,21797	-3,63150	-8,58431	52,97740
37	-4,21801	3,63146	-8,58345	52,97881
38	-13,27877	34,62904	-2222,01471	-498,86716
39	13,27863	-34,62900	-2222,01325	-498,87977
40	-2,62882	-5,73033	-58,76275	-10,19391
41	2,62877	5,73033	-58,76284	-10,19487
42	6,17398	-2,71428	23,38338	140,10473
43	-6,17405	2,71419	23,38527	140,10892
44	2,69421	2,23250	-2,70931	19,05106
45	-2,69423	-2,23250	-2,70922	19,05136
46	-1,83817	2,97391	-14,30937	-0,32859
47	1,83814	-2,97392	-14,30959	-0,32910
48	-4,55404	-9,40499	-156,16852	-10,49682
49	4,55401	9,40496	-156,16764	-10,49720
50	-2,93348	6,83783	-84,90649	-17,33475
51	2,93342	-6,83778	-84,90547	-17,33544
52	-7,54006	14,32138	-353,35118	17,30833
53	7,54001	-14,32143	-353,35506	17,30361
54	1,86268	-1,23441	0,42202	7,35455
55	-1,86269	1,23440	0,42211	7,35471
56	1,34826	1,90105	-5,41016	-1,34276
57	-1,34827	-1,90105	-5,41010	-1,34259
58	5,12017	3,05936	7,49681	90,50502
59	-5,12020	-3,05934	7,49723	90,50603
60	-1,18835	-2,76211	-13,84632	-6,98052

ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

61	1,18834	2,76210	-13,84621	-6,98059
62	-1,08181	1,34728	-2,46002	-2,13388
63	1,08181	-1,34729	-2,46007	-2,13397
64	-2,24039	4,68387	-38,85787	-6,86121
65	2,24037	-4,68385	-38,85768	-6,86143
66	1,36811	0,77271	0,67758	1,88986
67	-1,36812	-0,77271	0,67758	1,88989
68	7,78926	-5,38597	2,65529	208,68169
69	-7,78928	5,38593	2,65651	208,68339
70	-2,17907	-0,56750	4,10422	13,67128
71	2,17906	0,56750	4,10416	13,67107
72	-0,98631	-0,86775	-0,53318	-1,86174
73	0,98631	0,86775	-0,53318	-1,86175
74	1,14257	-0,50674	0,79189	-0,03493
75	-1,14257	0,50674	0,79191	-0,03491
76	-1,03352	1,70522	-4,74740	-3,63516
77	1,03351	-1,70523	-4,74749	-3,63528
78	0,94272	1,30000	-2,49129	-3,13510
79	-0,94273	-1,30002	-2,49136	-3,13508
80	1,16080	-0,65190	0,49751	-0,03516
81	-1,16080	0,65189	0,49753	-0,03514
82	2,13387	3,56201	-20,82249	0,52563
83	-2,13388	-3,56201	-20,82234	0,52582
84	-1,05810	-0,50024	0,61909	-0,77197
85	1,05810	0,50024	0,61909	-0,77194
86	-1,33305	0,98205	-0,15184	1,14364
87	1,33305	-0,98206	-0,15186	1,14360
88	1,17150	0,54762	0,77262	0,18974
89	-1,17150	-0,54763	0,77262	0,18975
90	-1,20268	0,93884	-0,31639	-0,09562
91	1,20269	-0,93884	-0,31640	-0,09562
92	1,14754	-0,07384	1,30594	0,26192
93	-1,14754	0,07384	1,30594	0,26193
94	-1,18505	-0,82760	0,03452	-0,06752
95	1,18505	0,82759	0,03452	-0,06752
96	-1,19721	0,82573	0,06965	0,05138
97	1,19721	-0,82573	0,06965	0,05138
98	1,32628	1,36263	-1,95452	0,17926
99	-1,32628	-1,36263	-1,95451	0,17928
100	-1,20814	-0,84284	0,03883	0,12799
101	1,20814	0,84284	0,03883	0,12798
102	-1,18462	0,81984	0,05906	-0,05884
103	1,18462	-0,81984	0,05906	-0,05883
104	1,19296	0,83283	0,03594	-0,00096



105	-1,19296	-0,83283	0,03594	-0,00096
106	-1,18737	0,84693	-0,02471	-0,07788
107	1,18737	-0,84693	-0,02472	-0,07788
108	1,19756	0,85972	-0,04409	-0,00251
109	-1,19756	-0,85972	-0,04409	-0,00251
110	1,04543	-0,07847	1,08061	-0,63446
111	-1,04543	0,07847	1,08061	-0,63445
112	1,19649	-0,85228	-0,02115	0,00001
113	-1,19649	0,85228	-0,02115	0,00001
114	1,19527	0,84532	-0,00046	0,00008
115	-1,19527	-0,84532	-0,00046	0,00008
116	1,19517	0,84525	-0,00047	-0,00074
117	-1,19517	-0,84525	-0,00047	-0,00074
118	-1,19522	0,84495	0,00069	0,00030
119	1,19522	-0,84495	0,00069	0,00030
120	1,19525	0,84531	-0,00046	-0,00002
121	-1,19525	-0,84531	-0,00046	-0,00002
122	-1,19528	0,84523	-0,00015	0,00032
123	1,19528	-0,84523	-0,00015	0,00032
124	1,19523	-0,84520	-0,00015	-0,00007
125	-1,19523	0,84520	-0,00015	-0,00007
126	1,19522	0,84510	0,00016	0,00000
127	-1,19522	-0,84510	0,00016	0,00000
128	-1,19522	0,84515	0,00000	-0,00007
129	1,19522	-0,84515	0,00000	-0,00007

Πίνακας 3.1.1 Οι τιμές των μεταβλητών και των D.C. συναρτήσεων

Παρατηρείται πως η ακολουθία σημείων που παράγεται από τον αλγόριθμο καταλήγει σε τέσσερα σημεία, τα  $(1.19522, 0.84510)$ ,  $(-1.19522, -0.84510)$ ,  $(-1.19522, 0.84510)$ ,  $(1.19522, -0.84510)$  (φαίνονται μόνο οι δύο πρώτες συντεταγμένες, που είναι οι άγνωστοι των εξισώσεων). Οι ακριβείς λύσεις του συστήματος είναι τα ζεύγη  $(\sqrt{\frac{10}{7}}, \sqrt{\frac{5}{7}})$ ,  $(-\sqrt{\frac{10}{7}}, \sqrt{\frac{5}{7}})$ ,  $(\sqrt{\frac{10}{7}}, -\sqrt{\frac{5}{7}})$ ,  $(-\sqrt{\frac{10}{7}}, -\sqrt{\frac{5}{7}})$ .

## 3.2

Το επόμενο παράδειγμα συστήματος είναι το παρακάτω:

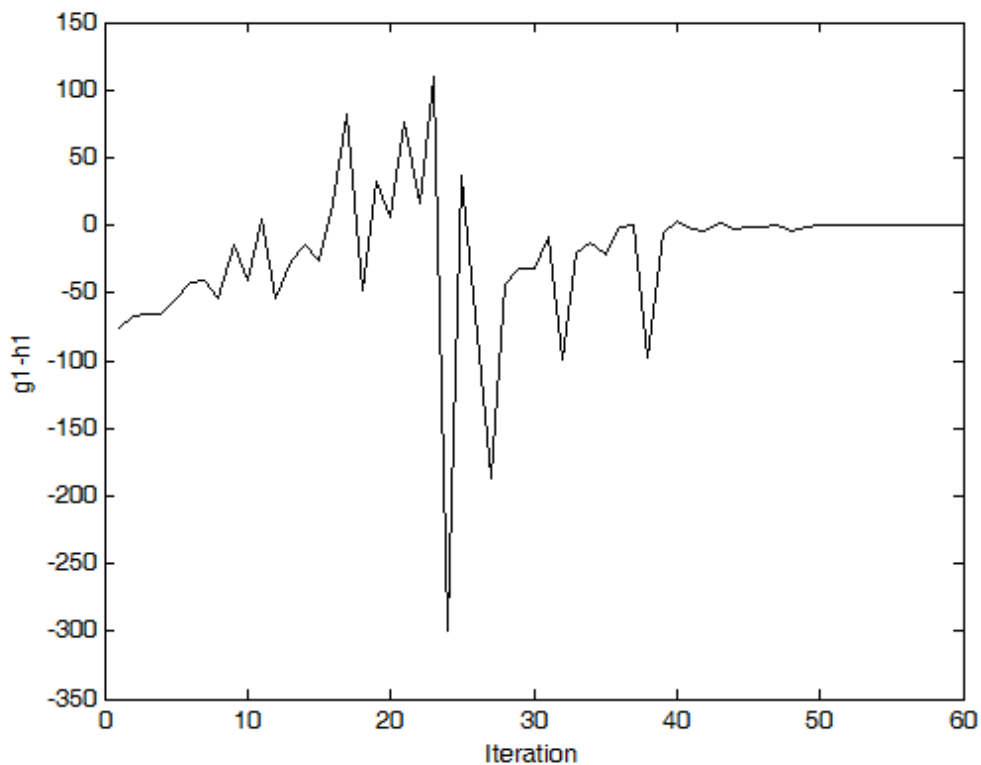
$$\begin{cases} 3x_1^2 - 2x_2^2 = 76.15046 \\ 4x_1 - x_3^2 = 11.0039 \\ 4x_1^2 + 2x_2^2 + 2x_3^2 - 33x_1 + 16x_2 - 24x_3 = -143.6388 \end{cases}$$

Ορίζοντας τις παρακάτω κυρτές συναρτήσεις:

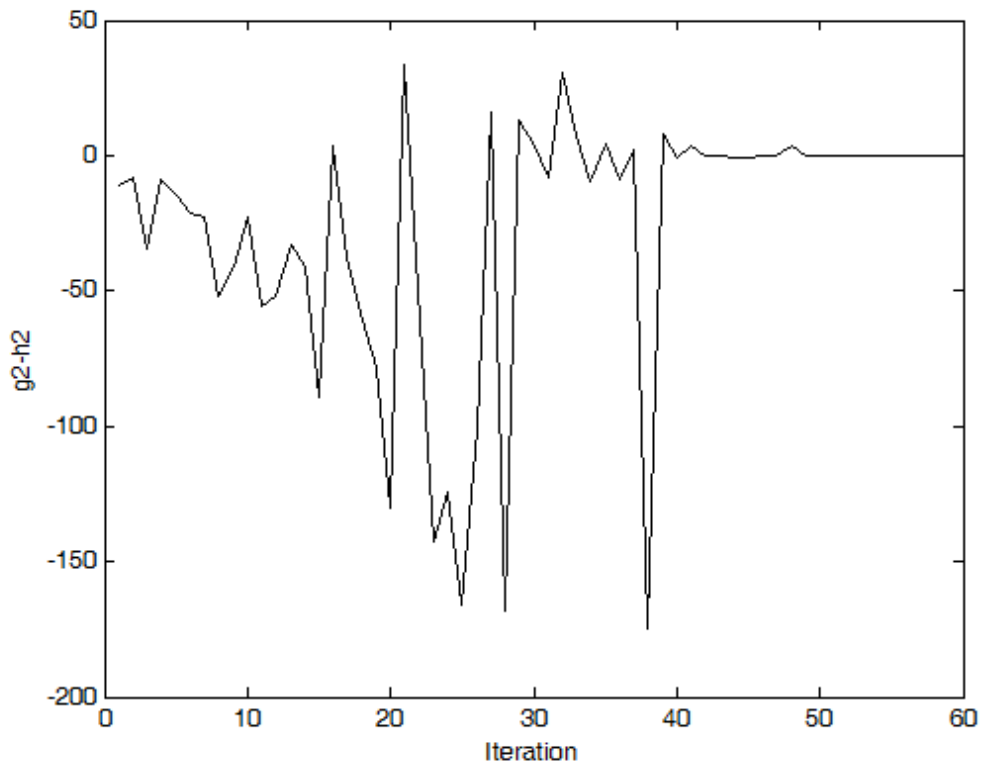
$$\begin{cases} g_1(\mathbf{x}) = 3x_1^2 \\ g_2(\mathbf{x}) = 4x_1 - 11.0039 \\ g_3(\mathbf{x}) = 4x_1^2 + 2x_2^2 + 2x_3^2 \\ h_1(\mathbf{x}) = 2x_2^2 + 76.15046 \\ h_2(\mathbf{x}) = x_3^2 \\ h_3(\mathbf{x}) = 33x_1 - 16x_2 + 24x_3 - 143.6388 \end{cases}$$

έχουμε την επιθυμητή μορφή εξισώσεων με D.C. όρους:

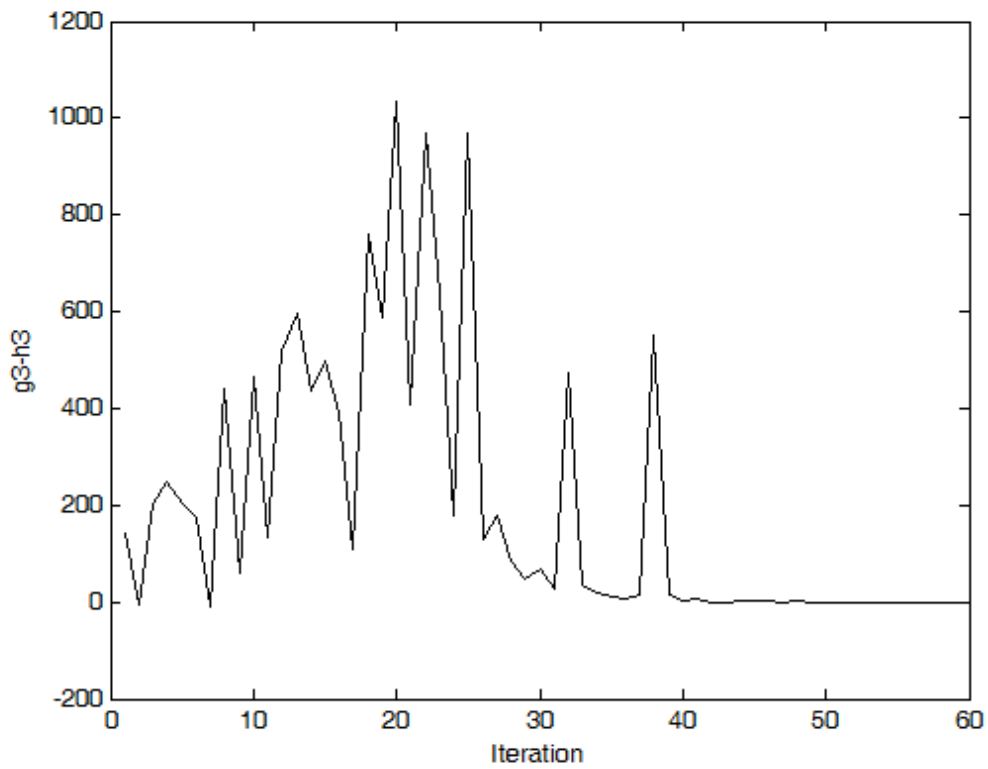
$$\begin{cases} g_1(\mathbf{x}) - h_1(\mathbf{x}) = 0 \\ g_2(\mathbf{x}) - h_2(\mathbf{x}) = 0 \\ g_3(\mathbf{x}) - h_3(\mathbf{x}) = 0 \end{cases}$$



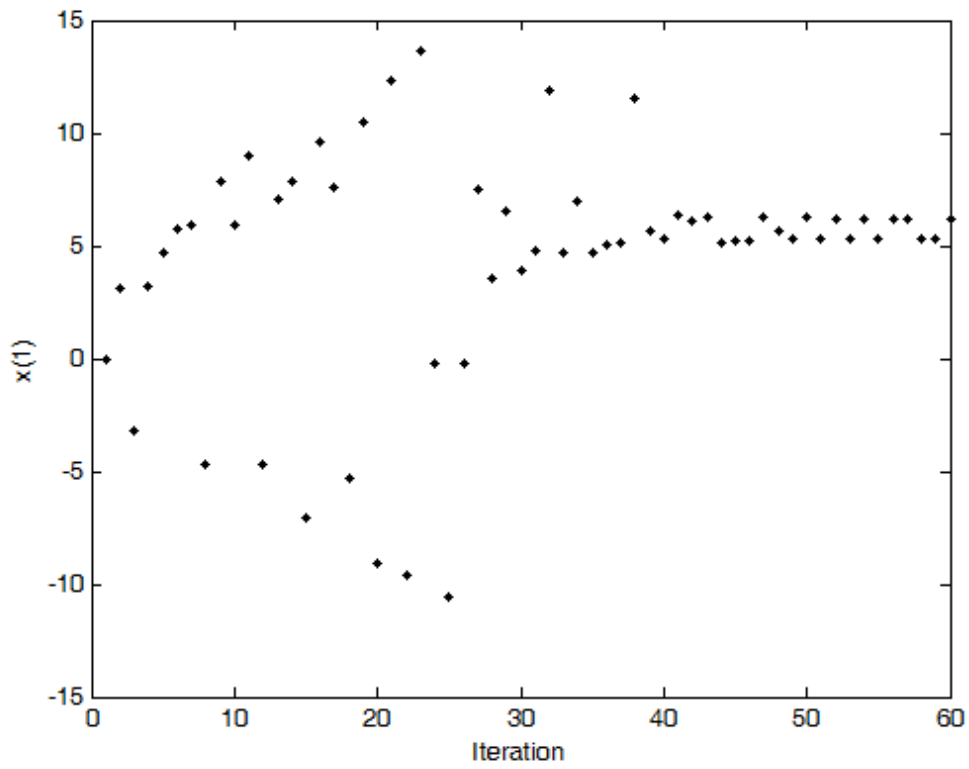
Σχήμα 3.2.1 Οι τιμές της πρώτης εξίσωσης του συστήματος



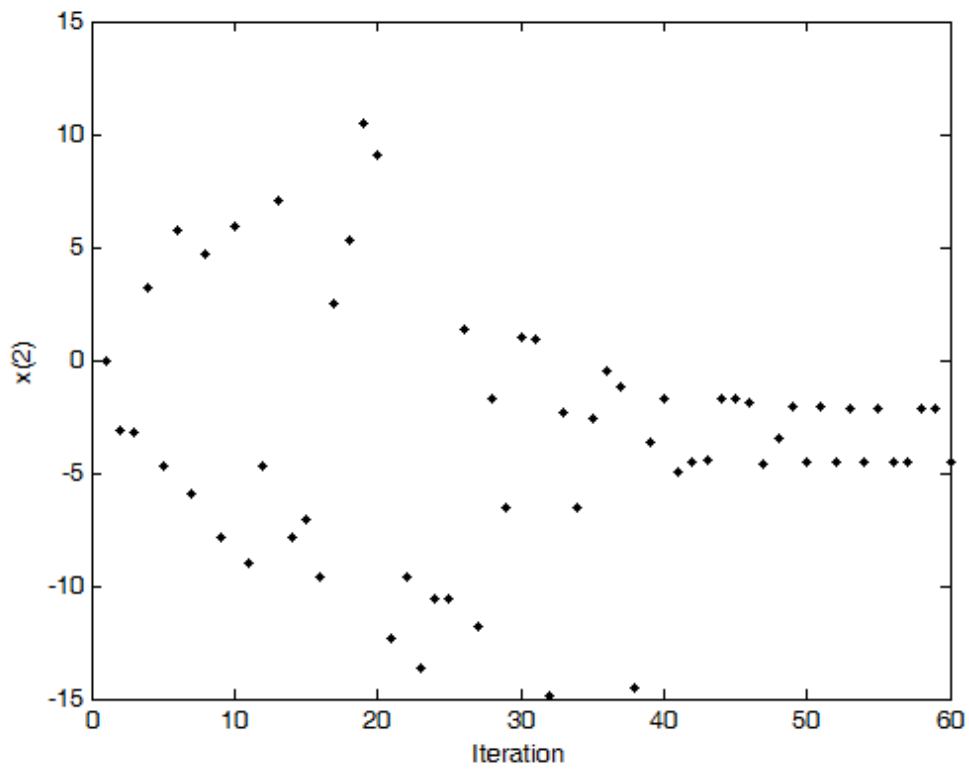
Σχήμα 3.2.2 Οι τιμές της δεύτερης εξίσωσης του συστήματος



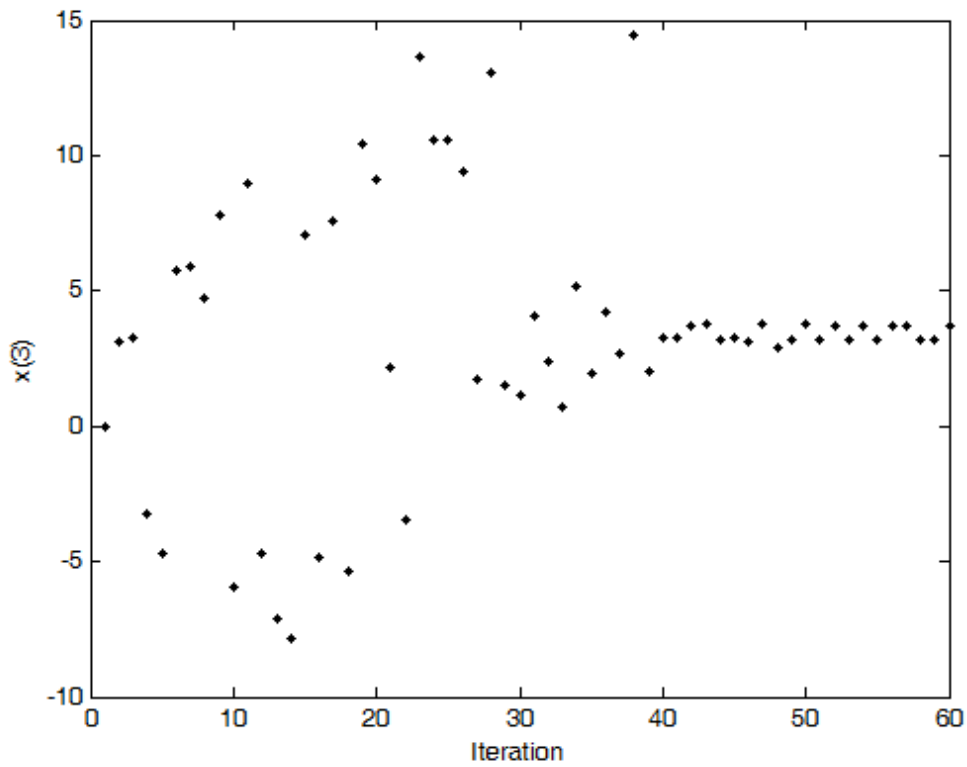
Σχήμα 3.2.3 Οι τιμές της τρίτης εξίσωσης του συστήματος



Σχήμα 3.2.4 Οι τιμές της μεταβλητής  $x_1$



Σχήμα 3.2.5 Οι τιμές της μεταβλητής  $x_2$



Σχήμα 3.2.6 Οι τιμές της μεταβλητής  $x_3$

Επανάληψη	$x_1$	$x_2$	$x_3$	$g_1 - h_1$	$g_2 - h_2$	$g_3 - h_3$
1.	0,00000	0,00000	0,00000	-76,15046	-11,00390	143,63880
2.	3,12372	-3,12370	3,12372	-66,39255	-8,26662	-6,33173
3.	-3,23700	-3,23701	3,23703	-65,67239	-34,43028	204,80507
4.	3,23702	3,23701	-3,23700	-65,67201	-8,53397	250,12332
5.	4,73070	-4,73068	-4,73068	-53,77059	-14,46041	204,40648
6.	5,76916	5,76915	5,76915	-42,86710	-21,21038	173,36836
7.	5,93072	-5,93070	5,93071	-40,97663	-22,45440	-7,91639
8.	-4,73068	4,73070	4,73070	-53,77147	-52,30616	440,94096
9.	7,80777	-7,80776	7,80777	-15,18874	-40,73403	61,36151
10.	5,93071	5,93070	-5,93069	-40,97691	-22,45416	466,53928
11.	8,99242	-8,99241	8,99242	4,71361	-55,89776	134,10071
12.	-4,73068	-4,73069	-4,73068	-53,77131	-52,30599	516,63146
13.	7,08865	7,08865	-7,08863	-25,90124	-32,89803	595,25010
14.	7,80778	-7,80777	-7,80776	-15,18858	-40,73393	436,13454

ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

15.	-7,08864	-7,08865	7,08866	-25,90180	-89,60762	496,00951
16.	9,63862	-9,63861	-4,87212	16,75308	3,81301	393,17046
17.	7,55943	2,51532	7,55942	82,63070	-37,91107	108,51983
18.	-5,33149	5,33151	-5,33150	-47,72601	-60,75474	760,23719
19.	10,44587	10,44586	10,44586	32,96593	-78,33651	588,28704
20.	-9,09229	9,09231	9,09231	6,51885	-130,04314	1032,30528
21.	12,33377	-12,33376	2,19178	75,97188	33,52728	409,02018
22.	-9,59610	-9,59611	-3,48055	15,93452	-61,50255	967,04565
23.	13,63279	-13,63278	13,63279	109,70295	-142,32567	635,26864
24.	-0,24558	-10,58709	10,58710	-300,14264	-124,07293	176,84664
25.	-10,60514	-10,60514	10,60515	36,31826	-165,89370	969,15491
26.	-0,22005	1,40096	9,42247	-79,93059	-100,66699	128,86158
27.	7,50450	-11,82948	1,69900	-187,07095	16,12748	176,85882
28.	3,57342	-1,68235	13,09269	-43,50304	-168,12878	84,14880
29.	6,55389	-6,53419	1,52354	-32,68127	12,89050	48,09601
30.	3,94113	0,96672	1,16846	-31,42201	3,39532	67,73568
31.	4,81342	0,88645	4,05403	-8,21509	-8,18542	28,80031
32.	11,84505	-14,90371	2,36335	-99,47598	30,79086	474,20520
33.	4,68185	-2,28245	0,71057	-20,81038	7,21861	34,67279
34.	7,00808	-6,49928	5,18768	-13,29197	-9,88362	18,63748
35.	4,73601	-2,57278	1,98784	-22,09964	3,98861	9,33834
36.	5,02511	-0,50854	4,23233	-0,91245	-8,81607	5,44696
37.	5,15295	-1,17898	2,68898	0,72814	2,37729	13,64507
38.	11,50482	-14,47563	14,47563	-98,15521	-174,52851	552,57353
39.	5,70083	-3,62892	2,00882	-4,99031	7,76406	13,64369
40.	5,33817	-1,72517	3,30400	3,38538	-0,56760	2,35002
41.	6,35637	-4,91894	3,30245	-3,33187	3,51544	7,73488
42.	6,12522	-4,51247	3,69140	-4,32027	-0,12945	-1,23564
43.	6,24927	-4,45406	3,76153	1,33262	-0,15590	0,06032
44.	5,12284	-1,71219	3,22632	-3,28332	-0,92169	1,41378
45.	5,19463	-1,72896	3,26444	-1,17638	-0,88192	1,43460
46.	5,23565	-1,88992	3,15484	-1,05795	-0,01430	1,60527
47.	6,26785	-4,58478	3,77634	-0,33306	-0,19327	0,51679
48.	5,68582	-3,48999	2,88383	-3,52494	3,42292	1,26221
49.	5,29768	-2,09220	3,19263	-0,70887	-0,00610	0,11902
50.	6,23016	-4,50465	3,75871	-0,28930	-0,21117	-0,14078
51.	5,31933	-2,09909	3,20578	-0,07699	-0,00362	0,12424
52.	6,22652	-4,48182	3,72864	-0,01550	-0,00059	0,02423
53.	5,32659	-2,11805	3,21020	-0,00500	-0,00295	0,00100
54.	6,22502	-4,47802	3,72788	-0,00318	-0,00090	-0,00130
55.	5,32686	-2,11881	3,20991	-0,00302	-0,00001	0,00105
56.	6,22509	-4,47796	3,72792	0,00048	-0,00092	-0,00062
57.	6,22503	-4,47782	3,72776	0,00084	0,00002	-0,00043
58.	5,32695	-2,11885	3,20997	-0,00028	0,00000	0,00105

59.	5,32701	-2,11901	3,21000	0,00016	0,00000	0,00000
60.	6,22505	-4,47792	3,72778	-0,00002	0,00000	0,00003

Πίνακας 3.2.1 Οι τιμές των μεταβλητών και των D.C. συναρτήσεων

Εδώ το σύνολο  $X$  ορίστηκε το  $[-20,20] \times [-20,20] \times [-20,20]$ . Το αντίστοιχο σύνολο  $Y$  είναι το  $[76.1505, 1200] \times [0,400] \times [0,3200]$ . Από τον πίνακα φαίνεται πως υπάρχουν δύο σημεία που ικανοποιούν τις εξισώσεις στο σύνολο  $X$ , τα  $(5.32701, -2.11901, 3.21000)$  και  $(6.22505, -4.47792, 3.72778)$ .

### 3.3

Στα προηγούμενα δύο παραδείγματα οι εξισώσεις ήταν της μορφής:

$$\mathbf{x}^T D \mathbf{x} + \mathbf{b}^T \mathbf{x} + c = 0$$

όπου η μήτρα  $D$  ήταν διαγώνια. Στο παράδειγμα αυτής της ενότητας οι εξισώσεις είναι της μορφής:

$$\mathbf{x}^T A \mathbf{x} + \mathbf{b}^T \mathbf{x} + c = 0$$

όπου η μήτρα  $A$  κάθε εξίσωσης δεν είναι διαγώνια, αλλά της μορφής:

$$A = LDL^T$$

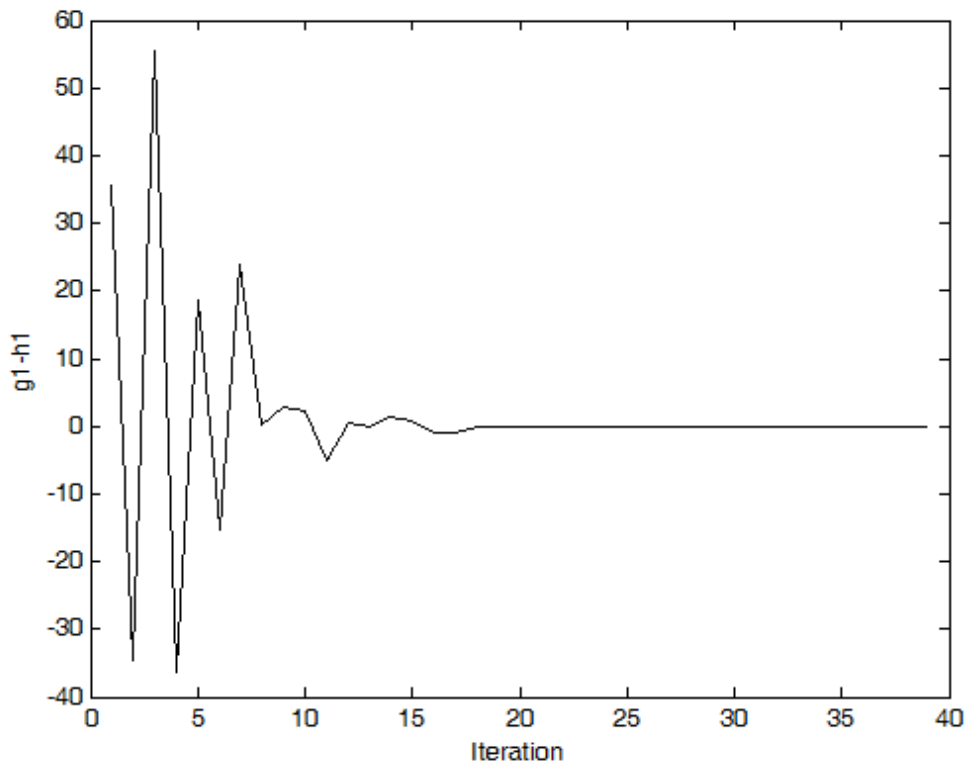
Η μήτρα  $L$  είναι κάτω τριγωνική με  $l_{ii} = 1, i = 1 \dots n$  και η μήτρα  $D$  είναι διαγώνια και κάποια στοιχεία της διαγωνίου της είναι θετικά, ενώ κάποια άλλα αρνητικά. Κάθε εξίσωση μπορεί να γραφεί με D.C. όρους ως εξής:

$$(\mathbf{x}^T LD_+ L^T \mathbf{x} + \mathbf{b}^T \mathbf{x} + c) - (\mathbf{x}^T LD_- L^T \mathbf{x}) = 0$$

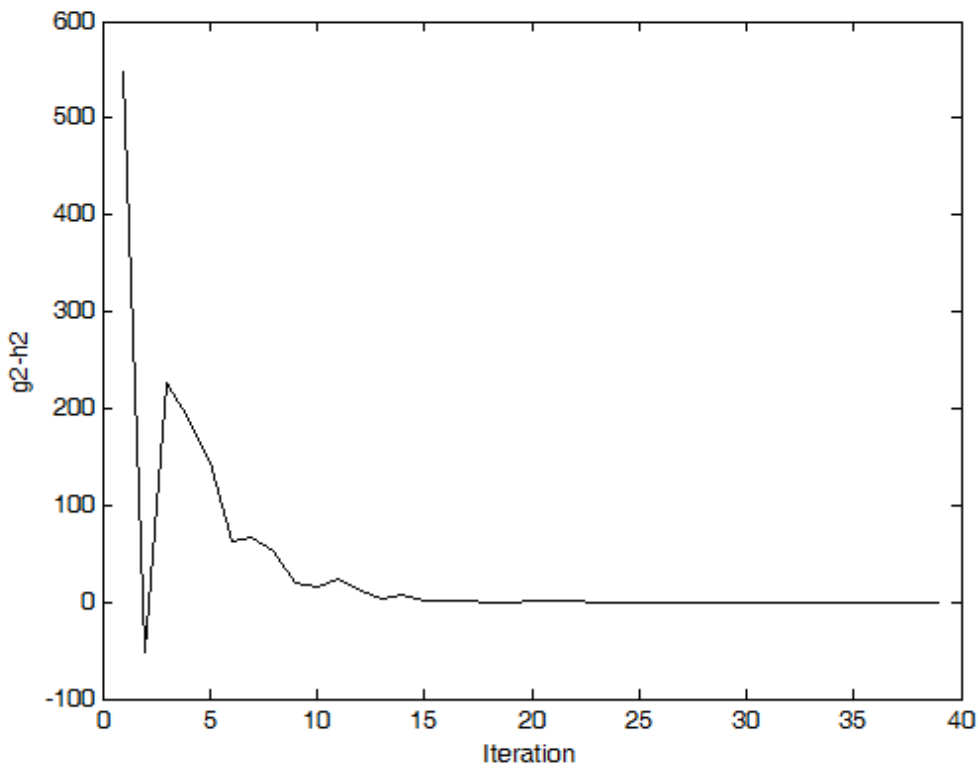
όπου

$$D = D_+ - D_-$$

και οι μήτρες  $D_+, D_-$  είναι διαγώνιες, τέτοιες ώστε η  $D_+$  έχει ως στοιχεία τα διαγώνια στοιχεία της  $D$  που είναι θετικά στις αντίστοιχες θέσεις και η μήτρα  $D_-$  έχει ως στοιχεία τις απόλυτες τιμές των αρνητικών στοιχείων της  $D$  στις αντίστοιχες θέσεις. Η μήτρα  $L_{4 \times 4}$  και το διάνυσμα  $\mathbf{b}_{4 \times 1}$  παίρνουν τυχαίες τιμές. Η μήτρα  $D_{4 \times 4}$  είναι συγκεκριμένη ώστε να είναι γνωστοί οι D.C. όροι, ενώ οι σταθερές  $c$  επιλέχθηκαν έτσι ώστε η λύση του συστήματος να είναι η  $(1,2,2,1)$ . Τα αποτελέσματα είναι τα παρακάτω:

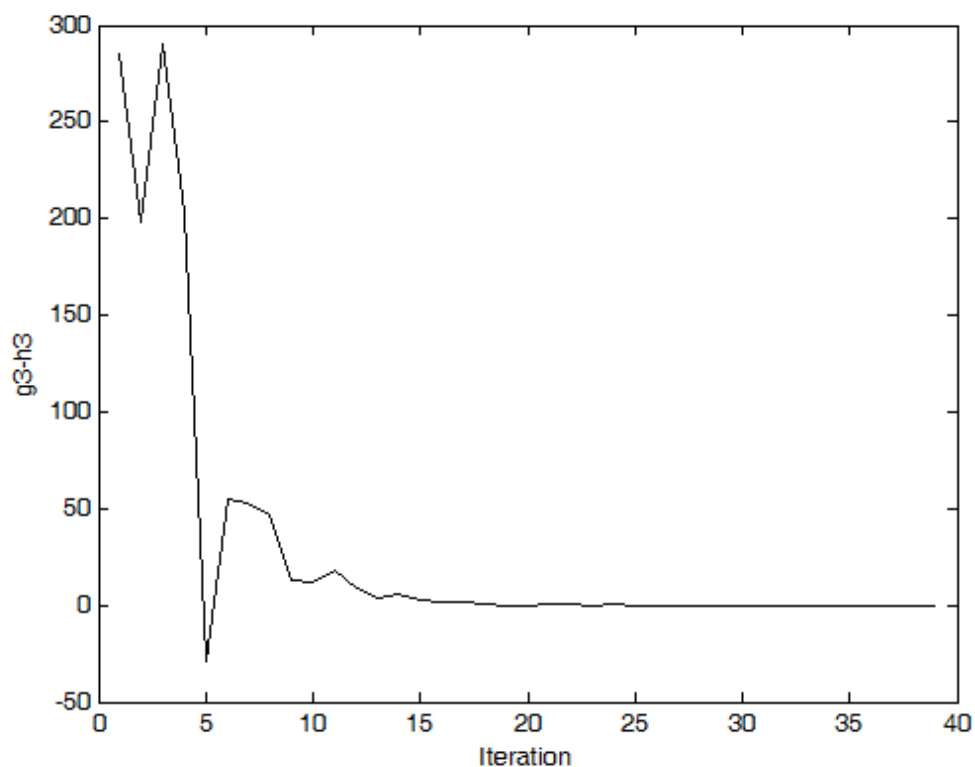


Σχήμα 3.3.1 Οι τιμές της πρώτης εξίσωσης

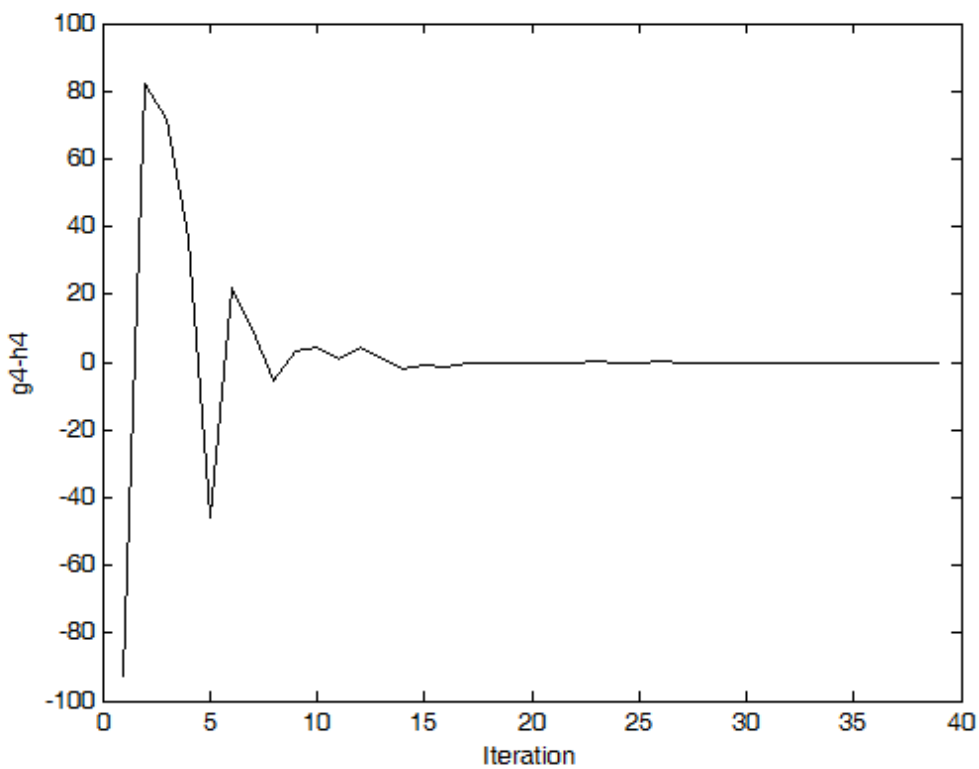


Σχήμα 3.3.2 Οι τιμές της δεύτερης εξίσωσης





Σχήμα 3.3.3 Οι τιμές της τρίτης εξίσωσης



Σχήμα 3.3.4 Οι τιμές της τέταρτης εξίσωσης

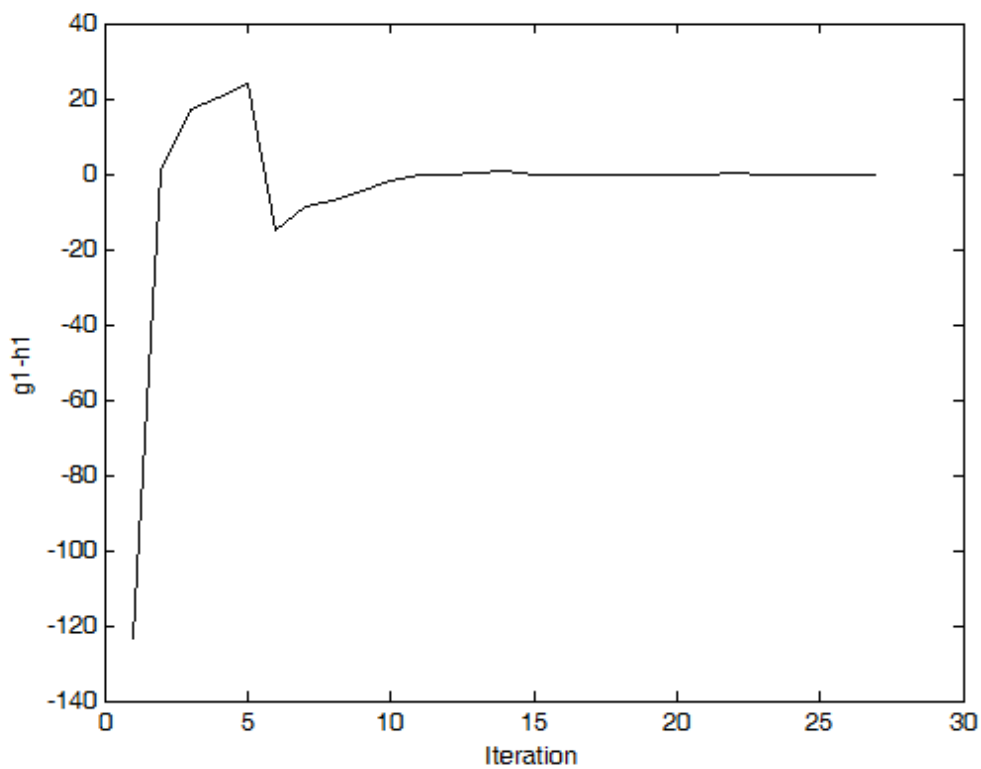
ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Επανάληψη	$x_1$	$x_2$	$x_3$	$x_4$	$g_1 - h_1$	$g_2 - h_2$	$g_3 - h_3$	$g_4 - h_4$
1.	0,0000	0,0000	0,0000	0,0000	35,46682	548,25699	284,71463	-92,67303
2.	1,7055	1,7064	1,6981	1,6981	-34,60919	-51,07238	197,74551	82,08026
3.	1,4818	1,4830	0,7160	1,4720	55,47797	227,21310	290,07794	71,91444
4.	0,6793	0,6801	1,8148	1,6585	-36,31562	188,88419	203,07816	36,67255
5.	0,6601	1,7293	1,9569	0,6554	18,76435	143,81262	-29,18927	-46,06440
6.	0,6230	1,1621	2,2302	1,4435	-15,17340	62,38759	54,67529	22,02415
7.	0,6259	2,2110	1,6494	0,8820	23,88474	65,80859	52,68595	10,39388
8.	1,6712	1,1665	2,2577	1,2108	0,30474	52,12434	46,44255	-5,67089
9.	0,6069	2,0885	1,8909	1,0093	2,94760	20,14343	12,70039	3,37760
10.	0,6066	2,0912	1,8980	1,0218	2,14029	15,63588	12,28470	4,57433
11.	0,6096	1,9504	1,9501	1,0823	-4,91790	23,09749	17,73039	0,76078
12.	0,6050	2,0790	1,9186	1,0349	0,53321	11,54777	9,20754	4,45409
13.	1,5677	1,2675	2,3852	1,2419	-0,06737	4,14620	3,46980	0,83997
14.	1,6734	1,2642	2,3778	1,2149	1,38076	6,64787	5,36489	-2,20692
15.	1,0852	1,9850	1,9985	0,9856	0,80101	2,33399	2,41568	-0,96073
16.	1,5908	1,3094	2,3694	1,2182	-0,85784	1,90385	1,78719	-1,46652
17.	1,5634	1,3003	2,3734	1,2287	-1,00200	2,25530	2,04697	-0,40926
18.	1,0180	1,9936	2,0015	1,0003	-0,08763	0,14371	0,73954	-0,13350
19.	1,0114	1,9959	2,0023	1,0005	-0,10219	-0,22863	0,14027	-0,09006
20.	0,9806	2,0007	1,9990	1,0019	-0,11359	0,60131	-0,14751	-0,04513
21.	0,9933	1,9981	1,9994	1,0018	-0,11861	0,49990	0,37029	-0,03360
22.	1,5918	1,2851	2,3875	1,2313	-0,01471	0,80110	0,52281	-0,07431
23.	0,9984	2,0001	1,9995	1,0001	0,00678	0,14306	0,10168	0,00499
24.	1,5949	1,2854	2,3891	1,2316	-0,00446	0,19125	0,18197	-0,00978
25.	1,0032	1,9988	2,0002	0,9997	0,00574	0,13139	0,09279	-0,06449
26.	1,5944	1,2857	2,3895	1,2317	-0,00080	0,03101	0,01243	0,01280
27.	1,5947	1,2859	2,3895	1,2316	-0,00029	0,01066	-0,00292	-0,00183
28.	0,9998	2,0000	1,9999	1,0000	0,00250	0,04084	0,02472	-0,00384
29.	1,5949	1,2859	2,3895	1,2316	-0,00006	0,00085	0,00137	-0,00044
30.	1,5950	1,2858	2,3895	1,2316	0,00047	0,00064	0,00121	-0,00041
31.	1,0000	2,0000	2,0000	1,0000	0,00138	0,00510	0,00384	-0,00060
32.	1,5949	1,2859	2,3895	1,2316	-0,00002	-0,00019	0,00058	-0,00029
33.	1,5949	1,2859	2,3895	1,2316	-0,00001	0,00047	0,00035	-0,00025
34.	1,0000	2,0000	2,0000	1,0000	-0,00006	0,00023	0,00017	-0,00003
35.	1,0000	2,0000	2,0000	1,0000	0,00000	0,00000	0,00000	0,00000
36.	1,0000	2,0000	2,0000	1,0000	0,00000	0,00000	0,00000	0,00000
37.	1,0000	2,0000	2,0000	1,0000	0,00000	0,00000	0,00000	0,00000
38.	1,0000	2,0000	2,0000	1,0000	0,00000	0,00000	0,00000	0,00000
39.	1,0000	2,0000	2,0000	1,0000	0,00000	0,00000	0,00000	0,00000

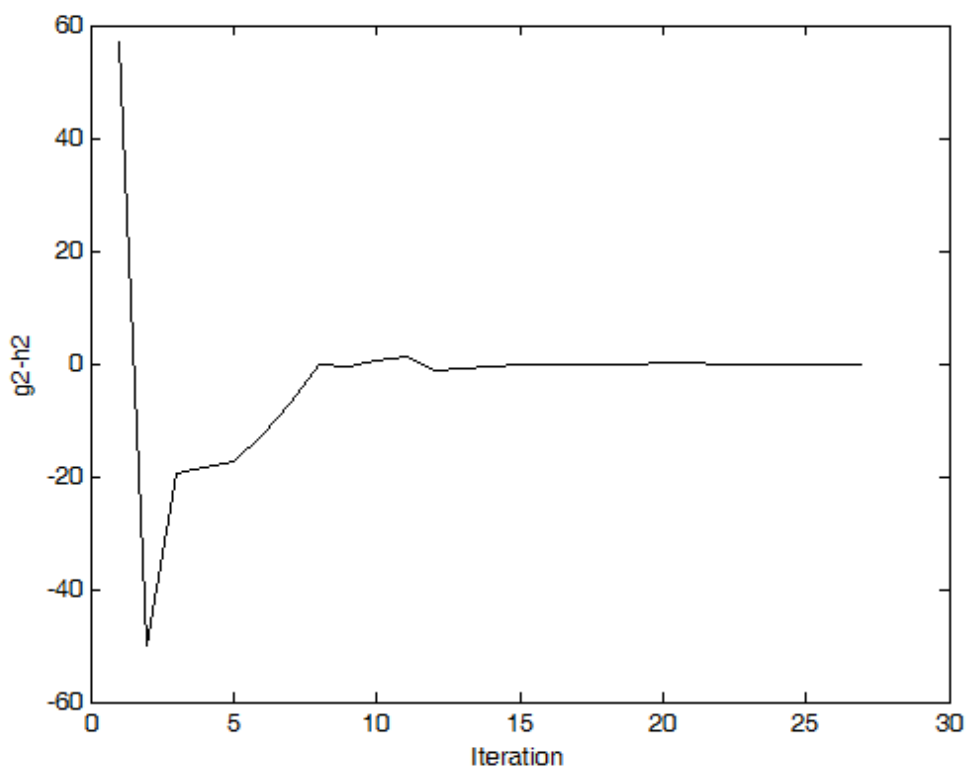
Πίνακας 3.3.1 Οι τιμές των μεταβλητών και των D.C. συναρτήσεων

### 3.4

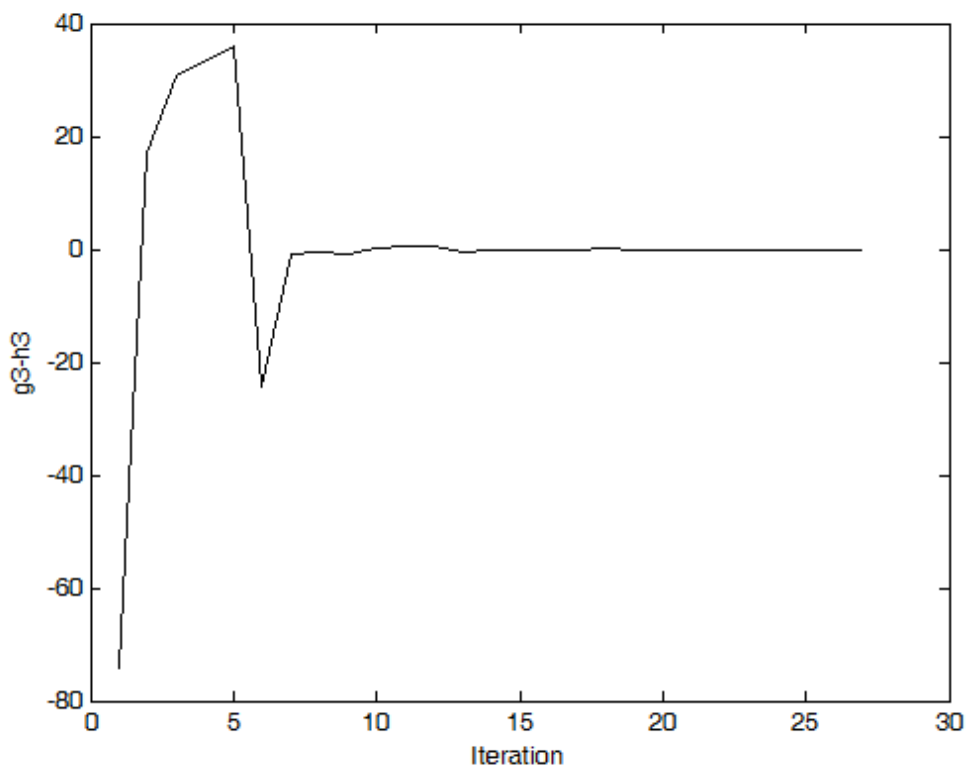
Το παράδειγμα αυτής της παραγράφου είναι παρόμοιο με το προηγούμενο, αλλά με 5 εξισώσεις και 5 αγνώστους. Το σύστημα είναι κατασκευασμένο ώστε η λύση να είναι η (1.2 , 1.4 , 1.6 , 1.4 , 1.2)



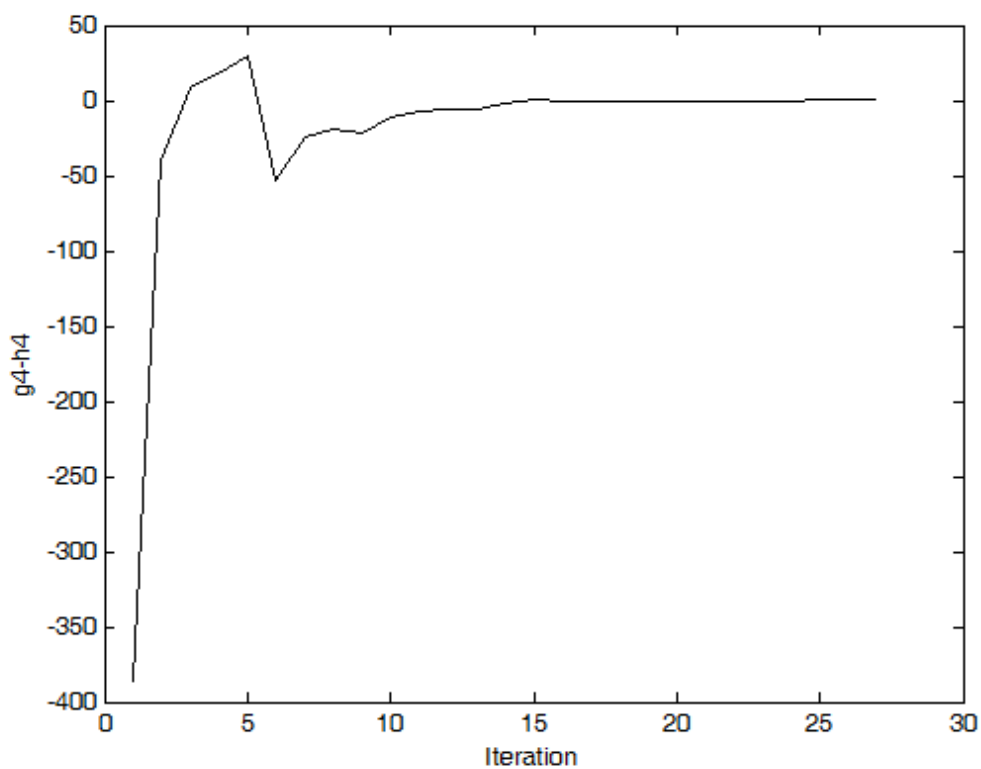
Σχήμα 3.4.1 Οι τιμές της πρώτης εξίσωσης



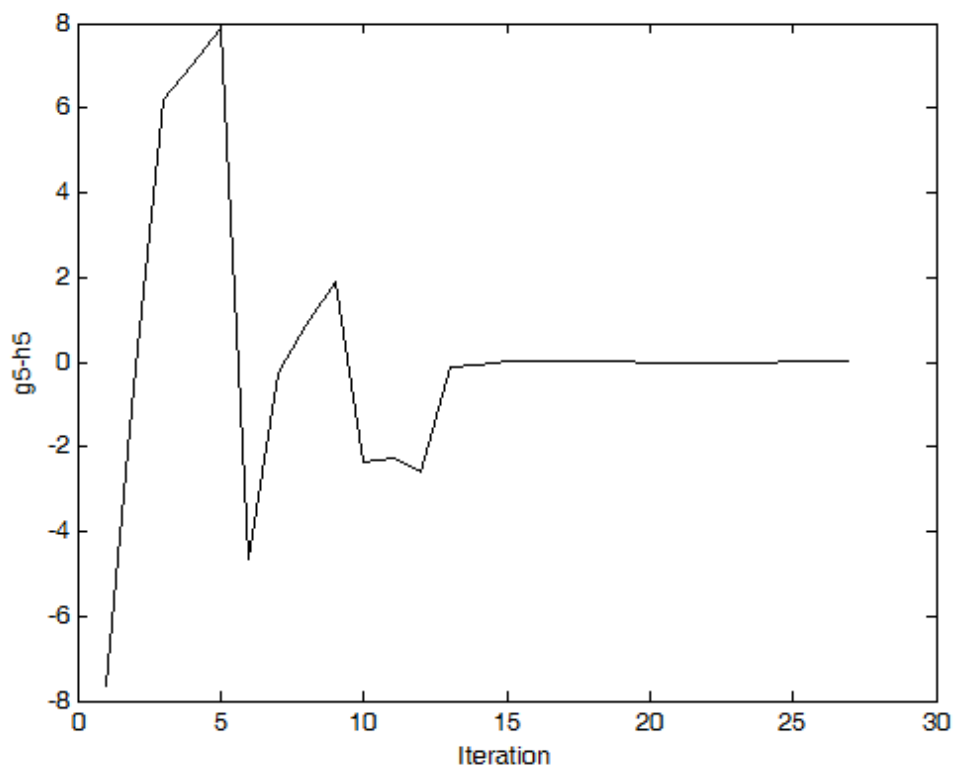
Σχήμα 3.4.2 Οι τιμές της δεύτερης εξίσωσης



Σχήμα 3.4.3 Οι τιμές της τρίτης εξίσωσης



Σχήμα 3.4.4 Οι τιμές της τέταρτης εξίσωσης



Σχήμα 3.4.5 Οι τιμές της πέμπτης εξίσωσης

ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

Επανάληψη	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
1.	0,0000	0,0000	0,0000	0,0000	0,0000
2.	1,2626	1,7328	1,2626	1,2626	1,2626
3.	1,3497	1,3882	1,3497	1,3497	1,3497
4.	1,3659	1,3738	1,3659	1,3659	1,3659
5.	1,3838	1,3578	1,3838	1,3838	1,3838
6.	0,9287	1,7039	1,4758	1,4148	1,0967
7.	0,9128	1,5231	1,5893	1,3581	1,1939
8.	0,9095	1,4463	1,6083	1,3731	1,2016
9.	0,9109	1,4384	1,5179	1,4002	1,2189
10.	1,7538	1,2386	1,5014	1,2935	1,1382
11.	1,7717	1,2276	1,5132	1,3004	1,1415
12.	1,7748	1,2606	1,5167	1,3008	1,1418
13.	1,2587	1,3900	1,5484	1,3919	1,1955
14.	1,2448	1,3936	1,5675	1,3994	1,1989
15.	1,1781	1,4096	1,6075	1,4016	1,2012
16.	1,1902	1,4046	1,5985	1,3997	1,2004
17.	1,1944	1,4013	1,5990	1,3998	1,2003
18.	1,1998	1,3989	1,5988	1,3981	1,2002
19.	1,1937	1,4015	1,5996	1,4005	1,2005
20.	1,2032	1,3991	1,5995	1,3994	1,1996
21.	1,2007	1,3996	1,5997	1,3998	1,1999
22.	1,2002	1,4000	1,5999	1,4000	1,2000
23.	1,2000	1,4000	1,6000	1,4000	1,2000
24.	1,2000	1,4000	1,6000	1,4000	1,2000
25.	1,2000	1,4000	1,6000	1,4000	1,2000
26.	1,2000	1,4000	1,6000	1,4000	1,2000
27.	1,2000	1,4000	1,6000	1,4000	1,2000

Πίνακας 3.4.1 Οι τιμές των μεταβλητών

Επανάληψη	$g_1 - h_1$	$g_2 - h_2$	$g_3 - h_3$	$g_4 - h_4$	$g_5 - h_5$
1.	-123,66143	57,16710	-74,37399	-386,07151	-7,66717
2.	0,59160	-50,07141	17,01590	-40,15324	-0,59138
3.	17,11784	-19,34153	30,57568	8,86641	6,21474
4.	20,43301	-18,42439	33,13585	18,35478	6,99373
5.	24,16340	-17,40515	36,01687	29,03162	7,88395
6.	-15,24616	-12,81242	-24,21581	-52,76027	-4,65378
7.	-8,61155	-6,89549	-1,04632	-24,64954	-0,24550
8.	-7,25778	-0,21925	-0,42768	-18,67688	0,86061
9.	-4,32417	-0,54925	-0,90881	-21,44185	1,91593
10.	-1,63795	0,39057	0,05195	-11,06416	-2,34681
11.	-0,49694	1,23402	0,49452	-6,81410	-2,25210
12.	-0,25398	-1,30689	0,56729	-6,01600	-2,58814
13.	0,06794	-1,02216	-0,47879	-5,32893	-0,12965
14.	0,64478	-0,64252	-0,25375	-1,85459	-0,05714
15.	-0,18242	-0,26811	-0,00817	0,22637	0,00766
16.	-0,17263	-0,26175	-0,02620	-0,68173	0,01340
17.	-0,09813	-0,05194	-0,00722	-0,39302	0,02458
18.	-0,09554	-0,05026	0,16180	-0,38175	0,02533
19.	-0,05248	-0,01912	-0,02041	-0,21696	0,03130
20.	-0,01394	0,00798	-0,00646	-0,06769	-0,01694
21.	-0,00945	0,01135	-0,00479	-0,05049	-0,00192
22.	0,00162	-0,00178	-0,00078	-0,00789	-0,00029
23.	-0,00065	-0,00051	-0,00013	-0,00104	-0,00003
24.	-0,00065	-0,00024	-0,00013	-0,00104	-0,00003
25.	-0,00006	-0,00004	-0,00003	0,00006	0,00001
26.	-0,00006	-0,00004	-0,00003	0,00006	0,00001
27.	-0,00006	-0,00004	-0,00003	0,00006	0,00001

Πίνακας 3.4.2 Οι τιμές των D.C. συναρτήσεων

### 3.5

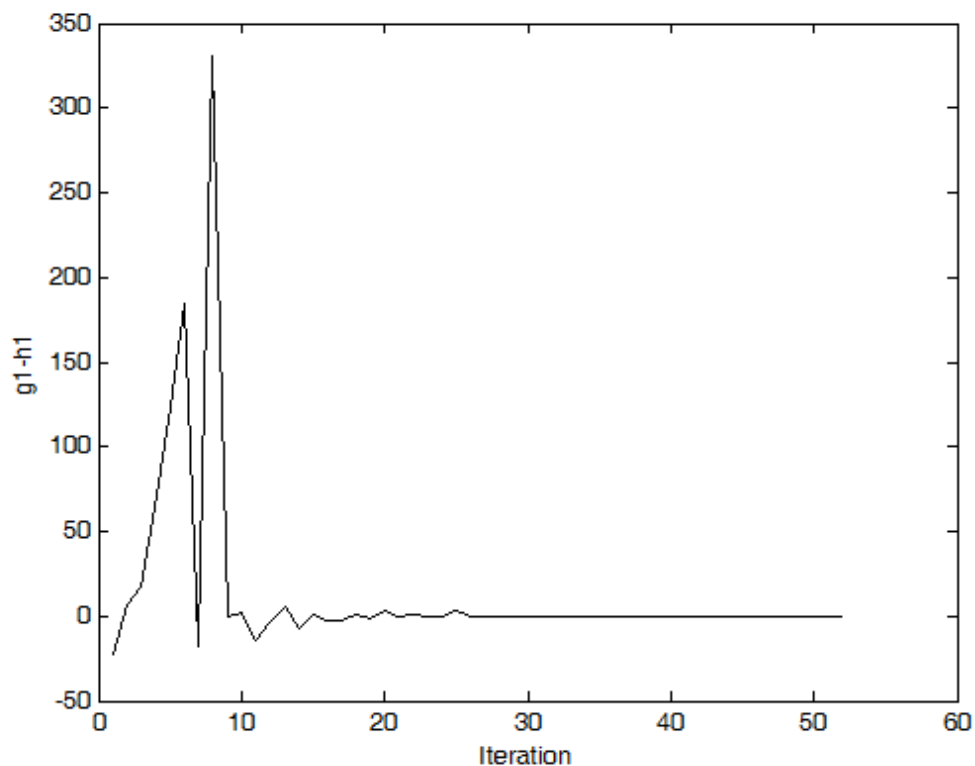
Το παράδειγμα αυτής της παραγράφου είναι το παρακάτω σύστημα που περιλαμβάνει και εκθετικές συναρτήσεις:

$$\begin{cases} \exp(x_1 + 1.2x_2) + x_1^2 - 2x_2^2 - 23.7492 = 0 \\ \exp(1.4x_1 + 1.6x_2) + 4x_1^2 - x_2^2 - 98.1035 = 0 \end{cases}$$

Το σύστημα μπορεί να γραφεί με D.C. όρους ως εξής:

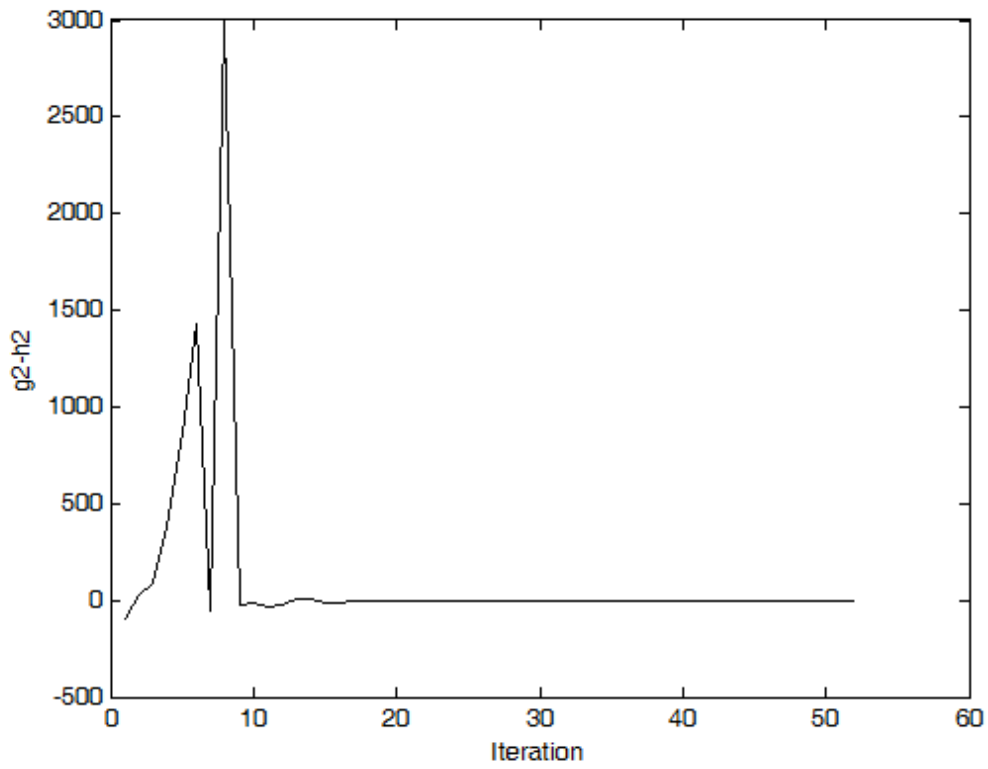
$$\begin{cases} (\exp(x_1 + 1.2x_2) + 2x_1^2 + x_2^2 - 23.7492) - (x_1^2 + 3x_2^2) = 0 \\ (\exp(1.4x_1 + 1.6x_2) + 5x_1^2 + x_2^2 - 98.1035) - (x_1^2 + 2x_2^2) = 0 \end{cases}$$

Οι εκθετικές συναρτήσεις της παραπάνω μορφής είναι πράγματι κυρτές. Αυτό αποδεικνύεται ως εξής: προφανώς η εκθετική συνάρτηση  $e^x$ ,  $x \in \mathbb{R}$  είναι κυρτή. Συνεπώς η σύνθεση αυτής με τη συνάρτηση  $Ax$ ,  $A \in \mathbb{R}^{1 \times n}$ ,  $x \in \mathbb{R}^n$  είναι επίσης κυρτή, με βάση το αποτέλεσμα 2 της παραγράφου 1.6. Το παράδειγμα είναι κατασκευασμένο ώστε το ζεύγος  $(1.2, 1.8)$  να αποτελεί λύση.

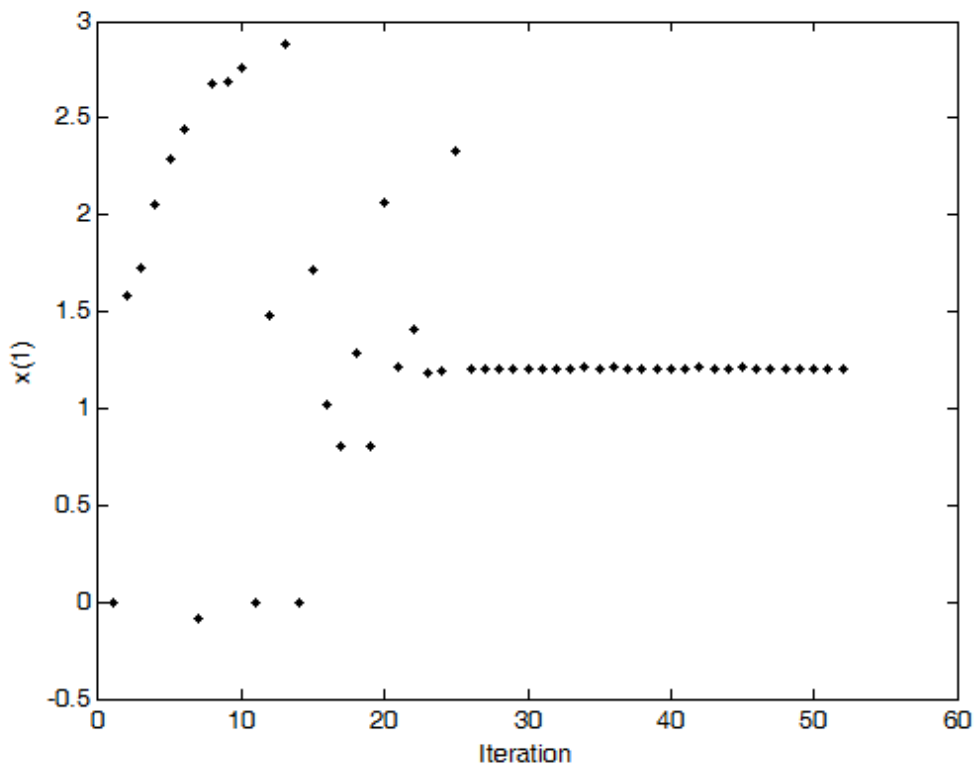


Σχήμα 3.5.1 Οι τιμές της πρώτης εξίσωσης

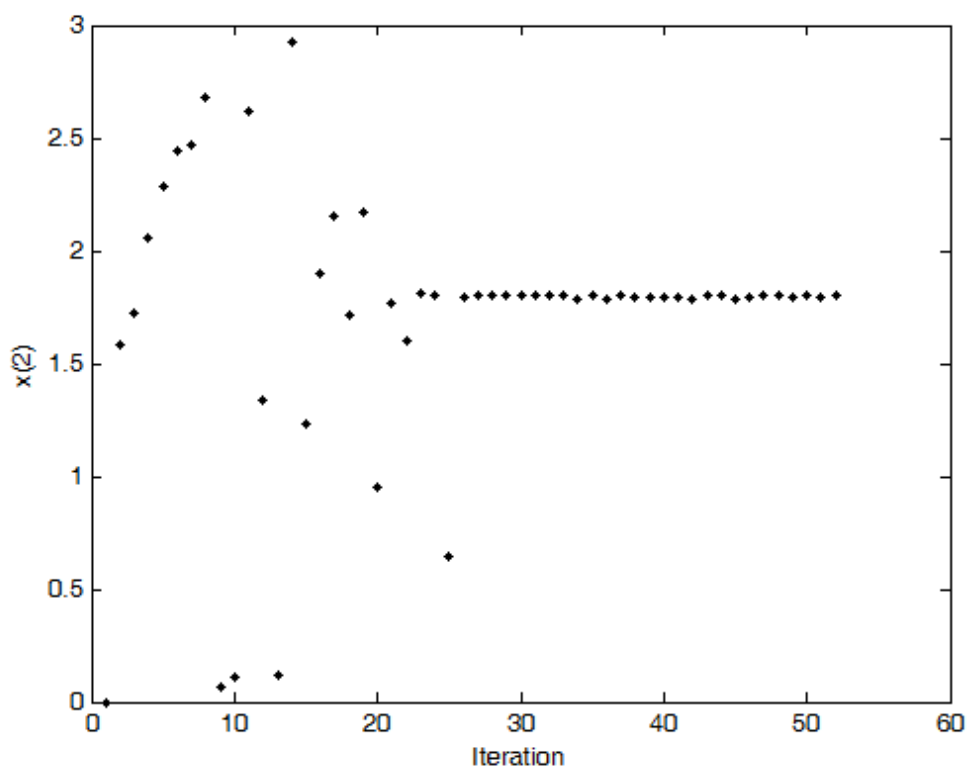




Σχήμα 3.5.2 Οι τιμές της δεύτερης εξίσωσης



Σχήμα 3.5.3 Οι τιμές της πρώτης μεταβλητής



Σχήμα 3.5.4 Οι τιμές της δεύτερης μεταβλητής

Επανάληψη	$x_1$	$x_2$	$g_1 - h_1$	$g_2 - h_2$
1.	0,00000	0,00000	-22,74920	-97,10350
2.	1,58434	1,58434	6,38100	25,36088
3.	1,72475	1,72475	17,72993	87,48547
4.	2,05279	2,05279	63,51898	387,19910
5.	2,28448	2,28448	123,33209	864,68281
6.	2,44044	2,44044	184,93845	1431,98349
7.	-0,08201	2,47318	-18,05766	-57,56155
8.	2,67690	2,67690	330,19375	2997,28289
9.	2,68852	0,06562	-0,61456	-21,30429
10.	2,75450	0,11067	1,75845	-11,31567
11.	-0,00139	2,61829	-14,34338	-39,11334
12.	1,47968	1,33914	-3,24285	-23,50077
13.	2,88092	0,11575	5,01117	3,01199
14.	-0,00155	2,92887	-7,35412	1,52119
15.	1,71242	1,23250	0,46802	-8,89770
16.	1,01709	1,90309	-2,82388	-10,32895
17.	0,80868	2,15537	-2,56814	-2,54622

ΚΕΦΑΛΑΙΟ 3 – ΠΑΡΑΔΕΙΓΜΑΤΑ - ΣΥΜΠΕΡΑΣΜΑΤΑ

18.	1,28363	1,71554	0,29580	-0,58268
19.	0,80970	2,17460	-2,00562	0,57084
20.	2,06640	0,94771	3,34661	0,28912
21.	1,21791	1,77012	-0,25554	-1,86932
22.	1,40664	1,60215	1,01202	0,25583
23.	1,18404	1,81563	-0,07056	0,04576
24.	1,19304	1,80450	-0,09421	-0,32687
25.	2,32930	0,64299	3,06728	-3,86201
26.	1,20439	1,79539	0,01076	-0,05912
27.	1,19997	1,79995	-0,00238	-0,01221
28.	1,19992	1,80008	-0,00033	0,00027
29.	1,19999	1,80000	-0,00015	-0,00059
30.	1,20002	1,79998	0,00004	-0,00021
31.	1,20011	1,79989	0,00020	-0,00171
32.	1,20011	1,79989	0,00026	-0,00141
33.	1,20011	1,79989	0,00020	-0,00171
34.	1,21706	1,78464	0,11192	0,15405
35.	1,20005	1,79986	-0,00228	-0,01366
36.	1,21706	1,78464	0,11192	0,15405
37.	1,20007	1,79993	0,00017	-0,00094
38.	1,20295	1,79722	0,01580	0,00712
39.	1,20295	1,79722	0,01580	0,00712
40.	1,20295	1,79722	0,01580	0,00712
41.	1,20295	1,79722	0,01580	0,00712
42.	1,21706	1,78464	0,11192	0,15405
43.	1,20005	1,79986	-0,00220	-0,01330
44.	1,20011	1,79989	0,00020	-0,00171
45.	1,21706	1,78464	0,11192	0,15405
46.	1,20295	1,79721	0,01583	0,00718
47.	1,20005	1,79986	-0,00235	-0,01404
48.	1,20005	1,79986	-0,00235	-0,01404
49.	1,20295	1,79721	0,01583	0,00718
50.	1,20005	1,79986	-0,00234	-0,01401
51.	1,20295	1,79721	0,01583	0,00718
52.	1,20005	1,79986	-0,00235	-0,01404

Πίνακας 3.5.1 Οι τιμές των μεταβλητών

### 3.4 Συμπεράσματα

Ανακεφαλαιώνοντας, ασχοληθήκαμε με το πρόβλημα επίλυσης συστημάτων μη γραμμικών εξισώσεων κυρτών διαφορών. Δείχθηκε πως, αντί αυτού, μπορεί να επιλυθεί ένα ισοδύναμο πρόβλημα ελαχιστοποίησης κοίλης συνάρτησης με κυρτούς περιορισμούς, εφόσον οι εξισώσεις γραφούν με D.C. όρους, και εφόσον ορίσουμε εμείς ένα σύνολο στο οποίο αναζητείται η λύση του συστήματος. Το νέο πρόβλημα αντιμετωπίστηκε με τον αλγόριθμο εξωτερικής προσέγγισης της K.L.Hoffman. Ο αλγόριθμος αυτός υλοποιήθηκε στο περιβάλλον MATLAB και εφαρμόστηκε σε πέντε παραδείγματα συστημάτων κυρτών διαφορών, αρχικά απλά και στη συνέχεια πιο σύνθετα.

Συμπερασματικά, κατά την εφαρμογή του αλγορίθμου στα παραπάνω παραδείγματα φαίνεται πως η γενίκευση του σε προβλήματα συστημάτων εξισώσεων κυρτών διαφορών με τον τρόπο που παρουσιάστηκε στο Κεφάλαιο 2, είναι επιτυχής. Επιτυγχάνεται δηλαδή η εύρεση λύσης στο κάθε σύστημα, αλλά και στις περιπτώσεις όπου υπάρχουν περισσότερες λύσεις (παράδειγμα 3.1 και παράδειγμα 3.2), ο αλγόριθμος τις εντοπίζει. Επίσης, όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, το εύρος των προβλημάτων στα οποία μπορεί να εφαρμοστεί είναι μεγάλο, αφού κάθε συνάρτηση στη  $C^2$  κλάση μπορεί να γραφεί ως συνάρτηση κυρτών διαφορών.

Ωστόσο, διαπιστώθηκε πως σε προβλήματα περισσότερων διαστάσεων (παραδείγματα 3.3 και 3.4, τεσσάρων και πέντε διαστάσεων αντίστοιχα), το σύνολο στο οποίο ορίζουμε πως θα αναζητηθεί η λύση μπορεί να καταστήσει το χρόνο εκτέλεσης του αλγορίθμου απαγορευτικό, έτσι το σύνολο αυτό περιορίστηκε σημαντικά. Το γεγονός αυτό οφείλεται κυρίως στο ότι καθώς ο αριθμός επαναλήψεων αυξάνεται, τα γραμμικά υποπροβλήματα μεγαλώνουν, αφού σε κάθε επανάληψη προστίθενται νέοι γραμμικοί περιορισμοί.

Εν γένει, από άποψη υπολογιστικού κόστους, τα προβλήματα κοίλης ελαχιστοποίησης ανήκουν στην κλάση NP-hard, ακόμα και όταν πρόκειται για απλές περιπτώσεις όπως η ελαχιστοποίηση τετραγωνικής συνάρτησης σε υπερκύβο (με ορισμένες εξαιρέσεις, όπως για παράδειγμα η ελαχιστοποίηση ευκλείδειας νόρμας σε υπερρθογώνιο). Περισσότερα στοιχεία για την πολυπλοκότητα των αντίστοιχων αλγορίθμων μπορούν να βρεθούν στο [10].

Αναφέρεται, τέλος, πως έχουν προταθεί νεότεροι τρόποι επίλυσης προβλημάτων κοίλης ελαχιστοποίησης, οι οποίοι δεν ανήκουν αμιγώς στην κατηγορία μεθόδων εξωτερικής προσέγγισης, αλλά συνδυάζουν και ιδέες των μεθόδων διαχωρισμού (branch and bound). Ο αλγόριθμος των H.P.Benson και R.Horst [12] είναι ένας από αυτούς, και έχει την ιδιότητα της εγγυημένης σύγκλισης σε ένα σημείο ολικού ελαχίστου. Το πλεονέκτημα είναι πως πλέον δεν είναι απαραίτητος ο υπολογισμός των νέων ακραίων σημείων των διαδοχικών συνόλων εξωτερικής προσέγγισης. Αντί αυτού, ο μοναδικός ουσιώδης μη γραμμικός

υπολογισμός που απαιτείται σε κάθε επανάληψη μπορεί να επιτευχθεί με απλές μεθόδους έρευνας γραμμής. Επίσης, άλλοι βασικοί υπολογισμοί που απαιτούνται σε κάθε επανάληψη περιλαμβάνουν είτε την επίλυση υπο-προβλημάτων γραμμικού προγραμματισμού είτε την επίλυση γραμμικών συστημάτων  $n + 1$  εξισώσεων, τα οποία έχουν μοναδική λύση. Πιθανό μειονέκτημα του αλγορίθμου είναι πως καθώς οι επαναλήψεις αυξάνονται, οι απαιτήσεις σε μνήμη και ο αριθμός των περιορισμών στα υπο-προβλήματα γραμμικού προγραμματισμού επίσης αυξάνονται. Η απόδοση αυτών των νεότερων μεθόδων σε ποικίλα προβλήματα μένει να εξεταστεί.

# Παράρτημα

## Κώδικας MATLAB

```
function f=diplwmatikh1
clear;

%Ορισμος συναρτησεων και κλισεων με βαση τα δεδομενα που εισαγει ο
χρηστης

[f_str n]=CreateFuncStructure;
g_str=CreateGradStructure;

%Κουτί για x
for i=1:n
    xmin(i)=-10;
    xmax(i)=10;
end
xmin=xmin';
xmax=xmax';

%Κάτω φράγματα του y
for i=1:n
    infin(i)=Inf;
end
a=zeros(2*n,n);

options=optimset('Algorithm','active-set');

a(:,1) = fmincon(@(z) fun1(z,n),zeros(2*n,1),[],[],[],[],[xmin' -
infin],[xmax' infin],@(z) nonlcon1(z,f_str,n),options);

a(:,2) = fmincon(@(z) fun2(z,n),zeros(2*n,1),[],[],[],[],[xmin' -
infin],[xmax' infin],@(z) nonlcon2(z,f_str,n),options);

for i=1:n
    ymin(i)=a(n+i,i);
end
ymin=ymin';

%Ανω φράγματα του y
ymax=[];

res1=[];
res2=[];

for i=1:n
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
res=zeros(2,n);
res(:,i)=[-1 1]';
res1=[res1;res];
end

for i=1:n
    res2(2*i-1)=-xmin(i);res2(2*i)=xmax(i);
end
res2=res2';
W=[con2vert(res1,res2) zeros(2^n,n)]';

values=[];

for i=1:(2*n)
    for j=1:(2^n)
        values(i,j)=f_str{i}(W(:,j));
    end
end

maxima=max(values');

for i=1:n
    ymax(i)=max([maxima(i) maxima(n+i)]);
end

ymax=ymax';

%Τελικό κουτί για (z)
zmin=[xmin' ymin']';
zmax=[xmax' ymax']';

%Εσωτερικό σημείο
p=fmincon(@ (ze) fun(ze,n),zeros(2*n+1,1),[],[],[],[],[zmin' -Inf],[
zmax' Inf],@(ze)nonlcon3(ze(1:4),f_str,n,ze(5)),options);

%Νέο κουτί για x

xmin=[];
xmax=[];
ymin=[];
for i=1:n
    xmin(i)=-50;
    xmax(i)=50;
end
xmin=xmin';
xmax=xmax';

%Νέα κάτω φράγματα του y

for i=1:n
    infin(i)=Inf;
end
a=zeros(2*n,n);
a(:,1) = fmincon(@(z) fun1(z,n),zeros(2*n,1),[],[],[],[],[xmin' -
infin],[xmax' infin],@(z) nonlcon1(z,f_str,n),options);
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
a(:,2) = fmincon(@(z) fun2(z,n),zeros(2*n,1),[],[],[],[],[xmin' -
infin],[xmax' infin],@(z) nonlcon2(z,f_str,n),options);

for i=1:n
    ymin(i)=a(n+i,i);
end
ymin=ymin';

%Νέα άνω φράγματα του y
ymax=[];

res1=[];
res2=[];

for i=1:n
    res=zeros(2,n);
    res(:,i)=[-1 1]';
    res1=[res1;res];
end

for i=1:n
    res2(2*i-1)=-xmin(i);res2(2*i)=xmax(i);
end
res2=res2';
W=[con2vert(res1,res2) zeros(2^n,n)]';

values=[];

for i=1:(2*n)
    for j=1:(2^n)
        values(i,j)=f_str{i}(W(:,j));
    end
end

maxima=max(values');

for i=1:n
    ymax(i)=max([maxima(i) maxima(n+i)]);
end

ymax=ymax';

%Νέο τελικό κουτί για (z)
zmin=[xmin' ymin]';
zmax=[xmax' ymax]';

p=p(1:2*n);

[LQ,Lb]=enclosing_linear_polyhedron(zmax,zmin,2*n); %Αρχικό πολύεδρο

V{1} = (con2vert(LQ,Lb))'; %Αντίστοιχες κορυφές
u(1) = minfunc(p,f_str,n);
x(:,1)=zeros(2*n,1);
z(:,1)=zeros(2*n,1);
num=length(Lb);
```



## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```

k=2;
stop=0;
y=[];

while stop==0
    y=[];
    Vtemp=V{k-1};
    for i=1:size(Vtemp,2)
        y(i)=minfunc(Vtemp(:,i),f_str,n);
    end

    [II,I]=min(y);
    x(:,k)=Vtemp(:,I);

    options =
    optimset('GradObj','on','GradConstr','off','Algorithm','active-set');
    lamvda = fmincon(@(l) id_grad(l),0,[],[],[],[],0,1,@(l)
    (nonlcon4(x(:,k)+l*(p-x(:,k)),f_str,n)),options);
    %Προβλημα ελαχιστοποίησης γραμμικής συναρτησης
    if lamvda > 0
        z(:,k) = x(:,k)+lamvda*(p-x(:,k)); %Εύρεση σημείου στο συνορο
        του επιτρεπτού συνόλου
    if minfunc(z(:,k),f_str,n)<u(k-1)
        u(k)=minfunc(z(:,k),f_str,n);
    else
        u(k)=u(k-1);
    end
    if abs(u(k)-minfunc(x(:,k),f_str,n))<1e-5
        stop=1;
    else
        [g,gg]=nonlcon4(z(:,k),f_str,n);
        J{k}=[];
        for i=1:4
            if abs(g(i))<1e-5
                J{k}=[J{k};i];
            end
        end
        Jtemp=max(J{k});
        num=num+1;
        gradients=[];
        funs=[];
        for i=1:(2*n)
            gradients(:,i)=g_str{i}(z(:,k));
            funs(i)=f_str{i}(z(:,k));
        end
        for i=1:n
            if Jtemp==i
                LQ(num,:)=gradients(:,i)';
                Lb(num,:)=z(n+i,k)+gradients(:,i)'*z(:,k)-
funs(i);
            end
        end
        for i=(n+1):(2*n)
            if Jtemp==i

```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
LQ(num,:) = gradients(:, i)';
Lb(num,:) = z(i, k) + gradients(:, i)' * z(:, k) - funs(i);
end

end %Εύρεση ενεργού περιορισμού και προσθήκη
νέου περιορισμού για το πολύεδρο
[VV, nr, nre] = lcon2vert(LQ, Lb);
V{k} = VV'; %Εύρεση κορυφών πολύεδρου

end

else
    stop=1;
end

k=k+1;
if k==200
    stop=1;
end
end
f=z;
end

function [f n]=CreateFuncStructure
f{1}=@(z) z'*[2 0 0 0; 0 1 0 0; 0 0 0 0; 0 0 0 0]*z;
f{2}=@(z) z'*[5 0 0 0; 0 1 0 0; 0 0 0 0; 0 0 0 0]*z;
f{3}=@(z) z'*[1 0 0 0; 0 3 0 0; 0 0 0 0; 0 0 0 0]*z;
f{4}=@(z) z'*[1 0 0 0; 0 2 0 0; 0 0 0 0; 0 0 0 0]*z+5;
n=2;

function f =CreateGradStructure
f{1}=@(z) [4 0 0 0; 0 2 0 0; 0 0 0 0; 0 0 0 0]*z+[0 0 -1 0]';
f{2}=@(z) [10 0 0 0; 0 2 0 0; 0 0 0 0; 0 0 0 0]*z+[0 0 0 -1]';
f{3}=@(z) [2 0 0 0; 0 6 0 0; 0 0 0 0; 0 0 0 0]*z+[0 0 -1 0]';
f{4}=@(z) [2 0 0 0; 0 4 0 0; 0 0 0 0; 0 0 0 0]*z+[0 0 0 -1]';

function [A,b] = enclosing_linear_polyhedron (x_max,x_min,n)

A=[eye(n); -eye(n)];
b=[x_max; -x_min];

end
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
function [f,fgrad] = fun(x,n)
    f = x(2*n+1);
    fgrad = [zeros(2*n,1);1];
end
```

```
function [f,fgrad] = fun1(x,n)
    f = x(n+1);
    fgrad=zeros(n,1);
    fgrad(n+1) = 1;
end
```

```
function [f,fgrad] = fun2(x,n)
    f = x(n+2);
    fgrad=zeros(n,1);
    fgrad(n+2) = 1;
end
```

```
function [f,fgrad] = id_grad(x)
    f = x;
    fgrad = 1;
end
```

```
function f = minfunc(x,s,n)
f=0;
for i=1:n
    f=f-s{i}(x)-s{n+i}(x)+2*x(n+i);
end
```

```
function [c ceq] = nonlcon1(z,s,n)
c= [s{1}(z)-z(n+1) s{n+1}(z)-z(n+1)]';
ceq=[];
```

```
function [c, ceq]= nonlcon2(z,s,n)
c= [s{2}(z)-z(n+2) s{n+2}(z)-z(n+2)]';
ceq= [];
```

```
function [c, ceq]= nonlcon3(z,s,n,e)
c= [s{1}(z)-z(n+1)-e s{2}(z)-z(n+2)-e s{3}(z)-z(n+1)-e
s{4}(z)-z(n+2)-e]';
ceq= [];
```

```
function [c, ceq]= nonlcon4(z,s,n)
c= [s{1}(z)-z(n+1) s{2}(z)-z(n+2) s{3}(z)-z(n+1) s{4}(z)-
z(n+2)]';
ceq= [];
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
function [V,nr] = con2vert(A,b)
% CON2VERT - convert a convex set of constraint inequalities into the
set
%           of vertices at the intersections of those
inequalities;i.e.,
%           solve the "vertex enumeration" problem. Additionally,
%           identify redundant entries in the list of inequalities.
%
% V = con2vert(A,b)
% [V,nr] = con2vert(A,b)
%
% Converts the polytope (convex polygon, polyhedron, etc.) defined by
the
% system of inequalities A*x <= b into a list of vertices V. Each ROW
% of V is a vertex. For n variables:
% A = m x n matrix, where m >= n (m constraints, n variables)
% b = m x 1 vector (m constraints)
% V = p x n matrix (p vertices, n variables)
% nr = list of the rows in A which are NOT redundant constraints
c = A\b;
if ~all(A*c < b);
    [c,f,ef] = fminsearch(@obj,c,'params',{A,b});
    if ef ~= 1
        error('Unable to locate a point within the interior of a
feasible region.')
    end
end
b = b - A*c;
D = A ./ repmat(b,[1 size(A,2)]);
[k,v2] = convhulln([D;zeros(1,size(D,2))]);
[k,v1] = convhulln(D);
if v2 > v1
    error('Non-bounding constraints detected. (Consider box
constraints on variables.)')
end
nr = unique(k(:));
G = zeros(size(k,1),size(D,2));
for ix = 1:size(k,1)
    F = D(k(ix,:),:);
    G(ix,:)=F\ones(size(F,1),1);
end
V = G + repmat(c',[size(G,1),1]);
[null,I]=unique(num2str(V,6),'rows');
V=V(I,:);
return
function d = obj(c,params)
A=params{1};
b=params{2};
d = A*c-b;
k=(d>=-1e-15);
d(k)=d(k)+1;
d = max([0;d]);
return

% Επίσης χρησιμοποιήθηκαν οι συναρτήσεις lcon2vert, vert2lcon,
%qlcon2vert που αποτελούν βελτίωση της con2vert και παρατίθενται
%παρακάτω
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
function [V,nr,nre]=lcon2vert(A,b,Aeq,beq,TOL,checkbounds)
%An extension of Michael Kleder's con2vert function, used for finding
the
%vertices of a bounded polyhedron in R^n, given its representation as
a set
%of linear constraints. This wrapper extends the capabilities of
con2vert to
%also handle cases where the polyhedron is not solid in R^n, i.e.,
where the
%polyhedron is defined by both equality and inequality constraints.
%
%SYNTAX:
%
% [V,nr,nre]=lcon2vert(A,b,Aeq,beq,TOL)
%
%The rows of the N x n matrix V are a series of N vertices of the
polyhedron
%in R^n, defined by the linear constraints
%
% A*x <= b
% Aeq*x = beq
%
%By default, Aeq=beq=[], implying no equality constraints. The output
"nr"
%lists non-redundant inequality constraints, and "nre" lists non-
redundant
%equality constraints.
%
%The optional TOL argument is a tolerance used for both rank-
estimation and
%for testing feasibility of the equality constraints. Default=1e-10.
%The default can also be obtained by passing TOL=[];
%
%
%EXAMPLE:
%
%The 3D region defined by x+y+z=1, x>=0, y>=0, z>=0
%is described by the following constraint data.
%
%
% A =
%
%      0.4082    -0.8165     0.4082
%      0.4082     0.4082    -0.8165
%     -0.8165     0.4082     0.4082
%
%
% b =
%
%      0.4082
%      0.4082
%      0.4082
%
%
% Aeq =
%
%      0.5774     0.5774     0.5774
%
%
% beq =
%
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
%          0.5774
%
%
%
% >> V=lcon2vert(A,b,Aeq,beq)
%
%          V =
%
%          1.0000    0.0000    0.0000
%          0.0000    0.0000    1.0000
%          -0.0000    1.0000    0.0000
%
%
%
%%initial argument parsing

nre=[];
nr=[];
if nargin<5 || isempty(TOL), TOL=1e-10; end
if nargin<6, checkbounds=true; end

switch nargin

    case 0

        error 'At least 1 input argument required'

    case 1

        b=[]; Aeq=[]; beq=[];

    case 2

        Aeq=[]; beq=[];

    case 3

        beq=[];
        error 'Since argument Aeq specified, beq must also be
specified'

end

b=b(:); beq=beq(:);

if xor(isempty(A), isempty(b))
    error 'Since argument A specified, b must also be specified'
end

if xor(isempty(Aeq), isempty(beq))
    error 'Since argument Aeq specified, beq must also be
specified'
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
end

nn=max(size(A,2)*~isempty(A),size(Aeq,2)*~isempty(Aeq));

if ~isempty(A) && ~isempty(Aeq) && (size(A,2)~=nn ||
size(Aeq,2)~=nn)

    error 'A and Aeq must have the same number of columns if both
non-empty'

end

inequalityConstrained=~isempty(A);
equalityConstrained=~isempty(Aeq);

[A,b]=rownormalize(A,b);
[Aeq,beq]=rownormalize(Aeq,beq);

if equalityConstrained && nargout>2

    [discard,nre]=lindep([Aeq,beq].',TOL);

    if ~isempty(nre) %reduce the equality constraints

        Aeq=Aeq(nre,:);
        beq=beq(nre);

    else
        equalityConstrained=false;
    end

end

end

%%Find 1 solution to equality constraints within tolerance

if equalityConstrained

    Neq=null(Aeq);

    x0=pinv(Aeq)*beq;

    if norm(Aeq*x0-beq)>TOL*norm(beq), %infeasible

        nre=[]; nr=[]; %All constraints redundant for empty
polytopes
        V=[];
        return;
    end
end
```

```

elseif isempty(Neq)

    V=x0(:).';
    nre=(1:nn).'; %Equality constraints determine everything.
    nr=[];%All inequality constraints are therefore redundant.
    return

end

rkAeq= nn - size(Neq,2);

end

%%
if inequalityConstrained && equalityConstrained

    AAA=A*Neq;
    bbb=b-A*x0;

elseif inequalityConstrained

    AAA=A;
    bbb=b;

elseif equalityConstrained && ~inequalityConstrained

    error('Non-bounding constraints detected. (Consider box
constraints on variables.)')

end

nnn=size(AAA,2);

if nnn==1 %Special case

    idxu=sign(AAA)==1;
    idxl=sign(AAA)==-1;
    idx0=sign(AAA)==0;

    Q=bbb./AAA;
    U=Q;
    U(~idxu)=inf;
    L=Q;
    L(~idxl)=-inf;

    [ub,uloc]=min(U);
    [lb,lloc]=max(L);

    if ~all(bbb(idx0)>=0) || ub<lb %infeasible

        V=[]; nr=[]; nre=[];
        return
    end
end

```



```

elseif ~isfinite(ub) || ~isfinite(lb)

    error('Non-bounding constraints detected. (Consider box
constraints on variables.)')

end

Zt=[lb;ub];

if nargout>1
    nr=unique([lloc,uloc]); nr=nr(:);
end

else

    if nargout>1
        [Zt,nr]=con2vert(AAA,bbb,TOL,checkbounds);
    else
        Zt=con2vert(AAA,bbb,TOL,checkbounds);
    end

end

if equalityConstrained && ~isempty(Zt)

    V=bsxfun(@plus,Zt*Neq.',x0(:).');

else

    V=Zt;

end

if isempty(V),
    nr=[]; nre=[];
end

function [V,nr] = con2vert(A,b,TOL,checkbounds)
% CON2VERT - convert a convex set of constraint inequalities into the
set
%           of vertices at the intersections of those
inequalities;i.e.,
%           solve the "vertex enumeration" problem. Additionally,
%           identify redundant entries in the list of inequalities.
%
% V = con2vert(A,b)
% [V,nr] = con2vert(A,b)
%
% Converts the polytope (convex polygon, polyhedron, etc.) defined by
the
% system of inequalities A*x <= b into a list of vertices V. Each ROW
% of V is a vertex. For n variables:
% A = m x n matrix, where m >= n (m constraints, n variables)
% b = m x 1 vector (m constraints)

```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
% V = p x n matrix (p vertices, n variables)
% nr = list of the rows in A which are NOT redundant constraints
%
% NOTES: (1) This program employs a primal-dual polytope method.
%         (2) In dimensions higher than 2, redundant vertices can
%         appear using this method. This program detects
redundancies
%         at up to 6 digits of precision, then returns the
%         unique vertices.
%         (3) Non-bounding constraints give erroneous results;
therefore,
%         the program detects non-bounding constraints and returns
%         an error. You may wish to implement large "box"
constraints
%         on your variables if you need to induce bounding. For
example,
%         if x is a person's height in feet, the box constraint
%         -1 <= x <= 1000 would be a reasonable choice to induce
%         boundedness, since no possible solution for x would be
%         prohibited by the bounding box.
%         (4) This program requires that the feasible region have some
%         finite extent in all dimensions. For example, the
feasible
%         region cannot be a line segment in 2-D space, or a plane
%         in 3-D space.
%         (5) At least two dimensions are required.
%         (6) See companion function VERT2CON.
%         (7) ver 1.0: initial version, June 2005
%         (8) ver 1.1: enhanced redundancy checks, July 2005
%         (9) Written by Michael Kleder
%
%Modified by Matt Jacobson - March 30, 2011
%
```

```
%%3/4/2012 Improved boundedness test - unfortunately slower than
Michael Kleder's
```

```
if checkbounds
```

```
[aa,bb,aaeq,bbeq]=vert2lcon(A,TOL);
```

```
if any(bb<=0) || ~isempty(bbeq)
    error('Non-bounding constraints detected. (Consider box
constraints on variables.)')
end
```

```
clear aa bb aaeq bbeq
```

```
end
```

```
dim=size(A,2);
```

```
%%Matt J initialization
if strictinpoly(b,TOL)
```

```
    c=zeros(dim,1);
```

```
else
```

```

slackfun=@(c)b-A*c;

%Initializer0
c = pinv(A)*b; %02/17/2012 -replaced with pinv()
s=slackfun(c);

if ~approxinpoly(s,TOL) %Initializer1

    c=Initializer1(TOL,A,b,c);
    s=slackfun(c);

end

if ~approxinpoly(s,TOL) %Attempt refinement

    %disp 'It is unusually difficult to find an interior
point of your polytope. This may take some time... '
    %disp ' '

    c=Initializer2(TOL,A,b,c);
    %[c,fval]=Initializer1(TOL,A,b,c,10000);
    s=slackfun(c);

end

if ~approxinpoly(s,TOL)
    %error('Unable to locate a point near the
interior of the feasible region.')
    V=[];
    nr=[];
    return
end

if ~strictinpoly(s,TOL) %Added 02/17/2012 to handle
initializers too close to polytope surface

    %disp 'Recurring...'

    idx=( abs(s)<=max(s)*TOL );

    Amod=A; bmod=b;
    Amod(idx,:)=[];
    bmod(idx)=[];

    Aeq=A(idx,:); %pick the nearest face to c
    beq=b(idx);

    faceVertices=lcon2vert(Amod,bmod,Aeq,beq,TOL,1);
    if isempty(faceVertices)

```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
disp 'Something''s wrong. Couldn''t find face
vertices. Possibly polyhedron is unbounded.'
keyboard
end

c=faceVertices(1,:).'; %Take any vertex - find local
recession cone vector
s=slackfun(c);

idx=( abs(s)<=max(s)*TOL );

Asub=A(idx,:); bsub=b(idx,:);

[aa,bb,aaeq,bbeq]=vert2lcon(Asub);
aa=[aa;aaeq;-aaeq];
bb=[bb;bbeq;-bbeq];

clear aaeq bbeq

[bmin,idx]=min(bb);

if bmin>=-TOL
disp 'Something''s wrong. We should have found a
recession vector (bb<0).'
keyboard
end

Aeq2=null(aa(idx,:)).';
beq2=Aeq2*c; %find intersection of polytope with
line through facet centroid.

linetips = lcon2vert(A,b,Aeq2,beq2,TOL,1);

if size(linetips,1)<2
disp 'Failed to identify line segment through
interior.'
disp 'Possibly {x: Aeq*x=beq} has weak
intersection with interior({x: Ax<=b}).'
keyboard
end

lineCentroid=mean(linetips);%Relies on boundedness

clear aa bb

c=lineCentroid(:);
s=slackfun(c);

end

b = s;
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
end
%%end Matt J initialization

D=bsxfun(@rdivide,A,b);

k = convhulln(D);
nr = unique(k(:));

G = zeros(size(k,1),dim);
ee=ones(size(k,2),1);
discard=false(1, size(k,1) );

for ix = 1:size(k,1) %02/17/2012 - modified

    F = D(k(ix,:),:);
    if linddep(F,TOL)<dim;
        discard(ix)=1;
        continue;
    end

    G(ix,:)=F\ee;

end

G(discard,:)=[];

V = bsxfun(@plus, G, c. ');

[discard,I]=unique( round(V*1e6), 'rows' );
V=V(I,:);

return

function [c,fval]=Initializer1(TOL, A,b,c,maxIter)

    thresh=-10*max(eps(b));

    if nargin>4
        [c,fval]=fminsearch(@(x) max([thresh;A*x-b]),
c,optimset('MaxIter',maxIter));
    else
        [c,fval]=fminsearch(@(x) max([thresh;A*x-b]), c);
    end

return

function c=Initializer2(TOL,A,b,c)
%norm( (I-A*pinv(A))*(s-b) ) subj. to s>=0
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
maxIter=100000;

[mm,nn]=size(A);

Ap=pinv(A);
Aaug=speye(mm)-A*Ap;
Aaugt=Aaug.';

M=Aaugt*Aaug;
C=sum(abs(M),2);
C(C<=0)=min(C(C>0));

slack=b-A*c;
slack(slack<0)=0;

% relto=norm(b);
% relto =relto + (relto==0);
%
% relres=norm(A*c-b)/relto;

IterThresh=maxIter;
s=slack;
ii=0;
%for ii=1:maxIter
while ii<=2*maxIter %HARDCODE

    ii=ii+1;
    if ii>IterThresh,
        %warning 'This is taking a lot of iterations'
        IterThresh=IterThresh+maxIter;
    end

s=s-Aaugt*(Aaug*(s-b))./C;
s(s<0)=0;

c=Ap*(b-s);
%slack=b-A*c;
%relres=norm(slack)/relto;
%if all(0<slack,1)||relres<1e-6||ii==maxIter, break; end

end

return
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
function [r,idx,Xsub]=lindep(X,tol)
%Extract a linearly independent set of columns of a given matrix X
%
%   [r,idx,Xsub]=lindep(X)
%
%in:
%
%   X: The given input matrix
%   tol: A rank estimation tolerance. Default=1e-10
%
%out:
%
%   r: rank estimate
%   idx: Indices (into X) of linearly independent columns
%   Xsub: Extracted linearly independent columns of X

if ~nnz(X) %X has no non-zeros and hence no independent columns

    Xsub=[]; idx=[];
    return
end

if nargin<2, tol=1e-10; end

[Q, R, E] = qr(X,0);

diagr = abs(diag(R));

%Rank estimation
r = find(diagr >= tol*diagr(1), 1, 'last'); %rank estimation

if nargout>1
    idx=sort(E(1:r));
    idx=idx(:);
end

if nargout>2
    Xsub=X(:,idx);
end

function [A,b]=rownormalize(A,b)
%Modifies A,b data pair so that norm of rows of A is either 0 or 1

if isempty(A), return; end

normsA=sqrt(sum(A.^2,2));
idx=normsA>0;
A(idx,:)=bsxfun(@rdivide,A(idx,:),normsA(idx));
b(idx)=b(idx)./normsA(idx);
```

```
function tf=approxinpoly(s,TOL)
```

```
    smax=max(s);
```

```
    if smax<=0
        tf=false; return
    end
```

```
    tf=all(s>=-smax*TOL);
```

```
function tf=strictinpoly(s,TOL)
```

```
    smax=max(s);
```

```
    if smax<=0
        tf=false; return
    end
```

```
    tf=all(s>=smax*TOL);
```

```
function [A,b,Aeq,beq]=vert2lcon(V,tol)
```

```
%An extension of Michael Kleder's vert2con function, used for finding
the
```

```
%linear constraints defining a polyhedron in  $R^n$  given its vertices.
This
```

```
%wrapper extends the capabilities of vert2con to also handle cases
where the
```

```
%polyhedron is not solid in  $R^n$ , i.e., where the polyhedron is
defined by
```

```
%both equality and inequality constraints.
```

```
%
```

```
%SYNTAX:
```

```
%
```

```
% [A,b,Aeq,beq]=vert2lcon(V,TOL)
```

```
%
```

```
%The rows of the  $N \times n$  matrix V are a series of N vertices of a
polyhedron
```

```
%in  $R^n$ . TOL is a rank-estimation tolerance (Default = 1e-10).
```

```
%
```

```
%Any point x inside the polyhedron will/must satisfy
```

```
%
```

```
% A*x <= b
```

```
% Aeq*x = beq
```

```
%
```

```
%up to machine precision issues.
```

```
%
```

```
%
```

```
%EXAMPLE:
```

```
%
```



## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
%Consider V=eye(3) corresponding to the 3D region defined
%by x+y+z=1, x>=0, y>=0, z>=0.
%
%
%
% >>[A,b,Aeq,beq]=vert2lcon(eye(3))
%
%
%
% A =
%
%     0.4082    -0.8165     0.4082
%     0.4082     0.4082    -0.8165
%    -0.8165     0.4082     0.4082
%
%
%
% b =
%
%     0.4082
%     0.4082
%     0.4082
%
%
%
% Aeq =
%
%     0.5774     0.5774     0.5774
%
%
%
% beq =
%
%     0.5774
%
%
%
%%initial stuff
%
%
% if nargin<2, tol=1e-10; end
%
%
% [M,N]=size(V);
%
%
% if M==1
%     A=[];b=[];
%     Aeq=eye(N); beq=V(:);
%     return
% end
%
%
%
% p=V(1,:).';
% X=bsxfun(@minus,V.',p);
%
%
%
% %In the following, we need Q to be full column rank
% %and we prefer E compact.
%
%
% if M>N %X is wide
%
%     [Q, R, E] = qr(X,0); %economy-QR ensures that E is compact.
```

```

                                %Q automatically full column rank since X
wide

    else%X is tall, hence non-solid polytope

        [Q, R, P]=qr(X); %non-economy-QR so that Q is full-column rank.

        [~,E]=max(P); %No way to get E compact. This is the
alternative.
        clear P
    end

    diagr = abs(diag(R));

    if nnz(diagr)

        %Rank estimation
        r = find(diagr >= tol*diagr(1), 1, 'last'); %rank estimation

        iE=1:length(E);
        iE(E)=iE;

        Rsub=R(1:r,iE).';

        if r>1

            [A,b]=vert2con(Rsub,tol);

        elseif r==1

            A=[1;-1];
            b=[max(Rsub);-min(Rsub)];

        end

        A=A*Q(:,1:r).';
        b=bsxfun(@plus,b,A*p);

        if r<N
            Aeq=Q(:,r+1:end).';
            beq=Aeq*p;
        else
            Aeq=[];
            beq=[];
        end

    else %Rank=0. All points are identical

        A=[]; b=[];
        Aeq=eye(N);
        beq=p;
    end

```

```

end

%           ibeq=abs (beq) ;
%           ibeq(~beq)=1;
%
%           Aeq=bsxfun (@rdivide,Aeq,ibeq) ;
%           beq=beq./ibeq;

function [A,b] = vert2con(V,tol)
% VERT2CON - convert a set of points to the set of inequality
constraints
%           which most tightly contain the points; i.e., create
%           constraints to bound the convex hull of the given points
%
% [A,b] = vert2con(V)
%
% V = a set of points, each ROW of which is one point
% A,b = a set of constraints such that A*x <= b defines
%       the region of space enclosing the convex hull of
%       the given points
%
% For n dimensions:
% V = p x n matrix (p vertices, n dimensions)
% A = m x n matrix (m constraints, n dimensions)
% b = m x 1 vector (m constraints)
%
% NOTES: (1) In higher dimensions, duplicate constraints can
%         appear. This program detects duplicates at up to 6
%         digits of precision, then returns the unique
constraints.
%         (2) See companion function CON2VERT.
%         (3) ver 1.0: initial version, June 2005.
%         (4) ver 1.1: enhanced redundancy checks, July 2005
%         (5) Written by Michael Kleder,
%
% Modified by Matt Jacobson - March 29,2011
%

k = convhulln(V);
c = mean(V(unique(k),:));

V = bsxfun (@minus,V,c);
A = nan(size(k,1),size(V,2));

dim=size(V,2);
ee=ones(size(k,2),1);
rc=0;

for ix = 1:size(k,1)
    F = V(k(ix,:),:);
    if linddep(F,tol) == dim
        rc=rc+1;
        A(rc,:)=F\ee;
    end
end

```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
    end
end

A=A(1:rc,:);
b=ones(size(A,1),1);
b=b+A*c';

% eliminate duplicate constraints:
[A,b]=rownormalize(A,b);
[discard,I]=unique( round([A,b]*1e6),'rows');

A=A(I,:); % NOTE: rounding is NOT done for actual returned results
b=b(I);
return
```

```
function [A,b]=rownormalize(A,b)
%Modifies A,b data pair so that norm of rows of A is either 0 or 1
```

```
    if isempty(A), return; end

    normsA=sqrt(sum(A.^2,2));
    idx=normsA>0;
    A(idx,:)=bsxfun(@rdivide,A(idx,:),normsA(idx));
    b(idx)=b(idx)./normsA(idx);
```

```
function [r,idx,Xsub]=lindep(X,tol)
%Extract a linearly independent set of columns of a given matrix X
%
%   [r,idx,Xsub]=lindep(X)
%
%in:
%
%   X: The given input matrix
%   tol: A rank estimation tolerance. Default=1e-10
%
%out:
%
%   r: rank estimate
%   idx: Indices (into X) of linearly independent columns
%   Xsub: Extracted linearly independent columns of X
```

```
    if ~nnz(X) %X has no non-zeros and hence no independent columns

        Xsub=[]; idx=[];
        return
    end
```

```
    if nargin<2, tol=1e-10; end
```

```
    [Q, R, E] = qr(X,0);

    diagr = abs(diag(R));
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
%Rank estimation
r = find(diagr >= tol*diagr(1), 1, 'last'); %rank estimation

if nargout>1
    idx=sort(E(1:r));
    idx=idx(:);
end

if nargout>2
    Xsub=X(:,idx);
end

function [varargout]=qlcon2vert(x0,varargin)
%A quicker version of lcon2vert that will skip some intensive steps
when a
%point x0 in the relative interior of the polyhedron is known a
priori.
%
% [V,nr,nre]=qlcon2vert(x0, A,b,Aeq,beq,TOL)
%
%Here x0 is the known relative interior point. The rest of the
input/output
%arguments are as in LCON2VERT.
%
%In the interest of speed, QLCON2VERT will perform minimal error
checking,
%shifting the burden to the user of verifying that the input data has
%appropriate properties. In particular, it will not try to verify
that x0
%is in fact in the relative interior. By default also, qlcon2vert
will skip the
%boundedness test, and presume the user knows the polyhedron to be
bounded already.
%However, you can re-activate the boundedness test by calling with
the syntax,
%
% [V,nr,nre]=qlcon2vert(x0, A,b,Aeq,beq,TOL,1)

if nargin<=3
    varargin(3:6)={[],[],[],0};

end

x0=x0(:);

[A,b,Aeq,beq]=deal(varargin{1:4});

b=b-A*x0;
```

## ΠΑΡΑΡΤΗΜΑ-ΚΩΔΙΚΑΣ MATLAB

```
if ~isempty(beq)
    beq(:)=0;
end

varargin{2}=b;
varargin{4}=beq;

[varargout{1:max(nargout,1)}] = lcon2vert(varargin{:});

V=varargout{1};

V=bsxfun(@plus,x0.',V); %undo coordinate translation

varargout{1}=V;
```

# Βιβλιογραφία

- [1] : R.Horst , H.Tuy “Global Optimization : Deterministic Approaches”, 3<sup>rd</sup> ed. , Springer 1995.
- [2] : C.A. Floudas “Deterministic Global Optimization : Theory, Methods and applications” , Kluwer Academic Publishers 2000.
- [3] : P.Hartman “On functions representable as difference of convex functions”, Pacific Journal of Mathematics, 9, 1959.
- [4] : T.Q.Phong , P.D.Tao , L.T.H. An “A method for solving D.C. Programming Problems:Application to fuel mixture nonconvex optimization problems”, Journal of Global Optimization , 6 , 1995
- [5] : E. Asplund “Differentiability of the Metric Projection in Finite Dimensional Euclidean Space” , Proceedings of the American Mathematical Society , 38 , 1973
- [6] : C.D.Maranas , C.A.Floudas “Global Minimum Potential Energy Conformations of Small Molecules”, Journal of Global Optimization , 4 , 1994
- [7] : S.Boyd , L.Vandenberghe “Convex Optimization” Cambridge University Press , 2004
- [8] : Σ.Κιναλής “Επίλυση συστημάτων εξισώσεων κυρτών συναρτήσεων με χρήση αλγορίθμου ολικής βελτιστοποίησης” Εθνικό Μετσόβιο Πολυτεχνείο, 2013
- [9] : K.L.Hoffman “A method for globally minimizing concave functions over convex sets” , Mathematical Programming, 20, North Holland Publishing Company , 1981
- [10]: S.A.Vavasis “Nonlinear Optimization, Complexity Issues”, Oxford University Press,1991
- [11]:R.Horst “On the global minimization of concave functions”, University of Oldenburg,1984
- [12]:H.P.Benson, R.Horst “A branch and bound-outer approximation algorithm for concave minimization over a convex set”, Computers and mathematics with applications, vol.21 , 1991