Βελτιστοποίηση – Εργαστηριακός Οδηγός
Σχολή Θετικών Επιστημών – Τμήμα Πληροφορικής

Διδάσκων: Ι. Η. Λαγαρής

1η εργαστηριακή άσκηση

Με την άσκηση αυτή θα προσπαθήσουμε να εγκαταστήσουμε το πακέτο βελτιστοποίησης Merlin και να έρθουμε σε μία πρώτη επαφή με το περιβάλλον χρήσης του. Το Merlin είναι ένα εργαλείο που αναπτύχθηκε στο τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων και ενσωματώνει διάφορες τεχνικές ολικής και μερικής ελαχιστοποίησης πραγματικών συναρτήσεων. Το Merlin έχει σχεδιαστεί για να τρέχει σε περιβάλλον Unix, είναι όμως εφικτή η παραμετροποίησή του για να εκτελείται και σε περιβάλλον DOS/Windows. Τέλος να αναφέρουμε ότι η γλώσσα επικοινωνίας μεταξύ χρήστη και Merlin είναι η FORTRAN, πράγμα που σημαίνει ότι προκειμένου να είναι εφικτή η ελαχιστοποίηση μίας συνάρτησης, αυτή πρέπει πρώτα να κωδικοποιηθεί σε FORTRAN.

1.1 Εγκατάσταση του Merlin

Ο κώδικας του πακέτου, όπως επίσης και τεκμηρίωση του διατίθενται ελεύθερα από την ιστοσελίδα http://merlin.cs.uoi.gr/. Τα βήματα που πρέπει να ακολουθήσει κάποιος για να εγκαταστήσει το Merlin είναι τα ακόλουθα:

1. wget http://merlin.cs.uoi.gr/files/merlin-x.y.z.tar.gz, όπου x.y.z είναι η έκδοση του Merlin που θα εγκαταστήσετε (π.χ. 3.1.1)
2. gunzip merlin-x.y.z.tar.gz
3. tar –xvf merlin- x.y.z.tar
4. rm merlin- x.y.z.tar
1. cd merlin-x.y.z
2. vi Makefile.inc
3. Κάνετε τις εξής αλλαγές:
    1. DESTDIR = $(HOME)/merlin-dist
    2. F77=g77
4. Αποθηκεύστε τις αλλαγές και βγείτε από τον vi (πατήστε ESC και εν συνεχεία ":wq" <enter>)
5. mkdir ~/merlin-dist
6. make (για να μεταφραστεί ο πηγαίος κώδικας)
7. make install (για να γίνει η εγκατάσταση)

8. Προσθέστε την παρακάτω γραμμή στο .profile (ή .bashrc αν χρησιμοποιήτε το bash shell) αρχείο που έχετε, για να βρίσκει το κέλυφος τη διαδρομή εκτέλεσης του Merlin:
   export PATH=$PATH:$HOME/merlin-dist/bin:

   Αν δεν προσθέσετε την παραπάνω εντολή θα πρέπει κάθε φορά να δίνετε την πλήρη διαδρομή για το εκτελέσιμο αρχείο του Merlin (δηλαδή ~/merlin-dist/bin/run-merlin)

1.2 Μία πρώτη γνωριμία με το Merlin

Όπως έχει προαναφερθεί το Merlin είναι ένα εργαλείο για ελαχιστοποίηση συναρτήσεων. Προκειμένου να είναι εφικτή η χρήση του, θα πρέπει ο χρήστης με κάποιο τρόπο να κωδικοποιήσει τη συνάρτηση που θέλει να μελετήσει (objective function). Συνοπτικά να αναφέρουμε ότι το Merlin αναμένει ως είσοδο από τον χρήστη την παρακάτω πληροφορία:

- Αντικειμενική συνάρτηση (υποχρεωτικό όρισμα)
- Το διάνυσμα της παραγώγου σε κάθε σημείο
- Τον Ιακωβιανό πίνακα
- Τον Εσσιανό πίνακα

Να θυμίσουμε και πάλι ότι η γλώσσα επικοινωνίας με το Merlin είναι η FORTRAN. Τέλος θα επιχειρήσουμε να δείξουμε τον τρόπο χρήσης του Merlin μέσα από ένα απλό παράδειγμα.

Ας υποθέσουμε λοιπόν ότι η αντικειμενική συνάρτηση που μελετάμε είναι η $f(x_1,x_2) = x_1^2 + (x_2 - 1)^4$. Είναι προφανές ότι η συνάρτηση αυτή έχει ελάχιστη τιμή το $f(0,1) = 0$. Για να την προσδιορίσουμε ως αντικειμενική θα πρέπει να ορίσουμε τη συνάρτηση FUNMIN σε FORTRAN. Η FUNMIN εν ολίγοις είναι η αντικειμενική συνάρτηση που μελετάμε κωδικοποιημένη με όρους FORTRAN, στο παράδειγμά μας:

```
function funmin (x,n)
implicit double precision (a-h,o-z)
        dimension x(n)
        funmin = x(1)**2 + (x(2)-1)**4
end
```

Πέραν της συναρτήσεως FUNMIN, υπάρχει και ένας άλλος τρόπος να εισάγουμε την αντικειμενική συνάρτηση στο Merlin, και αυτός είναι η ρουτίνα SUBSUM. Η SUBSUM μπορεί να χρησιμοποιηθεί εναλλακτικά της FUNMIN στην περίπτωση που η αντικειμενική συνάρτηση μπορεί να γραφεί και ως άθροισμα τετραγώνων, οπότε μας δίνει μία ευελιξία και δυνατότητα να χρησιμοποιήσουμε μία ευρεία γκάμα τεχνικών που αναφέρονται σε αυτού του είδους τις συναρτήσεις. Στο παράδειγμα μας ισχύει $f(x_1,x_2) = (x_1)^2 + \left((x_2 - 1)^2\right)^2 = y_1^2 + y_2^2$, όπου $y1 = x_1$ και $y_2 = (x_2 - 1)^2$ και ο κώδικας σε FORTRAN είναι ο ακόλουθος:

```
subroutine subsum (m,n,x,f)
 implicit double precision (a-h,o-z)
        dimension x(n), f(m)
        f(1) = x(1)
        f(2) = (x(2)-1)**2
 end
```

Έχοντας ορίσει τη συνάρτηση που θέλουμε να ελαχιστοποιήσουμε μπορούμε πλέον να χρησιμοποιήσουμε το Merlin για να βρούμε τοπικά ή / και ολικά ελάχιστα. Ωστόσο όπως θα γνωρίζεται και από τη θεωρία, κάθε τεχνική ελαχιστοποίησης απαιτεί και επιπλέον πληροφορία προκειμένου να μπορεί να εκτελεστεί, και η πληροφορία αυτή δεν είναι άλλη από την πρώτη και δεύτερη παράγωγο της αντικειμενικής συνάρτησης σε ένα σημείο x_0. Έτσι λοιπόν υπάρχουν και οι ακόλουθες τρεις συναρτήσεις τις οποίες το Merlin χρησιμοποιεί κατ' απαίτηση της μεθόδου ελαχιστοποίησης όποτε χρειάζεται να γίνει υπολογισμός της παραγώγου, Να σημειωθεί ότι η κωδικοποίηση των τριών συναρτήσεων που θα αναφερθούν δεν είναι υποχρεωτική, καθότι σε περίπτωση μη εισαγωγής τους, το Merlin ακολουθεί μία αριθμητική διαδικασία για τον υπολογισμό των παραγώγων. Ωστόσο η ποιότητα του αποτελέσματος, όπως είναι φυσικό αλλοιώνεται και ενδεχομένως να μην έχει την επιθυμητή ακρίβεια. Στο παράδειγμά μας ο κώδικας για τις τρεις συναρτήσεις είναι ο παρακάτω:

```
subroutine granal (n,x,grad)
implicit double precision (a-h,o-z)
        dimension x(n), grad(n)
        grad(1) = 2 * x(1)
        grad(2) = 4 * (x(2)-1)**3
end
```

granal: διάνυσμα παραγώγου στο σημείο x, δηλαδή $grad(i) = \frac{df}{dx_i}$. Ο χρήστης πρέπει να καθορίσει με την εντολή ***anal*** ότι για τον υπολογισμό της παραγώγου το Merlin θα χρησιμοποιεί τη ρουτίνα granal.

```
subroutine janal (m,n,x,fj,ld)
implicit double precision (a-h,o-z)
        dimension x(n), fj(ld,n)
        fj(1,1) = 1
        fj(1,2) = 0
        fj(2,1) = 0
        fj(2,2) = 2 * (x(2)-1)
end
```

janal: ο Ιακωβιανός πίνακας, δηλαδή $fj(i,j) = \frac{\partial f_i}{\partial x_j}$. Η ρουτίνα αυτή χρησιμοποιείται από μεθόδους ελαχιστοποίησης όταν η αντικειμενική συνάρτηση εκφράζεται ως άθροισμα τετραγώνων. Ο χρήστης πρέπει να καθορίσει με την εντολή ***janal*** ότι για τον υπολογισμό του Ιακωβιακού πίνακα το Merlin θα χρησιμοποιεί τη ρουτίνα janal.

```
subroutine hanal (h,ld,n,x)
implicit double precision (a-h,o-z)
        dimension x(n),h(ld,n)
        h(1,1) = 2
        h(1,2) = 0
        h(2,1) = 0
        h(2,2) = 12 * (x(2)-1)**2
end
```

hanal: ο Εσσιανός πίνακας – δεύτερες παράγωγοι τις συνάρτησης, δηλαδή $h(i,j) = \frac{\partial^2 f}{\partial x_i \partial x_j}$. Όπως είναι προφανές, ο Εσσιανός πίνακας είναι συμμετρικός. Έτσι ο χρήστης δεν έχει παρά να εισάγει μόνο τον κάτω τριγωνικό πίνακα, ενώ τον υπόλοιπο τον συμπληρώνει από μόνο του το Merlin. Ο χρήστης πρέπει να καθορίσει με την εντολή ***hessian*** ότι για τον υπολογισμό του Εσσιανού πίνακα το Merlin θα χρησιμοποιεί τη ρουτίνα hanal.

Έχοντας λοιπόν κωδικοποιήσει όλη την απαραίτητη για την ελαχιστοποίηση πληροφορία σε ένα αρχείο, έστω το test.f, γράφοντας την εντολή ***run-merlin test.f*** στη γραμμή εντολών, εκκινείτε το περιβάλλον εργασίας του Merlin με αντικειμενική συνάρτηση αυτή που περιγράφεται στο αρχείο test.f. Αρχικά το Merlin θα σας προτρέψει να καθορίσετε τη διάσταση του προβλήματος (δηλαδή τη διάσταση της ανεξάρτητης μεταβλητής) και το πλήθος των όρων στο άθροισμα τετραγώνων (στην περίπτωση που δεν έχετε εισάγει τη ρουτίνα SUBSUM, απλά δίνεται μηδέν). Στην περίπτωσή μας η είσοδος θα είναι 2 (διάσταση x) 2 (όροι στο άθροισμα τετραγώνων). Πλέον το μπορούμε να χρησιμοποιήσουμε οποιαδήποτε τεχνική υλοποιεί το Merlin και ταιριάζει στο πρόβλημά μας για να εκτελέσουμε την ελαχιστοποίηση.

Κλείνοντας την παρουσίαση να σημειώσουμε ότι το Merlin διαθέτει εσωτερική βιβλιοθήκη τεκμηρίωσης για τις υποστηριζόμενες εντολές. Η χρήση της εντολής ***help <όνομα εντολής>*** δίνει πληροφορίες για την εντολή. Επιπλέον με την εντολή ***list*** προβάλλεται στον χρήστη μία λίστα με αλφαβητική κατάταξη όλων των εντολών του Merlin.

<u>2η εργαστηριακή άσκηση</u>

2.1 Διαχείριση των παραμέτρων ελαχιστοποίησης

Έχοντας δει στην προηγούμενη άσκηση τον τρόπο με τον οποίο μπορούμε να κωδικοποιήσουμε την αντικειμενική συνάρτηση στο Merlin, θα προχωρήσουμε στην 2η εργαστηριακή άσκηση στην επεξήγηση ορισμένων βασικών εντολών του Merlin για τη διαχείριση των παραμέτρων. Οι εντολές που θα μελετήσουμε τις εξής εντολές:

- POINT: ανάθεση τιμών στις παραμέτρους
- LMARGIN: καθορισμός κάτω ορίου στο εύρος τιμών μίας παραμέτρου
- RMARGIN: καθορισμός άνω ορίου στο εύρος τιμών μίας παραμέτρου
- LDEMARGIN: απαλοιφή κάτω ορίου στο εύρος τιμών μίας παραμέτρου
- RDEMARGIN: απαλοιφή άνω ορίου στο εύρος τιμών μίας παραμέτρου
- FIX: Καθήλωση των τιμών μίας παραμέτρου
- LOOSE: Απαλοιφή καθηλωμένης τιμής μίας παραμέτρου
- FIXALL: Καθήλωση των τιμών όλων των παραμέτρων
- LOOSALL: Απαλοιφή όλων καθηλωμένων τιμών των παραμέτρων
- SHORTDIS: Προβολή πληροφοριών (τιμές, εύροι διακύμανσης, κ.τ.λ.) για όλες τις παραμέτρους.

<u>Παραδείγματα:</u>

1. point 1 10.2 3 23.87 5-9 0.2 11- 3.3
Αναθέτει στην 1η παράμετρο την τιμή 10.2, στην 3η την τιμή 23.87, στις παραμέτρους 5 έως 9 την τιμή 0.2 και σε όλες τις παραμέτρους από την 11η και πέρα την τιμή 3.3.

2. lmargin 1 10.2 2-4 1.1
Ορίζει στην 1η παράμετρο κάτω φράγμα την τιμή 10.2 και στις παραμέτρους 2 έως 4 την τιμή 1.1 ως κάτω φράγμα.

3. rdemargin 1-5
Απαλείφει το άνω φράγμα για τις παραμέτρους 1 έως 5.

4. fix 4 6-10
Καθηλώνει τις παραμέτρους 4 και 6 έως 10 στις τρέχουσες τιμές τους.

5. loose 3 4-7
Απαλοιφή των καθηλωμένων τιμών για τις παραμέτρους 3 και 4 έως 7.

ΕΡΩΤΗΣΕΙΣ:

Έχοντας ως αναφορά την αντικειμενική συνάρτηση που είδαμε στην άσκηση 1, απαντήστε στις παρακάτω ερωτήσεις. Όπου σας ζητείται να εφαρμόσετε έναν αλγόριθμο ελαχιστοποίησης χρησιμοποιήστε την εντολή auto, χωρίς να καθορίσετε κάποια επιπλέον αλλαγή στις ιδιότητες της μεθόδου.

1. Δώστε τη σειρά εντολών που πρέπει να εισαχθούν στο Merlin, προκειμένου να εντοπίσετε ένα ελάχιστο της συνάρτησης για $x_1 \in [-10, -5], x_2 \in [2,5]$.

....
....
....
....
....
....

2. Με ποιο τρόπο μπορείτε να χρησιμοποιήσετε το Merlin για να εντοπίσετε ένα τοπικό μέγιστο της αντικειμενικής συνάρτησης; Γράψτε το τοπικό μέγιστο που εντοπίσατε.

....
....
....
....
....
....

3. Δώστε την ακολουθία εντολών που εντοπίζει ένα τοπικό ελάχιστο της αντικειμενικής συνάρτησης δεδομένου ότι $x_1 = 2, x_2 \in [-1,1]$.

....
....
....
....
....
....
....
....
....
....

4. Δεδομένου ότι $x_1 = 3$ για ποια τιμή του x_2 η αντικειμενική συνάρτηση εμφανίζει ένα τοπικό ελάχιστο. Αντίστοιχα όταν $x_2 = 3$ για ποια τιμή του x_1 η αντικειμενική συνάρτηση εμφανίζει τοπικό ελάχιστο; Αναφέρατε τη σειρά εντολών που χρησιμοποιήσατε στο Merlin για να εκτελέσετε τον υπολογισμό.

....
....
....
....
....
....
....
....
....

3η εργαστηριακή άσκηση

3.1 Ελαχιστοποίηση στο Merlin (I)

Ο χρήστης του Merlin έχει πρόσβαση στις τεχνικές ελαχιστοποίησης του Merlin με κλήση των παρακάτω εντολών:

BFGS, DFP, TRUST, TOLMIN, CONGRA, ROLL, SIMPLEX, LEVE, AUTO

Όλες οι εντολές με την κλήση τους δίνουν τη δυνατότητα στο χρήστη να ρυθμίσει τις παραμέτρους εισόδου και εξόδου του αντίστοιχου αλγορίθμου. Μεταξύ άλλων οι βασικές κοινές παράμετροι είναι:

- NOC : Μία προσέγγιση του πάνω ορίου στο πλήθος κλήσεων της αντικειμενικής συνάρτησης κατά τη διάρκεια της ελαχιστοποίησης.
- PRINT: Καθορίζει την έξοδο της μεθόδου (τί θα δει ο χρήστης ως αποτέλεσμα). Επιτρεπόμενες τιμές:
  0. Καμία εκτύπωση
  1. Εκτύπωση των κατώτερων τιμών για τη συνάρτηση καθώς εντοπίζονται κατά τη διάρκεια εκτέλεσης του αλγορίθμου ελαχιστοποίησης.

  Να σημειωθεί ότι με το πέρας εκτέλεσης του αλγορίθμου ελαχιστοποίησης εμφανίζονται στον χρήστη κάποια στατιστικά σχετικά με το πλήθος κλήσεων διαφόρων συναρτήσεων (π.χ. της FUNMIN).

Όσον αφορά στον υπολογισμό των παραγώγων, όπως έχουμε δει και στις προηγούμενες ασκήσεις, ο χρήστης έχει τη δυνατότητα να καθορίζει, μέσω των αντίστοιχων συναρτήσεων, τον τρόπο υπολογισμού των τελευταίων αναλυτικά. Ωστόσο το Merlin έχει τη δυνατότητα να υπολογίζει – προσεγγίζει τις παραγώγους με χρήση πεπερασμένων διαφορών. Τρεις επιλογές είναι διαθέσιμες για την περίπτωση αυτή: FAST, QUAD, NUMER (κατά αύξουσα απαίτηση σε υπολογιστική ισχύ και σε ακρίβεια προσέγγισης). Για τον έλεγχο της ακρίβειας των παραγώγων, υπάρχει η εντολή GRADIS, η οποία παρουσιάζει τα σφάλματα υπολογισμού παραγώγων μεταξύ των διαφόρων μεθόδων. Για παράδειγμα, αν η τρέχουσα κατάσταση υπολογισμού είναι ANAL και καλέσουμε την εντολή GRADIS FAST θα δούμε τα σφάλαματα υπολογισμού παραγώγων μεταξύ κατάστασης υπολογισμού ANAL και FAST.

Τέλος να σημειωθεί ότι η μέθοδος AUTO δεν υλοποιεί κάποιον ιδιαίτερο αλγόριθμο, απλά εφαρμόζει όλες τις ενσωματωμένες μεθόδους του Merlin και επιλέγει το καλύτερο αποτέλεσμα.

3.2 Μέθοδοι ελαχιστοποίησης – Οι εντολές BFGS και TOLMIN

Η μέθοδος BFGS (*Broyden-Fletcher-Goldfarb-Shanno*) χρησιμοποιείται για την επίλυση μη γραμμικών προβλημάτων ελαχιστοποίησης χωρίς περιορισμούς. Είναι μία μέθοδος γραμμικής αναζήτησης στην οποία ο Εσσιανός πίνακας αναλύεται σε άνω και κάτω τριγωνικό με τη μέθοδο Choleski, δηλαδή $H = LL^T$. Περισσότερες πληροφορίες μπορείτε για τη μέθοδο μπορείτε να βρείτε στην Wikipedia, στο λήμμα BFGS (*http://en.wikipedia.org/wiki/BFGS*).

Η μέθοδος TOLMIN είναι ακριβώς ίδια με την BFGS, με μόνη διαφορά ότι για τον υπολογισμό του Εσσιανού πίνακα χρησιμοποιείται η ανάλυση Goldfarb-Idnani $Z^T BZ = I$.

3.3 Μέθοδοι ελαχιστοποίησης – Ελαχιστοποίηση αθροίσματος τετραγώνων. Η εντολή LEVE

Η μέθοδος LEVE του Merlin, υλοποιεί τον αλγόριθμο Levenberg-Marquardt και ενδείκνυται για ελαχιστοποίηση συναρτήσεων που γράφονται ως αθροίσματα τετραγώνων. Επομένως είναι απαραίτητο να έχετε ορίσει και τη συνάρτηση SUBSUM αν σκοπεύετε να χρησιμοποιήσετε την εντολή LEVE. Τέλος, επειδή η μέθοδος χρησιμοποιεί και τον Ιακωβιανό πίνακα, συνιστάται η κωδικοποίησή και της αντίστοιχης ρουτίνας στο Merlin, εφόσον είναι εφικτός ο αναλυτικός υπολογισμός των ζητούμενων παραγώγων.

Περισσότερες πληροφορίες μπορείτε να βρείτε στην ιστοσελίδα της Wikipedia, στο λήμμα Levenberg-Marquardt algorithm (*http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm*)

ΕΡΩΤΗΣΕΙΣ:

Για τις ερωτήσεις που ακολουθούν, θα έχετε ως αναφορά την αντικειμενική συνάρτηση

$$f(x_1,x_2) = x_1^2\left(1-\sigma\upsilon\nu^2 x_2\right) + x_2^2\left(2\sigma\upsilon\nu^2 x_1 - 2\sigma\upsilon\nu x_1 + \eta\mu^2 x_1\right),\ x_1,x_2 \in \left(0,\frac{\pi}{2}\right)$$

1. Δώστε μία μορφή αθροίσματος τετραγώνων για την αντικειμενική συνάρτηση. Υπολογίστε την παράγωγο, τον Ιακωβιανό και τον Εσσιανό πίνακα της νέας αντικειμενικής συνάρτησης και υλοποιήστε τις αντίστοιχες ρουτίνες χρήστη του Merlin.

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

2. Χρησιμοποιήστε τις μεθόδους που περιγράψαμε στη σημερινή άσκηση και υπολογίστε ένα τοπικό ακρότατο της f. Επιλέξτε τυχαία ένα αρχικό σημείο κοινό για όλες τις μεθόδους. Ποια έδωσε καλύτερο αποτέλεσμα (συγκρίνετε την τιμή του ελαχίστου, το πλήθος κλήσεων της αντικειμενικής συνάρτησης και τον χρόνο εκτέλεσης); Χρησιμοποιήστε τις συναρτήσεις που υλοποιήσατε για τον υπολογισμό των διαφόρων παραγώγων.

....
....
....
....
....
....
....
....

....
....
....
....
....
....
....
....
....
....
....
....
....

3. Επαναλάβετε το πείραμα της 2ης ερώτησης, αλλά αυτή τη φορά αφήστε το Merlin να υπολογίσει τις διάφορες παραγώγους αριθμητικά (Ορίστε σε FAST την κατάσταση λειτουργίας αριθμητικού υπολογισμού παραγώγων).
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

....
....

4η εργαστηριακή άσκηση

4.1 Ελαχιστοποίηση στο Merlin (II)

Στην προηγούμενη άσκηση είδαμε τις μεθόδους που διαθέτει το Merlin για ελαχιστοποίηση. Στη σημερινή άσκηση θα συνεχίσουμε τη μελέτη της ελαχιστοποίησης στο περιβάλλον εργασίας του Merlin. Όλες οι αναφερόμενες μέθοδοι ελαχιστοποίησης είναι αριθμητικές. Αυτό σημαίνει ότι ξεκινώντας από μία αρχική τιμή των παραμέτρων, εφαρμόζουν μία διαδικασία επαναληπτικά για να εντοπίσουν μία νέα τιμή των παραμέτρων πιο κοντά στο ελάχιστο. Η διαδικασία επαναλαμβάνεται μέχρις ότου ένα κριτήριο σύγκλισης ικανοποιηθεί. Επομένως γίνεται κατανοητό ότι όλες οι προαναφερθείσες μέθοδοι υπολογίζουν τοπικά ελάχιστα. Μία κοινή πρακτική για να βελτιώσουμε την απόδοση (= εύρεση ενός τοπικού ελαχίστου πιο κοντά στο ολικό) είναι να εκτελούμε πολλές φορές μία μέθοδο και να κρατάμε κάθε φορά το καλύτερο ελάχιστο.

4.2 Μέθοδοι ελαχιστοποίησης – Η εντολή DFP

Η μέθοδος DFP (*Davidon-Fletcher-Powell*),όπως και η TOLMIN, είναι ίδια με την BFGS, Είναι δηλαδή και αυτή μία μέθοδος γραμμικής αναζήτησης σε φθίνουσα κατεύθυνση. Όπως και η BFGS, ο Εσσιανός πίνακας αναλύεται με τη μέθοδο Choleski.
Περισσότερες πληροφορίες μπορείτε για τη μέθοδο μπορείτε να βρείτε στην Wikipedia, στο λήμμα Davidon-Fletcher-Powell_formula (*http://en.wikipedia.org/wiki/Davidon-Fletcher-Powell_formula*).

4.2 Μέθοδοι ελαχιστοποίησης – Η εντολή TRUST

Η εντολή TRUST, υλοποιεί έναν αλγόριθμο περιοχών εμπιστοσύνης (trust region) καλώντας την BFGS και έχοντας αναλύσει τον Εσσιανό πίνακα με τη μέθοδο Choleski. Οι μέθοδοι περιοχών εμπιστοσύνησς βασίζονται στο γεγονός ότι προσπαθούν να προσεγγίσουν την αντικειμενική συνάρτηση σε μία περιοχή με ένα τετραγωνικό μοντέλο, έτσι ώστε να εφαρμόσουν ελαχιστοποίηση στο μοντέλο αυτό αντί της αρχικής συνάρτησης. Ο λόγος που γίνεται αυτό είναι γιατί η ελαχιστοποίηση τετραγωνικών συναρτήσεων είναι πιο εύκολη και αποδοτική. Αν η προσέγγιση σε μία περιοχή είναι καλή τότε επεκτείνονται τα όρια της περιοχής, σε αντίθετη περίπτωση συρικνώνονται.
Περισσότερες πληροφορίες μπορείτε για μεθόδους περιοχών εμπιστοσύνης μπορείτε να βρείτε στην Wikipedia, στο λήμμα Trust_region (*http://en.wikipedia.org/wiki/Trust_region*).

4.3 Μέθοδοι ελαχιστοποίησης – Η εντολή CONGRA

Η εντολή CONGRA, υλοποιεί τον αλγόριθμο των συζυγών κλίσεων (CONjugate GRAdient) για τον υπολογισμό του ελαχίστου. Να θυμίσουμε ότι οι συζυγείς κλίσεις είναι μία μέθοδος γραμμικής αναζήτησης, όπου η φθίνουσα κατεύθυνση δίνεται από τον τύπο:

$$s^{(k)} = -\nabla f^{(k)} + \beta^{(k)} s^{(k-1)}, \quad \beta^{(k)} \in \mathbb{R}$$

Περισσότερες πληροφορίες μπορείτε για μεθόδους συζυγών κλίσεων μπορείτε να βρείτε στην Wikipedia, στο λήμμα Conjugate_gradient (*http://en.wikipedia.org/wiki/Conjugate_gradient*).
Στο Merlin υλοποιούνται τρεις αλγόριθμοι που σχετίζονται με συζυγείς κλίσεις:

- Ο αλγόριθμος Polak-Ribiere
- Ο αλγόριθμος Fletcher-Reeves
- Ο γενικευμένος αλγόριθμος Polak-Ribiere

Για νακαθορίσετε ποιός από τους παραπάνω αλγορίθμους θα χρησιμοποιηθεί ορίζετε κατάλληλα την παράμετρο METHOD της CONGRA. Οι επιτρεπόμενες τιμές είναι:

- PR για τον αλγόριθμο Polak-Ribiere

- FR για τον αλγόριθμο Fletcher-Reeves
- GPR για τον γενικευμένο αλγόριθμο Polak-Ribiere

<u>ΕΡΩΤΗΣΕΙΣ:</u>

Για τις ερωτήσεις που ακολουθούν χρησιμοποιήστε την παρακάτω αντικειμενική συνάρτηση:

$$f(x)=\sum_{i=1}^{3}\left[x_i-10\sigma\upsilon\nu(2\pi x_i)\right],\ \ x=[x_1,x_2,x_3]^T\in[-1,1]$$

1. Υπολογίστε την παράγωγο και τον Εσσιανό πίνακα της αντικειμενικής συνάρτησης, και υλοποιήστε τις αντίστοιχες ρουτίνες στο Merlin.

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

2. Υπολογίστε ένα τοπικό ακρότατο της αντικειμενικής συνάρτησης ξεκινώντας από ένα τυχαίο αρχικό σημείο. Επαναλάβετε το πείραμα πολλές φορές. Καταγράψτε το αρχικό σημείο και το ακρότατο που υπολογίσατε. Χρησιμοποιήστε αναλυτικό υπολογισμό των παραγώγων. Στην περίπτωση της CONGRA, χρησιμοποιήστε όλες τις διαθέσιμες μεθόδους.

....

5η εργαστηριακή άσκηση

5.1 Ελαχιστοποίηση στο Merlin (III)

Μέχρι τώρα μελετήσαμε μεθόδους ελαχιστοποίησης που χρησιμοποιούν παραγώγους για τον εντοπισμό του ελαχίστου. Στη σημερινή άσκηση θα ολοκληρώσουμε τη επισκόπηση των μεθόδων ελαχιστοποίησης του Merlin με μία κατηγορία μεθόδων που ακολουθούν διαφορετική προσέγγιση (χωρίς τη χρήση παραγώγων).

5.2 Μέθοδοι ελαχιστοποίησης – Η εντολή SIMPLEX

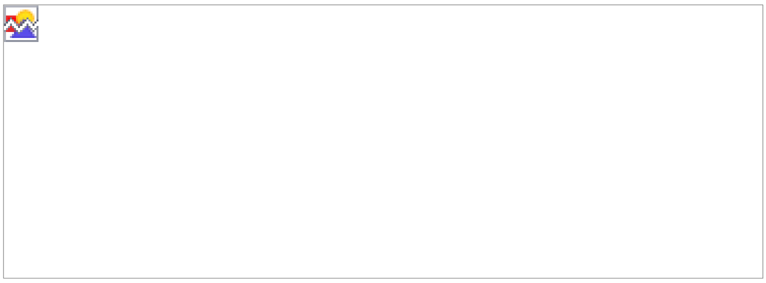
Η μέθοδος SIMPLEX ανήκει στην κλάση των μεθόδων για μη γραμμική βελτιστοποίηση, και δε θα πρέπει να συγχέεται με την SIMPLEX μέθοδο του γραμμικού προγραμματισμού. Βασίζεται στην χρήση ενός SIMPLEX, έννοια δανεισμένη από την τοπολογία και αντιστοιχεί σε ένα πολύτοπο με N+1 κόμβους στον N-διάστατο χώρο. Έτσι στην περίπτωση που N = 1, το πολύτοπο αντιστοιχεί σε ευθεία γραμμή, όταν N=2 το πολύτοπο γίνεται τρίγωνο, όταν N=3 μιλάμε για πολύτοπα που είναι τετράεδρα και ούτο καθ' εξής. Η βασική ιδέα είναι να φέρουμε ένα SIMPLEX στην περιοχή του ελαχίστου προσαρμόζοντάς το στην τοπική γεωμετρία του χώρου και να προσπαθήσουμε να το συρικνώσουμε γύρω από την περιοχή του ελαχίστου. Να σημειωθεί ότι ο εν λόγω αλγόριθμος δεν χρησιμοποιεί υπολογισμό παραγώγων.

5.3 Μέθοδοι ελαχιστοποίησης – Η εντολή ROLL

Η μέθοδος ROLL ανήκει στην κατηγορία μεθόδων που αναζητούν πρότυπα Η μέθοδος εκτελείται εξετάζοντας την τοπική γεωμετρία της αντικειμενικής συνάρτησης κάνοντας κατάλληλα βήματα προς κάθε κατεύθυνση χωριστά. Όπως και η SIMPLEX, η μέθοδος ROLL δεν χρησιμοποιεί παραγώγους στον υπολογισμό του ελαχίστου.

ΕΡΩΤΗΣΕΙΣ:

Για τις ακόλουθες αντικειμενικές συναρτήσεις βρείτε ένα τοπικό ελάχιστο, χρησιμοποιώντας μία από τις διαθέσιμες μεθόδους του Merlin. Ποια μέθοδος ταιριάζει καλύτερα στο κάθε πρόβλημα; Γράψτε τα απαραίτητα βήματα για να χρησιμοποιήσετε την μέθοδο που επιλέξατε.



6η εργαστηριακή άσκηση

6.1 Ενσωμάτωση του Merlin σε προγράμματα χρήστη – η συνάρτηση OPTIMA

Στις ασκήσεις που είδαμε έως τώρα, ακολουθήσαμε την ίδια διαδικασία για τη χρήση του Merlin σε προβλήματα βελτιστοποίησης. Δηλαδή, αρχικά υλοποιούσαμε τις διάφορες ρουτίνες που απαιτεί η χρήση του Merlin και εν συνεχεία σε γραμμή εντολών, εκτελούσαμε την ελαχιστοποίηση, οπότε και παίρναμε το αποτέλεσμα μας στο περιβάλλον εργασίας του Merlin. Θα δούμε στη σημερινή άσκηση πώς μπορούμε να καλέσουμε το περιβάλλον εργασίας του Merlin μέσα από προγράμματα που γράφει ο ίδιος ο χρήστης και να χρησιμοποιήσουμε το αποτέλεσμα για περαιτέρω πράξεις. Αυτό γίνεται μέσω της ρουτίνας OPTIMA, το πρωτότυπο της οποίας είναι το ακόλουθο:

```
SUBROUTINE OPTIMA (N,M,XP,XV,XLL,XRL,IXAT,ICODE,FINP,FOUT,GRMS,NF,NG,NH,NJ)
```

όπου:

N: Το πλήθος των παραμέτρων.

M: Το πλήθος των όρων στο άθροισμα τετραγώνων.

XP: Το XP είναι ένας πίνακας που αρχικά έχει το σημείο εκκίνησης (αρχικοποίηση μεθόδου) ενώ με το πέρας της εκτέλεσης περιέχει την τιμή του ελαχιστοποιητή (POINT).

XV: Με το πέρας της εκτέλεσης το XV είναι η τιμή του ελαχίστου που βρήκαμε.

XLL: Πίνακας που περιέχει το κάτω φράγμα για κάθε παράμετρο (LMARGIN)

XRL: Πίνακας που περιέχει το άνω φράγμα για κάθε παράμετρο (RMARGIN).

IXAT: Πίνακας που περιέχει τις τιμές στις οποίες καθηλώνεται μία μεταβλητή (FIX).

ICODE: Ένας πίνακας ακεραίων 4 θέσεων, όπου:

1. Αν ICODE(1) = 1, τότε οι τιμές του πίνακα XP θα χρησιμοποιηθούν για αρχικοποίηση των παραμέτρων.
2. Αν ICODE(2) = 1, τότε οι τιμές του πίνακα XLL θα χρησιμοποιηθούν ως κάτω φράγματα για τις παραμέτρους.
3. Αν ICODE(3) = 1, τότε οι τιμές του πίνακα XRL θα χρησιμοποιηθούν ως άνω φράγματα για τις παραμέτρους.
4. Αν ICODE(4) = 1, τότε οι τιμές του πίνακα IXAT θα χρησιμοποιηθούν ως για να καθηλωθούν οι παράμετροι.

Αν κάποια τιμή του ICODE είναι μηδέν, τότε οι αντίστοιχη ενέργεια δεν εκτελείται.

FINP: Η διαδρομή του αρχείου με τις εντολές που θα εκτελεστούν από το Merlin για να γίνει η ελαχιστοποίηση. Είναι στην ουσία οι εντολέςπου θα δίναμε στη γραμμή εντολών προκειμένου να γίνει η εκτέλεση της ελαχιστοποίησης.

FOUT: Η διαδρομή του αρχείου στο οποίο θα αποθηκευτεί η έξοδος του Merlin.
Η τιμή /dev/null στο UNIX έχει ως αποτέλεσμα να μην αποθηκευτεί κάπου η έξοδος, ενώ η τιμή ' ' οδηγεί την έξοδο του Merlin στην standard output συσκευή.

GRMS: Με το πέρας της ελαχιστοποίησης στην GRMS επιστρέφεται το μέσο τετραγωνικό σφάλμα στην τιμή της παραγώγου για το σημείο XP.

NF: Πλήθος κλήσεων αντικειμενικής συνάρτησης.

NG: Πλήθος εκτιμήσεων της παραγώγου (π.χ. κλήσεων συνάρτησης granal).

NH: Πλήθος εκτιμήσεων του Εσσιανού πίνακα. (π.χ. κλήσεων συνάρτησης hanal).

NJ: Πλήθος εκτιμήσεων του Ιακωβιανού πίνακα (π.χ. κλήσεων συνάρτησης janal).

ΠΡΟΣΟΧΗ:

Να τονίσουμε ότι και πάλι θα πρέπει να κωδικοποιηθούν οι αντίστοιχες ρουτίνες του Merlin, FUNMIN/SUBSUM, GRANAL, JANAL, HANAL.

ΕΡΩΤΗΣΕΙΣ:

Υλοποιήστε ένα πρόγραμμα στο οποίο αφού έχετε καθορίσει την αντικειμενική συνάρτηση, τις παραγώγους τις κ.τ.λ. (έχετε δηλαδή ορίσει τις απαραίτητες ρουτίνες στο Merlin) θα καλείτε επαναληπτικά μία μέθοδο ελαχιστοποίησης, και κάθε φορά θα κρατάτε το ελάχιστο που βρήκατε. Σε κάθε επανάληψη θα ορίζετε ένα νεο σημείο αρχικοποίησης των παραμέτρων τυχαία. Μόλις οι επαναλήψεις φτάσουν ένα μέγιστο όριο τότε το πρόγραμμα θα σταματάει την εκτέλεσή του και θα επιστρέφει το καλύτερο ελάχιστο που έχει βρει. Ο μέγιστος αριθμός επαναλήψεων θα δίνετε με την εκκίνηση του προγράμματος από τον χρήστη, μέσω πληκτρολογίου. Ως αντικειμενική συνάρτηση χρησιμοποιήστε μία από αυταίς που υλοποιήσατε στο προηγούμενο εργαστήριο.

Μερικές οδηγίες για Fortran:

1. Για να διαβάσετε από το πληκτρολόγιο έναν ακέραιο αριθμό και να τον αποθηκεύσετε στην μεταβλητή N χρησιμοποιήτε την εντολή READ (*,*) N
2. Για να τυπώσετε έναν πραγματικό αριθμό R στην οθόνη χρησιμοποιήστε την εντολή WRITE (*,*) R.
3. Δομή επανάληψης σε Fortran:

   Do <label> I=1,N

   ... statements ...

   <label> CONTINUE

   Το <label> είναι μια ετικέτα (π.χ. ένας αριθμός) που καθορίζει τη γραμμή στην οποία γράφεται. Έτσι στον παραπάνω βρόχο, τα *statments* θα εκτελεστούν N φορές, δηλαδή ότι βρίσκεται από τη γραμμή του *Do* μέχρι τη γραμμή με την ετικέτα *<label>* θα εκτελεστεί για όσο ισχύει η συνθήκη του Do.
4. Τυχαίοι αριθμοί στο [0,1] επιστρέφονται με τη συνάρτηση rand().

Βιβλιογραφία

[1] Merlin-3.0 A multidimensional optimizatio environment, D.G. Papageorgiou, I. N. Demetropoulos, I. E. Lagris, Computer Physics Communications 109 (1998) p. 227-249.
[2] PANMIN: Sequential and Parallel Global Optimization Procedures with a variety of options for the Local Search Strategy Computer Physics Communications 159 (2004) 63-69 (with D. G. Papageorgiou and F. V. Theos) .
[3] Numerical Optimization, Jorge Nocedal, Stephen J. Wright, εκδόσεις Springer
[4] Wikipedia, http://en.wikipedia.org

<u>Σύντομος οδηγός για Fortran 77</u>

1. Τύποι δεδομένων
2. Οι βασικές εντολές
3. Συναρτήσεις
4. Είσοδος – Έξοδος
5. Καθολικές μεταβλητές
6. Παράδειγμα κώδικα