MERLIN - LC

Εγχειρίδιο χρήσης

Κεφάλαιο 1

Εισαγωγή

1.1 Τι προβλήματα λύνει το Merlin

Η ελαχιστοποίηση μιάς πολυδιάστατης συνάρτησης είναι ένα πρόβλημα κοινό σε πολλά ερευνητικά πεδία. Πλήθος προβλημάτων στους τομείς της μηχανολογίας, φυσικής, χημείας κα., ανάγονται συχνά στην επίλυση ενός προβλήματος ελαχιστοποίησης. Ως παραδείγματα αναφέρουμε την επίλυση συστημάτων μη γραμμικών εξισώσεων, παραλακτικές μεθόδους και την προσαρμογή δεδομένων σε αναλυτικές συναρτήσεις, που είναι μιά καθημερινή ανάγκη στην πειραματική έρευνα. Οι δυσκολίες της πολυδιάστατης ελαχιστοποίησης είναι πολλές και ποικίλες (ο αναγνώστης παραπέμπεται για ενημέρωση στις αναφορές [5-10]). Είναι γνωστό ότι δεν υπάρχει μέθοδος που να αντιμετωπίζει ικανοποιητικά όλα τα προβλήματα. Η συνάρτηση μπορεί να μην έχει γνωστή αναλυτική μορφή, οπότε οι παράγωγοί της πρέπει να εκτιμηθούν αριθμητικά, πράγμα που κοστίζει σε χρόνο υπολογισμού και ακρίβεια. Στην περίπτωση που ταυτόχρονα απαιτείται να ικανοποιούνται γνωστές συνθήκες μεταξύ των μεταβλητών το πρόβλημα αποκτά περιορισμούς και βέβαια γίνεται πιό δύσκολο.

Το Merlin είναι ένα ολοκληρωμένο περιβάλλον που κατασκευάστηκε για την επίλυση προβλημάτων ελαχιστοποίησης στοχεύοντας ταυτόχρονα να είναι φιλικό στη χρήση, φορητό μεταξύ διαφορετικών υπολογιστικών συστημάτων και εύκολα επεκτάσιμο. Αντιμετωπίζει προβλήματα ελαχιστοποίησης της μορφής:

Εύρεση τοπικού ελαχίστου της συνάρτησης:

$$f(\mathbf{x}) \qquad \mathbf{x} \in \Re^N \qquad \mathbf{x} = [\, x_1, x_2, \ldots x_N \,]^T$$

υπό τους περιορισμούς ορίων:

$$a_i \le x_i \le b_i \qquad i = 1, 2, \ldots N$$

Ειδική μέριμνα έχει ληφθεί για την αντιμετώπιση προβλημάτων όπου η αντικειμενική συνάρτηση είναι της μορφής:

Εύρεση τοπικού ελαχίστου της συνάρτησης:

$$f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x})^2 \qquad \mathbf{x} \in \Re^N \qquad \mathbf{x} = [\, x_1, x_2, \ldots x_N \,]^T$$

υπό τους περιορισμούς ορίων:

$$a_i \le x_i \le b_i \qquad i = 1, 2, \ldots N$$

Επιπλέον η τρέχουσα έκδοση Merlin-LC μπορεί να αντιμετωπίσει προβλήματα με γραμμικούς περιρισμούς της μορφής:

Εύρεση τοπικού ελαχίστου της συνάρτησης:

$f(\mathbf{x})$ $\quad \mathbf{x} \in \Re^N \quad$ $\mathbf{x} = [\, x_1, x_2, \ldots x_N \,]^T$

υπό τους γραμμικούς περιορισμούς:

$\mathbf{c}_j^T \mathbf{x} = \mathbf{b}_j \quad j=1, \ldots meq$

$\mathbf{c}_j^T \mathbf{x} \leq \mathbf{b}_j \quad j=meq+1, \ldots m$

Ο χρήστης καθοδηγεί το πρόγραμμα εισάγοντας εντολές, ενώ παρέχεται πλήθος βοηθητικών λειτουργιών. Διαθέτει τη δική του γλώσσα προγραμματισμού (MCL) που επιτρέπει την ανάπτυξη αποτελεσματικών στρατηγικών ελαχιστοποίησης γιά την αντιμετώπιση σύνθετων προβλημάτων. Η κωδικοποίηση σε ANSI-Fortran-77 [10] εξασφαλίζει την φορητότητα μεταξύ διαφορετικών υπολογιστικών συστημάτων. Πλήρεις οδηγίες εγκατάστασης του προγράμματος δίνονται στα συνοδευτικά αρχεία. Το Merlin υποστηρίζεται πλήρως, ακόμη και μέσω e-mail. Η διεύθυνση είναι: Merlin@cc.uoi.gr

1.2 Συμβολισμός

Για την περιγραφή του Merlin χρησιμοποιούνται οι ακόλουθες συμβάσεις:

α) Γραφή

Τα μικρά έντονα γράμματα (**x**, **g** κλπ.) δηλώνουν διανύσματα στο χώρο των N διαστάσεων R^N, ενώ τα έντονα κεφαλαία (**G**, **H** κλπ.) δηλώνουν πίνακες.

Οι εντολές του Merlin, κώδικας Fortran και είσοδος/έξοδος του υπολογιστή δείχνονται με τη χρήση `μη αναλογικής γραφής Courier`.

Έντονα γράμματα **`μη αναλογικής γραφής Courier`** στα παραδείγματα και τους πίνακες, δείχνουν την είσοδο του χρήστη.

β) Μαθηματικός συμβολισμός

$f(\mathbf{x})$	Η αντικειμενική συνάρτηση. Τα στοιχεία x_i αναφέρονται ως μεταβλητές ή παράμετροι της συνάρτησης.
$f_i(\mathbf{x})$	Οι επιμέρους όροι της αντικειμενικής συνάρτησης, όταν αυτή αποτελεί άθροισμα τετραγώνων.
g	Το διάνυσμα των πρώτων παραγώγων της συνάρτησης με στοιχεία $g_i = \frac{\partial f}{\partial x_i}$
$\mathbf{c}_j$	Τα διανύσματα των γραμμικών περιορισμών με j=1,... m
G	Ο πίνακας των δεύτερων παραγώγων της συνάρτησης με στοιχεία $G_{ij} = \frac{\partial^2 f}{\partial x_i \, \partial x_i}$
H	Ο αντίστροφος του πίνακα των δευτέρων παραγώγων με στοιχεία $H_{ij} = \left[\mathbf{G}^{-1}\right]_{ij}$
a_i, b_i	Τα κάτω και άνω όρια των παραμέτρων x_i αντίστοιχα. Αναφέρονται και ως αριστερά και δεξιά όρια.
N	Ο αριθμός των παραμέτρων της συνάρτησης $f(\mathbf{x})$.
M	Ο αριθμός των όρων της συνάρτησης $f(\mathbf{x})$, όταν αυτή έχει την ειδική μορφή του αθροίσματος τετραγώνων.
meq	Ο αριθμός των γραμμικών περιορισμών που δίνονται ως ισότητες.
m	Συνολικός αριθμός των γραμμικών περιορισμών.

γ) Άλλες τυπογραφικές συμβάσεις

`<cr>`	Δηλώνει πάτημα του πλήκτρου Return.
`<space>`	Δηλώνει τον κενό χαρακτήρα.
...	Δηλώνει ότι τα σύμβολα που προηγούνται μπορεί να επαναληφθούν.
[1]	Οι αριθμοί σε αγκύλες αποτελούν βιβλιογραφικές αναφορές.

Κεφάλαιο 2

Τι πρέπει να προγραμματίσει ο χρήστης

Το Merlin έχει σχεδιαστεί έτσι ώστε να είναι φορητό και εύκολα προσαρμόσιμο σε ένα πλήθος δυνατών εφαρμογών. Πρίν τη χρήση του πακέτου ο χρήστης ανάλογα με τις ανάγκες του, πρέπει να προετοιμάσει (σε μορφή πηγαίου κώδικα) μιά σειρά από υποπρογράμματα.

2.1 Η αντικειμενική συνάρτηση

Πρόκειται για τη συνάρτηση της οποίας αναζητούμε ελάχιστο. Πρέπει να δηλωθεί ώς:

```
FUNCTION FUNMIN ( X, N )
DIMENSION X(N)
```

Τα ορίσματα της συνάρτησης είναι:

X	- είσοδος -	Μονοδιάστατος πίνακας με τις τιμές των παραμέτρων x_i.
N	- είσοδος -	Ο αριθμός των παραμέτρων της συνάρτησης.
FUNMIN	- έξοδος -	Η τιμή της αντικειμενικής συνάρτησης f(**x**).

Υπόδειγμα της παραπάνω συνάρτησης funmin γιά τη συνάρτηση Rosenbrock [13]:

$$f(x_1, x_2) = 100\,(x_2 - x_1^2)^2 + (1-x_1)^2$$

φαίνεται στον πίνακα 1.

Πίνακας 1: Υπόδειγμα της συνάρτησης funmin. Έχει κωδικοποιηθεί η συνάρτηση Rosenbrock.

```
FUNCTION FUNMIN ( X, N )
DIMENSION X(N)
X1 = X(1)
X2 = X(2)
FUNMIN = 100.*(X2-X1**2)**2 + (1.-X1)**2
END
```

2.2 Οι παράγωγοι

Εάν είναι αναλυτικά γνωστές οι πρώτες παράγωγοι της συνάρτησης (όλες, ή ορισμένες μόνο), αυτές μπορεί να προγραμματιστούν:

```
SUBROUTINE GRANAL ( N, X, GRAD )
DIMENSION X(N), GRAD(N)
```

Τα ορίσματα του υποπρογράμματος είναι:

N	- είσοδος -	Ο αριθμός των παραμέτρων της συνάρτησης.
X	- είσοδος -	Μονοδιάστατος πίνακας με τις τιμές των παραμέτρων x_i.
GRAD	- έξοδος -	Μονοδιάστατος πίνακας με τις τιμές των πρώτων παραγώγων g_i.

Υπόδειγμα του υποπρογράμματος granal γιά τη συνάρτηση Rosenbrock δίνεται στον πίνακα 2. Σημειώστε ότι πολλές φορές δεν είναι δυνατόν να έχουμε αναλυτική μορφή παρά μόνο γιά μερικές μόνο από τις παραγώγους της συνάρτησης. Σε αυτή την περίπτωση θα πρέπει να προγραμματίσουμε όσες γνωρίζουμε, ενώ οι υπόλοιπες θα παραλειφθούν. Η ύπαρξη αναλυτικής μορφής γιά μερικές από τις παραγώγους δηλώνεται κατά τη διάρκεια της χρήσης του Merlin με την εντολή `MIXED` (βλέπε παράγραφο 4.1).

Εάν καμία από τις παραγώγους της συνάρτησης δεν είναι αναλυτικά γνωστή, το υποπρόγραμμα granal πρέπει να έχει την τετριμμένη μορφή:

```
SUBROUTINE GRANAL ( N, X, GRAD )
DIMENSION X(N), GRAD(N)
END
```

Πίνακας 2: Υπόδειγμα του υποπρογράμματος granal. Έχουν κωδικοποιηθεί οι παράγωγοι της συνάρτησης Rosenbrock.

```
      SUBROUTINE GRANAL ( N, X, GRAD )
      DIMENSION X(N), GRAD(N)
      X1 = X(1)
      X2 = X(2)
      A = X2-X1*X1
      GRAD(1) = -400.*A*X1 - 2.*(1.-X1)
      GRAD(2) = 200.*A
      END
```

2.3 Ειδική μορφή της αντικειμενικής συνάρτησης Άθροισμα τετραγώνων

Εάν η αντικειμενική συνάρτηση είναι της μορφής:

$$f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x})^2$$

μπορεί να προγραμματιστεί και ως υποπρόγραμμα που θα επιστρέφει τις επιμέρους τιμές f_i :

```
SUBROUTINE SUBSUM ( M, N, X, F )
DIMENSION X(N), F(M)
```

Τα ορίσματα του υποπρογράμματος είναι:

M	- είσοδος -	Ο αριθμός των επιμέρους όρων της συνάρτησης.
N	- είσοδος -	Ο αριθμός των παραμέτρων της συνάρτησης.
X	- είσοδος -	Μονοδιάστατος πίνακας με τις τιμές των παραμέτρων x_i.
F	- έξοδος -	Μονοδιάστατος πίνακας με τις τιμές των επιμέρους όρων f_i.

Υπόδειγμα του υποπρογράμματος subsum γιά τη συνάρτηση Rosenbrock δίνεται στον πίνακα 3. Μόνο μία από τις `FUNCTION FUNMIN` ή `SUBROUTINE SUBSUM` πρέπει να προγραμματιστεί. Σε περίπτωση που η αντικειμενική συνάρτηση είναι άθροισμα τετραγώνων και προγραμματίσουμε την `SUBROUTINE SUBSUM` θα πρέπει αυτό να δηλωθεί κατά τη χρήση του Merlin μέσω της εντολής `SOS` (βλέπε παράγραφο 5.2), ενώ η `FUNCTION FUNMIN` θα πρέπει να έχει τη μορφή:

```
FUNCTION FUNMIN ( X, N )
DIMENSION X(N)
END
```

Σημειώστε ότι αν προγραμματιστεί η `SUBROUTINE SUBSUM`, όλοι οι ελαχιστοποιητές του Merlin μπορούν να τη χρησιμοποιήσουν. Η `FUNCTION FUNMIN` δεν μπορεί να χρησιμοποιηθεί από τον ελαχιστοποιητή `LEVE` (βλέπε παράγραφο 9.11), ειδικό για συναρτήσεις που είναι αθροίσματα τετραγώνων.

Εάν η συνάρτηση δεν είναι άθροισμα τετραγώνων και προγραμματίσουμε τη συνάρτηση funmin, τότε το υποπρόγραμμα subsum πρέπει να έχει τη μορφή:

```
SUBROUTINE SUBSUM ( M, N, X, F )
DIMENSION X(N), F(M)
END
```

Πίνακας 3: Υπόδειγμα του υποπρογράμματος subsum. Έχουν κωδικοποιηθεί οι επιμέρους όροι f_i της συνάρτησης Rosenbrock.

```
      SUBROUTINE SUBSUM ( M, N, X, F )
      DIMENSION X(N), F(M)
      X1 = X(1)
      X2 = X(2)
      F(1) = 10.*(X2-X1**2)
      F(2) = 1.-X1
      END
```

2.4 Ο πίνακας των δευτέρων παραγώγων

Εάν είναι αναλυτικά γνωστές οι δεύτερες παράγωγοι της αντικειμενικής συνάρτησης, αυτές μπορεί να προγραμματιστούν:

```
      SUBROUTINE HANAL ( H, LD, N, X )
      DIMENSION H(LD,N) X(N)
```

Τα ορίσματα του υποπρογράμματος είναι:

H	- έξοδος -	Ο πίνακας των δευτέρων παραγώγων της συνάρτησης.
LD	- είσοδος -	Η πρώτη διάσταση το πίνακα H, όπως έχει δηλωθεί από το καλών υποπρόγραμμα.
N	- είσοδος -	Ο αριθμός των παραμέτρων της συνάρτησης.
X	- είσοδος -	Μονοδιάστατος πίνακας με τις τιμές των παραμέτρων x_i.

Υπόδειγμα του υποπρογράμματος hanal για τη συνάρτηση Rosenbrock δίνεται στον πίνακα 4. Σημειώστε ότι πρέπει να συμπληρωθεί μόνο το κάτω τριγωνικό μέρος του H. Εάν οι δεύτερες παράγωγοι δεν είναι αναλυτικά γνωστές, τότε το υποπρόγραμμα hanal παίρνει την τετριμένη μορφή:

```
      SUBROUTINE HANAL ( H, LD, N, X )
      DIMENSION H(LD,N) X(N)
      END
```

Πίνακας 4: Υπόδειγμα του υποπρογράμματος hanal. Έχουν κωδικοποιηθεί οι δεύτερες παράγωγοι της συνάρτησης Rosenbrock.

```
      SUBROUTINE HANAL (H, LD, N, X )
      DIMENSION H(LD,N), X(N)
      X1 = X(1)
      X2 = X(2)
      H(1,1) = -400*(X2-3*X1**2) + 2
      H(2,2) = 200
      H(2,1) = -400*X1
      END
```

2.5 Οι γραμμικοί περιορισμοί

Εάν υπάρχουν γραμμικοί περιορισμοί, αυτοί γίνονται γνωστοί μέσω του υποπρογράμματος ulcdat που έχει τη μορφή:

```
      SUBROUTINE ULCDAT ( M, MEQ, IA, A, B )
      DIMENSION A(IA,*), B(*)
```

Τα ορίσματα του υποπρογράμματος είναι:

M	- έξοδος -	Ο συνολικός αριθμός των γραμμικών περιορισμών.
MEQ	- έξοδος -	Ο αριθμός των γραμμικών περιοισμών που δίνονται ως ισότητες.
IA	- είσοδος -	Η πρώτη διάσταση του πίνακα A όπως έχει δηλωθεί από το καλών πρόγραμμα.
A	- έξοδος -	Ο πίνακας των γραμμικών περιοισμών.
B	- έξοδος -	Μονοδιάστατος πίνακας με τις σταθερές b_i.

Υπόδειγμα του υποπρογράμματος ULCDAT για τους περιορισμούς:

$$x_1 + x_2 + x_3 + x_4 = 1$$
$$-(x_1 + x_2) \le -2$$

φαίνεται στον πίνακα 5. Εάν δεν υπάρχουν γραμμικοί περιορισμοί, τότε το υποπρόγραμμα ulcdat παίρνει την τετριμμένη μορφή:

```
      SUBROUTINE ULCDAT ( M, MEQ, IA, A, B )
      DIMENSION A(IA,*), B(*)
      M = 0
      MEQ = 0
      END
```

Πίνακας 5: Υπόδειγμα του υποπρογράμματος ulcdat.

```
      SUBROUTINE ULCDAT ( M, MEQ, IA, A, B )
      DIMENSION A(IA,*), B(*)
      M = 2
      MEQ = 1
      DO 1, J=1,M
      DO 1, I=1,IA
1     A(I,J) = 0
      DO 2,I=1,4
2     A(I,1) = 1
      A(1,2) = -1
      A(2,2) = -1
      B(1) = 1
      B(2) = -2
      END
```

2.6 Το κυρίως πρόγραμμα

Το Merlin έχει κατασκευαστεί ώς υποπρόγραμμα, έτσι ώστε να είναι δυνατή η κλήση του για την επίλυση προβλημάτων ελαχιστοποίησης μέσα από άλλα προγράμματα. Είναι δηλωμένο ώς:

```
SUBROUTINE MERLIN ( N, M, VERSIM, MAXW, IQUIT )
DIMENSION VERSIM(MAXW)
```

Τα ορίσματα του υποπρογράμματος είναι:

N	- είσοδος -	Ο αριθμός των παραμέτρων της συνάρτησης.
M	- είσοδος -	Ο αριθμός των επιμέρους όρων της συνάρτησης, όταν αυτή αποτελεί άθροισμα τετραγώνων (διαφορετικά πρέπει να θέσουμε M=0).
VERSIM	- είσοδος -	Μονοδιάστατος πίνακας μήκους MAXW που χρησιμοποιείται ώς βοηθητικός χώρος αποθήκευσης από τις μεθόδους BFGS, DFP, TOLMIN, LEVE και SIMPLEX. Το ποσό μνήμης που απαιτείται γιά κάθε μία από αυτές σαν συνάρτηση των N και M φαίνεται στον πίνακα 6.
MAXW	- είσοδος -	Το μέγεθος του βοηθητικού πίνακα VERSIM.
IQUIT	- έξοδος -	Ακέραια μεταβλητή. Η τιμή της καθορίζεται από το χρήστη όταν δοθεί η εντολή QUIT.

Υπόδειγμα κλήσης του υποπρογράμματος MERLIN δίνεται στον πίνακα 7.

Πίνακας 6: Απαιτούμενος χώρος μνήμης για τις διάφορες μεθόδους ελαχιστοποίησης (N = αριθμός παραμέτρων, M = αριθμός όρων στο άθροισμα τετραγώνων)

Μέθοδος	Απαιτούμενες λέξεις μνήμης
BFGS	$N^2 + N$
DFP	$N^2 + N$
SIMPLEX	$N^2 + N$
LEVE	$N\ M$
TRUST	$N^2 + N$
TOLMIN	$N^2 + 11\ N$

Πίνακας 7: Υπόδειγμα κυρίως προγράμματος.

```
      PROGRAM MASTER
C  MXV = MAXIMUM NUMBER OF PARAMETERS FOR THE OBJECTIVE FUNCTION.
      PARAMETER ( MXV = 200 )
      PARAMETER ( MAXW = MXV**2+11*MXV )
      DIMENSION VERSIM(MAXW)

      WRITE (*,*) 'ENTER # 0F VARIABLES'
      READ (*,*) N
      WRITE (*,*) 'ENTER # 0F TERMS'
      READ (*,*) M

      CALL MERLIN(N,M,VERSIM,MAXW,IQUIT)
      END
```

Κεφάλαιο 3

Βασική χρήση του Merlin

3.1 Το λειτουργικό σύστημα του Merlin

Όπως έχει ήδη αναφερθεί το Merlin αποτελεί ένα διαλογικό πρόγραμμα που κατευθύνεται από το χρήστη μέσω εντολών. Το λειτουργικό σύστημα του Merlin, δέχεται τις εντολές που εισάγει ο χρήστης, τις αναγνωρίζει, και ενεργεί ανάλογα. Αναγνωρίζονται τα εξής είδη εντολών:

α) Απλές εντολές. Πρόκειται για το βασικό σύνολο εντολών του Merlin που περιγράφονται στα επόμενα κεφάλαια. Μπορούν να εισαχθούν και συντομευμένες στον ελάχιστο αριθμό χαρακτήρων που τις κάνει να ξεχωρίζουν από τις υπόλοιπες. Πχ. η εντολή `SHORTDIS`, μπορεί να εισαχθεί ως `SHORT`, `SHO` ή `SH`, όχι όμως ως `S`, αφού θα προκύψει σύγχιση με άλλες εντολές όπως `STEP`, `SIMPLEX`, κα.

β) Μετονομασίες εντολών. Εξηγούνται στην παράγραφο 5.7.

γ) Κλήσεις παραμετρικών εντολοπακέτων. Εξηγούνται στο κεφάλαιο 8.

3.2 Τρόποι αναφοράς στις παραμέτρους της αντικειμενικής συνάρτησης

Κατά τη διάρκεια χρήσης του Merlin, υπάρχει ένας ομοιόμορφος τρόπος αναφοράς στις παραμέτρους της αντικειμενικής συνάρτησης. Όπου λοιπόν χρειάζεται αυτές αναφέρονται με δύο τρόπους:

α) Με τη χρήση ενός ακέραιου δείκτη που παίρνει τιμές από 1 έως N. Ο δείκτης 1 αναφέρεται στην πρώτη παράμετρο x_1, ο δείκτης 2 στην δεύτερη x_2 , κ.ο.κ.

β) Με τη χρήση ενός συμβολικού ονόματος. Κάθε μία από τις παραμέτρους της συνάρτησης μπορεί να αποκτήσει ένα συμβολικό όνομα με τη χρήση της εντολής `GODFATHER` (βλέπε παράγραφο 5.9). Πχ. η πρώτη παράμετρος μπορεί να ονομαστεί `RHO` και η δεύτερη `SIGMA`. Τα συμβολικά ονόματα μπορεί να έχουν μέγεθος μέχρι 10 χαρακτήρες και η επιλογή τους αφήνεται στη απόλυτη διάθεση του χρήστη.

Οι δύο τρόποι αναφοράς μπορούν να χρησιμοποιούνται ταυτόχρονα.

3.3 Ανάθεση αρχικών τιμών. Η εντολή POINT

Πρίν από κάθε διαδικασία ελαχιστοποίησης πρέπει να δοθούν αρχικές τιμές στις παραμέτρους της συνάρτησης. Ένας τρόπος είναι με τη χρήση της εντολής POINT (υπάρχουν και άλλοι τρόποι, βλέπε τις εντολές INIT, INSPECT και PICK). Η εντολή POINT αναμένει για είσοδο μία γραμμή της μορφής:

```
δείκτης τιμή δείκτης τιμή ...
```

Πχ:

```
1 1.4 2 -2.5
```

ή εάν έχουν δοθεί συμβολικά ονόματα στις παραμέτρους της συνάρτησης:

```
RHO 1.4 SIGMA -2.5
```

Και στα δύο ανωτέρω παραδείγματα η πρώτη παράμετρος της συνάρτησης λαμβάνει την αρχική τιμή 1.4 ενώ η δεύτερη -2.5, δηλαδή $x_1 = 1.4$ και $x_2 = -2.5$. Ο τρόπος αυτός εισαγωγής δεδομένων ονομάζεται "είσοδος μιάς γραμμής" και συναντάται σε όλες τις εντολές που έχουν σχέση με τις παραμέτρους της συνάρτησης. Εάν δεν επιθυμούμε να εισάγουμε τίποτε, αρκεί ένα απλό πάτημα του <cr>. Αμέσως μετά την "είσοδο μιάς γραμμής" το Merlin ζητά και πάλι τιμές για τις παραμέτρους. Ο πλήρης διάλογος φαίνεται στον πίνακα 8.

Όταν όλες οι παράμετροι της συνάρτησης έχουν πάρει αρχικές τιμές, το Merlin καλεί την αντικειμενική συνάρτηση funmin (ή το υποπρόγραμμα subsum) και υπολογίζει την τιμή της συνάρτησης για τις δεδομένες τιμές των παραμέτρων.

Πίνακας 8: Πλήρης διάλογος ανάθεσης αρχικών τιμών με την εντολή POINT.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
point  <cr>
POINT
1  1.4  2  -2.5  <cr>
POINT
<cr>

    /\/\/\/\/\/\     Merlin is at your command !!!
```

3.4 Οι εντολές SHORTDIS, VALDIS και TERMDIS

Αμέσως μετά την απόδοση αρχικών τιμών ο χρήστης μπορεί να ελέγξει τις τιμές των παραμέτρων και της συνάρτησης με την εντολή SHORTDIS. Τυπικό αποτέλεσμα από τη χρήση της εντολής φαίνεται στον πίνακα 9. Εκτός από τις τιμές των παραμέτρων και της συνάρτησης, εμφανίζεται ο αριθμός κλήσεων της συνάρτησης, ο αριθμός κλήσεων του υποπρογράματος των πρώτων παραγώγων granal (βλέπε παράγραφο 2.2) και ο αριθμός κλήσεων του υποπρογράμματος των δευτέρων παραγώγων hanal (βλέπε παράγραφο 2.4). Οι τιμές της συνάρτησης και των μετρητών μόνο, εμφανίζονται με τη χρήση της εντολής VALDIS. Στην περίπτωση που η συνάρτηση είναι άθροισμα τετραγώνων (ο χρήστης έχει προγραμματίσει την subroutine subsum), ή εντολή TERMDIS εμφανίζει τις τιμές των επιμέρους όρων f_i.

Πίνακας 9: Τυπικό αποτέλεσμα από τη χρήση της εντολής SHORTDIS.

```
    /\/\/\/\/\     Merlin is at your command !!!
shortdis  <cr>
SHORTDIS

------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                           2               0              0
Since last reset:                2               0              0
------------------------------------------------------------------------------
Index  Name       Fix       Parameter value       Left margin   Right margin
    1) rho        .......    1.40000000000000
    2) sigma      .......   -2.50000000000000
       Value .......      1989.32000000000

    /\/\/\/\/\     Merlin is at your command !!!
```

3.5 Η διαδικασία της ελαχιστοποίησης Πίνακες ελέγχου

Το Merlin διαθέτει ενσωματωμένους 8 αλγορίθμους ελαχιστοποίησης. Καθένας καλείται με τη χρήση αντίστοιχης εντολής, όπως φαίνεται στον πίνακα 10. Γιά κάθε αλγόριθμο, υπάρχει μιά σειρά παραμέτρων που καθορίζει τη λειτουργία του και μπορεί να αλλαχθούν από το χρήστη. Αυτοί ονομάζονται "παράμετροι ελέγχου" της μεθόδου και για την αλλαγή τους το Merlin εμφανίζει ένα "πίνακα ελέγχου". Στον πίνακα ελέγχου παρουσιάζονται αρχικά οι προκαθορισμένες τιμές των παραμέτρων ελέγχου της μεθόδου. Τυπικό παράδειγμα από τη χρήση της BFGS φαίνεται στον πίνακα 11. Ο πίνακας ελέγχου είναι διαφορετικός γιά κάθε μέθοδο και η σημασία των υπό-αλλαγή παραμέτρων ελέγχου για τους ελαχιστοποιητές εξηγείται στο κεφάλαιο 9. Εάν ο χρήστης επιθυμεί να αλλάξει τις τιμές ορισμένων από αυτές θα πρέπει να εισάγει τις αλλαγές του αμέσως μετά την προτροπή:

```
Enter changes:
```

Στο παράδειγμα του πίνακα 11, ο χρήστης άλλαξε την πρώτη παράμετρο (NUMBER OF CALLS: προσεγγιστικός μέγιστος επιτρεπτός αριθμός κλήσεων της αντικειμενικής συνάρτησης). Γενικά, αλλαγές στούς πίνακες ελέγχου γίνονται με την εισαγωγή μιάς γραμμής, με τη μορφή:

```
δείκτης_παραμέτρου  νέα_τιμή  δείκτης_παραμέτρου  νέα_τιμή  ...
```

Φυσικά απλό πάτημα του <cr> δηλώνει ότι ο χρήστης δεν επιθυμεί αλλαγές. Οι αλλαγές που γίνονται στους πίνακες ελέγχου διατηρούνται, και οι νέες τιμές που θέτει ο χρήστης εμφανίζονται στις διαδοχικές κλήσεις της ίδιας μεθόδου. Ο πίνακας 12 δείχνει τον πίνακα ελέγχου της BFGS, όπως θα παρουσιαζόταν κατά την δεύτερη κλήση του ελαχιστοποιητή.

Αμέσως μετά την εμφάνιση του πίνακα ελέγχου και τις (πιθανές) αλλαγές από το χρήστη, η επιλεγμένη μέθοδος ελαχιστοποίησης αρχίζει να εκτελείται. Ο χρήστης πληροφορείται την πορεία της ελαχιστοποίησης από μηνύματα της μορφής:

```
Iter: 1   Lower value:      389.532621415120        Calls: 9 of 300
```

όπου αναφέρεται η τιμή και οι κλήσεις της συνάρτησης που έχουν δαπανηθεί, καθώς και ο προσεγγιστικός μέγιστος αριθμός κλήσεων που ζητήθηκε από το χρήστη. Μετά το πέρας της ελαχιστοποίησης, αναφέρεται ο λόγος τερματισμού της μεθόδου (οι λόγοι τερματισμού περιγράφονται στην παράγραφο 10.2), με μια γραμμή της μορφής:

```
BFGS: The gradient criterion is satisfied
```

καθώς ο αριθμός κλήσεων της αντικειμενικής συνάρτησης, ο αριθμός κλήσεων του υποπρογράμματος των πρώτων παραγώγων granal, και ο αριθμός επαναλήψεων της μεθόδου:

```
Function evaluations:    71
Gradient evaluations:    51
Iterations:              47
```

Η τιμές των παραμέτρων της συνάρτησης μετά το πέρας της ελαχιστοποίησης μπορούν να ελεγθούν με την εντολή SHORTDIS.

Πίνακας 10: Οι αλγόριθμοι ελαχιστοποίησης του Merlin.

Εντολή	Χρήση πρώτων παραγώγων	Ειδικός γιά ελάχιστα τετράγωνα	Αναφορές
BFGS	√		[14, 6]
DFP	√		[15, 6]
CONGRA	√		[8, 16, 6]
SIMPLEX			[17, 18]
ROLL			[1]
TRUST			[7, p. 139]
TOLMIN	√		[19, 20]
LEVE		√	[21, 22, 6]

Πίνακας 11: Τυπικό παράδειγμα χρήσης της μεθόδου BFGS.

```
    /\/\/\/\/\     Merlin is at your command !!!
bfgs
BFGS
------------------------------------------------------------------------------
                                BFGS Panel
------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) NOC        Number of Calls                 300      Any int >= 1
 2) USEG       Use/Recalculate Gradient          0      {1,0}
 3) USEH       Use/Recalculate Hessian           0      {1,0}
 4) ITER       Iterations (-1 = Inf)            -1      Any int >= -1
 5) GMODE      Gradient Mode: Fixed/Mad          0      {0,1}
 6) XTOL       X-convergence Criterion       1.E-15     0. < real < 1.
 7) GTOL       G-convergence Criterion       1.E-15     0. < real < 1.
 8) FTOL       F-convergence Criterion          0.      0. <= real < 1.
 9) LSMETH     LS Method: Inexact/Exact          1      {1,2}
10) LSITER     Max. LS Iterations               30      Any int >= 1
11) RHO        Line Search Rho                .0001     0. <= real <= 1.
12) SIGMA      Line Search Sigma               .9       0. <= real <= 1.
13) PRINT      Printout Level                    1      {0,1,2}
14) CANCEL     Cancel/Proceed Minimiza...        1      {0,1}
------------------------------------------------------------------------------
Enter changes:
1 1000 <cr>
   Iter: 1   Lower value:     1.66909406463916        Calls: 7 of 300
------------------------------------------------------------------------------
   Iter: 2   Lower value:    1.171916564019462E-02    Calls: 9 of 300
------------------------------------------------------------------------------
   Iter: 3   Lower value:    1.742289904114637E-03    Calls: 10 of 300
------------------------------------------------------------------------------
   Iter: 4   Lower value:    1.303508001642794E-03    Calls: 11 of 300
------------------------------------------------------------------------------
   Iter: 5   Lower value:    6.582846410241483E-04    Calls: 13 of 300
------------------------------------------------------------------------------
   Iter: 6   Lower value:    2.043636357417198E-04    Calls: 14 of 300
------------------------------------------------------------------------------
   Iter: 7   Lower value:    9.853221569410267E-06    Calls: 15 of 300
------------------------------------------------------------------------------
   Iter: 8   Lower value:    3.737443067126273E-07    Calls: 16 of 300
------------------------------------------------------------------------------
   Iter: 9   Lower value:    2.044852936294917E-08    Calls: 17 of 300
------------------------------------------------------------------------------
   Iter: 10   Lower value:    2.183543054719823E-13    Calls: 18 of 300
------------------------------------------------------------------------------
   Iter: 11   Lower value:    3.906392742630301E-17    Calls: 19 of 300
------------------------------------------------------------------------------
   Iter: 12   Lower value:    1.057078161272313E-24    Calls: 20 of 300
------------------------------------------------------------------------------
   Iter: 13   Lower value:    1.244921116051909E-30    Calls: 21 of 300
------------------------------------------------------------------------------
                  BFGS: Further progress is not possible
------------------------------------------------------------------------------
Function evaluations:    22
Gradient evaluations:    15
Iterations:              14
------------------------------------------------------------------------------

    /\/\/\/\/\     Merlin is at your command !!!
```

Πίνακας 12: Ο πίνακας ελέγχου της BFGS, όπως θα παρουσιαζόταν κατά την δεύτερη κλήση της μεθόδου.

```
------------------------------------------------------------------------------
                                BFGS Panel
------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) NOC        Number of Calls                1000      Any int >= 1
 2) USEG       Use/Recalculate Gradient          0      {1,0}
 3) USEH       Use/Recalculate Hessian           0      {1,0}
 4) ITER       Iterations (-1 = Inf)            -1      Any int >= -1
 5) GMODE      Gradient Mode: Fixed/Mad          0      {0,1}
 6) XTOL       X-convergence Criterion       1.E-15     0. < real < 1.
 7) GTOL       G-convergence Criterion       1.E-15     0. < real < 1.
 8) FTOL       F-convergence Criterion          0.      0. <= real < 1.
 9) LSMETH     LS Method: Inexact/Exact          1      {1,2}
10) LSITER     Max. LS Iterations               30      Any int >= 1
11) RHO        Line Search Rho                .0001     0. <= real <= 1.
12) SIGMA      Line Search Sigma               .9       0. <= real <= 1.
13) PRINT      Printout Level                    1      {0,1,2}
14) CANCEL     Cancel/Proceed Minimiza...        1      {0,1}
------------------------------------------------------------------------------
```

3.6 Τερματισμός του προγράμματος

Υπάρχουν τρείς διαφορετικές εντολές που τερματίζουν τη λειτουργία του Merlin.

STOP Η λειτουργία του Merlin και του καλούντος προγράμματος σταματά αμέσως. (Γίνεται χρήση της εντολής Fortran STOP.)

RETURN Το Merlin επιστρέφει αμέσως τον έλεγχο στο καλών πρόγραμμα. (Γίνεται χρήση της εντολής Fortran RETURN.)

QUIT Το Merlin επιστρέφει τον έλεγχο στο καλών πρόγραμμα, όπως και με την εντολή RETURN. Προηγουμένως όμως ο χρήστης ερωτάται γιά την τιμή της "μεταβλητής επιστροφής" (ονομάζεται IQUIT, βλέπε παράγραφο 2.6). Το καλών πρόγραμμα μπορεί στη συνέχεια να ελέγξει την τιμή της και να καθορίσει ανάλογα τη δράση του.

Εάν χρησιμοποιηθούν οι εντολές STOP ή RETURN, η τιμή της επιστρεφόμενης μεταβλητής IQUIT είναι 0.

Κεφάλαιο 4

Παράγωγοι

4.1 Τρόποι υπολογισμού των παραγώγων

Στην τρέχουσα έκδοση του Merlin υπάρχουν πέντε διαφορετικοί τρόποι υπολογισμού των μερικών παραγώγων της συνάρτησης $g_i = \frac{\partial f}{\partial x_i}$.

FAST Κάθε μία από τις παραγώγους g_i προσεγγίζεται αριθμητικά από ένα τύπο εμπρόσθιων διαφορών δαπανώντας 2 κλήσεις της αντικειμενικής συνάρτησης.

QUAD Κάθε μία από τις παραγώγους g_i προσεγγίζεται αριθμητικά από ένα τύπο κεντρικών διαφορών δαπανώντας 4 κλήσεις της συνάρτησης.

NUMER Κάθε μία από τις παραγώγους g_i προσεγγίζεται αριθμητικά από ένα τύπο κεντρικών διαφορών δαπανώντας τουλάχιστον 6 κλήσεις της συνάρτησης.

ANAL Ολόκληρο το διάνυσμα **g** των πρώτων παραγώγων υπολογίζεται αναλυτικά από το υποπρόγραμμα granal που έχει προετοιμάσει ο χρήστης.

MIXED Κάθε μία από τις παραγώγους g_i υπολογίζεται με ένα διαφορετικό τρόπο. Αυτό είναι χρήσιμο στην περίπτωση που δεν υπάρχουν αναλυτικοί τύποι για όλες τις παραγώγους g_i. Η εντολή MIXED αναμένει είσοδο μιάς γραμμής στη μορφή:

δείκτης τρόπος_υπολογισμού δείκτης τρόπος_υπολογισμού ...

Πχ.

```
1 anal 2 numer
```

ή χάριν συντομίας

```
1 a 2 n
```

όπου καθορίζεται ότι η g_1 θα υπολογιστεί αναλυτικά από το υποπρόγραμμα granal του χρήστη, ενώ η g_2 θα προσεγγιστεί με τον τρόπο NUMER.

4.2 Έλεγχος των παραγώγων Εντολές GRADDIS και GRADCHECK

Οι τιμές των παραγώγων g_i για τις τρέχουσες τιμές των παραμέτρων x_i μπορούν να ελεγθούν με την εντολή GRADDIS. Τυπικό παράδειγμα χρήσης της GRADDIS φαίνεται στον πίνακα 13, όπου η g_1 υπολογίζεται με τον τρόπο NUMER και η g_2 με τον τρόπο ANAL.

Σύγκριση του τρόπου υπολογισμού των παραγώγων και κυρίως επιβεβαίωση ότι οι αναλυτικές παράγωγοι έχουν προγραμματιστεί σωστά, μπορεί να γίνει με την εντολή GRADCHECK. Τυπικό παράδειγμα χρήσης της GRADCHECK φαίνεται στον πίνακα 14.

Το Merlin περιέχει ενσωματωμένο μηχανισμό που αποτρέπει άσκοπους επαναληπτικούς υπολογισμούς των παραγώγων όταν αυτές απαιτούνται στο ίδιο σημείο. Πχ. Από τρείς διαδοχικές εντολές GRADDIS, μόνο η πρώτη θα υπολόγιζε στην πραγματικότητα τις παραγώγους.

GRADDIS Εμφανίζει τις τιμές των παραγώγων για τον τρέχοντα τρόπο υπολογισμού.

GRADCHECK Ελέγχει τον υπολογισμό των παραγώγων με τον τρέχοντα τρόπο, συγκριτικά με κάποιο άλλο τρόπο υπολογισμού (που ζητείται ως είσοδος από το χρήστη).

Πίνακας 13: Παράδειγμα χρήσης της εντολής GRADDIS.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
graddis  <cr>
GRADDIS

Index  Name      Fix       Gradient                Gradient-Mode
-------------------------------------------------------------
    1) Rho       .......  1.920600000000000E+04        ANAL
    2) Sigma     ....... -2.400000000000000E+03        NUMER

    /\/\/\/\/\/\     Merlin is at your command !!!
```

Πίνακας 14: Δείγμα από τη χρήση της GRADCHECK.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
gradcheck  <cr>
GRADCHECK

Enter derivative mode (fast, quad, numer, anal):
anal

-------------------------------------------------------------------------------
Index  Name       Fix          ANAL           NUMER             Rel. Error
-------------------------------------------------------------------------------
    1) Rho        .......   1.92060E+04    1.92060E+04           1.267E-13
    2) Sigma      .......  -2.40000E+03   -2.40000E+03           0.000E+00

RMS Error:  8.9605E-14

    /\/\/\/\/\/\     Merlin is at your command !!!
```

4.3 Αυτόματη αλλαγή τρόπου παραγώγισης Η εντολή MAD

Κατά τη χρήση των ελαχιστοποιητών BFGS, DFP, CONGRA και TOLMIN που απαιτούν τον υπολογισμό των πρώτων παραγώγων της συνάρτησης, έχει παρατηρηθεί ότι όταν ο ελαχιστοποιητής βρίσκεται πολύ μακριά από το ελάχιστο, χονδρικές προσεγγίσεις των παραγώγων είναι αρκετές. Καθώς η μέθοδος πλησιάζει στο ελάχιστο, απαιτούνται προσεγγίσεις των παραγώγων υψηλότερης ακρίβειας. Το Merlin περιέχει ενσωματωμένο μηχανισμό που του επιτρέπει να αλλάζει αυτόματα τον τρόπο υπολογισμού των παραγώγων κάτω από ορισμένα κριτήρια.

Συγκεκριμένα ο τρόπος υπολογισμού της παραγώγου g_i αλλάζει σε ακριβέστερο (από FAST σε QUAD ή από QUAD σε NUMER), όταν η απόλυτη τιμή της παραγώγου g_i είναι μικρότερη από ένα προκαθορισμένο όριο (FQTHRE, QNTHRE). Ο τρόπος υπολογισμού της παραγώγου g_i αλλάζει σε έναν λιγότερο ακριβή (από QUAD σε FAST ή από NUMER σε QUAD), όταν η απόλυτη τιμή της παραγώγου g_i είναι μεγαλύτερη από ένα προκαθορισμένο όριο (FQTHRE, QNTHRE).

Οι ανωτέρω παράμετροι ελέγχονται από την εντολή MAD. Ο πίνακας ελέγχου της εντολής MAD περιέχει τις παραμέτρους:

`FAST <==> QUAD THRESHOLD` Το όριο (FQTHRE) στην τιμή της παραγώγου για αλλαγή από FAST σε QUAD.

`QUAD <==> NUMER THRESHOLD` Το όριο (QNTHRE) στην τιμή της παραγώγου για αλλαγή από QUAD σε NUMER.

`MAD PRINTOUT (OFF/ON)` Καθορίζει αν θα υπάρχει έξοδος που θα πληροφορεί το χρήστη για την αλλαγή του τρόπου παραγώγισης. Επιτρεπτές τιμές είναι:

- 0 Δεν παράγεται έξοδος.
- 1 Παράγεται έξοδος που πληροφορεί το χρήστη για την αλλαγή του τρόπου παραγώγισης και για το λόγο της αλλαγής.

Για να ενεργοποιηθεί ο αυτόματος τρόπος αλλαγής παραγώγων από τις μεθόδους ελαχιστοποίησης, πρέπει στους αντίστοιχους πίνακες ελέγχου να δοθεί η τιμή 1 στην παράμετρο GRADIENT MODE (βλέπε παράγραφο 9.3). Μετά το πέρας κάθε ελαχιστοποίησης που έχει γίνει με την αυτόματη αλλαγή παραγώγων, ο τρόπος υπολογισμού των παραγώγων τίθεται σε MIXED. Έχει βρεθεί ότι η αυτόματη αλλαγή παραγώγων επιταχύνει τους ελαχιστοποιητές, ειδικά στις περιπτώσεις συναρτήσεων με μεγάλο αριθμό μεταβλητών.

Κεφάλαιο 5

Εντολές ελέγχου

5.1 Έλεγχος του τρόπου λειτουργίας του Merlin Εντολές MODEDIS και LIMITS

Οι παράμετροι που επηρρεάζουν τη λειτουργία του Merlin μπορούν να ελεγθούν με τη χρήση της εντολής MODEDIS. Η εντολή LIMITS παρουσιάζει τις μέγιστες τιμές εσωτερικών παραμέτρων του προγράμματος και είναι χρήσιμη, όταν πραγματοποιούνται επεκτάσεις (βλέπε κεφάλαιο 12). Τυπικό αποτέλεσμα από τη χρήση των εντολών φαίνεται στον πίνακα 15.

Πίνακας 15: Αποτέλεσμα από χρήση των εντολών MODEDIS και LIMITS.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
modedis  <cr>
MODEDIS

Current operation modes
------------------------------------------------------------------------------
Number of variables:       2                Largest real:           1.+300
Number of terms:           2                Smallest nonzero real:  1.-300
Backup mode:              NOBACK            Machine's accuracy:     1.E-15
Derivative mode:          ANAL              Target value:           - NONE -
Printout mode:            FULLPRINT         User Plug-in modules:       1
Processing mode:          IAF
Prolog command:            - NONE -
Epilog command:            - NONE -
Function evaluation:      ON
Functional form           General function
------------------------------------------------------------------------------

    /\/\/\/\/\/\     Merlin is at your command !!!
limits  <cr>
LIMITS

Current limits are:
-----------------------------------------------------
Maximum number of minimization parameters:          200
Maximum number of terms in the sum of squares:      200
Maximum number of aliases:                           20
Maximum nesting level:                               20
Number of numerical flags:                           20
Number of character flags:                           10
Length of a character flag:                          30
Maximum number of plug-in modules:                   50

Current MCL memory:                                10000
Maximum number of MCL files:                          20
Maximum length of an MCL object code,
that fits in memory:          20000 characters 4000 lines

    /\/\/\/\/\/\     Merlin is at your command !!!
```

5.2 Μορφή της αντικειμενικής συνάρτησης. Εντολές GENERAL και SOS

Το Merlin αναγνωρίζει δύο μορφές αντικειμενικών συναρτήσεων, που επιλέγονται με αντίστοιχες εντολές:

GENERAL Καθορίζει ότι η αντικειμενική συνάρτηση είναι μία γενική μη γραμμική συνάρτηση. Για τον υπολογισμό της ο χρήστης πρέπει να έχει προετοιμάσει τον πηγαίο κώδικα Fortran για την `FUNTION FUNMIN` (βλέπε παράγραφο 2.1). Αυτή είναι η προκαθορισμένη μορφή συνάρτησης.

SOS Καθορίζει ότι η αντικειμενική συνάρτηση είναι άθροισμα τετραγώνων όπως περιγράφεται στην παράγραφο 3.3. Για τον υπολογισμό της ο χρήστης πρέπει να έχει προετοιμάσει τον πηγαίο κώδικα Fortran για την `SUBROUTINE SUBSUM` (βλέπε παράγραφο 2.3).

Και στις δύο ανωτέρω περιπτώσεις, εάν η παράγωγοι της συνάρτησης ειναι αναλυτικά γνωστές θα πρέπει να προγραμματιστούν ως `SUBROUTINE GRANAL` (βλέπε παράγραφο 2.2). Μόνο μία από τις `FUNCTION FUNMIN` ή `SUBROUTINE SUBSUM` πρέπει να προγραμματιστεί.

5.3 Τρόπος λειτουργίας. Εντολές IAF και BATCH

Το Merlin είναι ένα διαλογικό πρόγραμμα που όμως μπορεί να χρησιμοποιηθεί σε μη διαλογικό τρόπο λειτουργίας batch χωρίς την παρέμβαση του χρήστη. Αυτό μπορεί να γίνει είτε με την χρήση των οδηγιών διαμόρφωσης `INPUT_FILE` και `OUTPUT_FILE` στο αρχείο `CONFIG` (βλέπε παραγράφους 12.6, 12.7), ή με την ανακατεύθυνση των μονάδων εισόδου/εξόδου όταν το λειτουργικό σύστημα το επιτρέπει. Πχ. για το λειτουργικό σύστημα unix:

```
merlin < input_file > output_file
```

Στην περίπτωση που κατά τη διάρκεια της εκτέλεσης συμβεί κάποιο λάθος (πχ. υπάρχει μιά λανθασμένη εντολή στο αρχείο εισόδου), οι ενέργειες του Merlin καθορίζονται με τις εντολές:

IAF Στην περίπτωση λαθών το Merlin απλώς τυπώνει το προτρεπτικό σήμα και αναμένει την επόμενη εντολή. Στην περίπτωση εισόδου από αρχείο, διαβάζεται από το αρχείο η επόμενη εντολή. Αυτός είναι ο προκαθορισμένος τρόπος λειτουργίας.

BATCH Στην περίπτωση λαθών, το Merlin τερματίζει τη λειτουργία του τυπώνοντας κατάλληλα διαγνωστικά μηνύματα.

Ο καθορισμός του διαλογικού ή μη τρόπου λειτουργίας γίνεται είτε με τις ανωτέρω εντολές ή με την οδηγία διαμόρφωσης `MODE` στο αρχείο `CONFIG` (βλέπε παράγραφο 11.14).

5.4 Έλεγχος εξόδου
Εντολές FULLPRINT, HALFPRINT και NOPRINT

Το ποσό της εξόδου που παράγεται από το Merlin, ελέγχεται με τις εντολές:

`FULLPRINT` Όλα τα διαγνωστικά μηνύματα και οι προτροπές του Merlin εμφανίζονται. Αυτή είναι η προκαθορισμένη συμπεριφορά.
`HALFPRINT` Εμφανίζονται μόνο διαγνωστικά μηνύματα σε περίπτωση σφαλμάτων.
`NOPRINT` Δεν παράγεται καθόλου έξοδος.

Η προκαθορισμένη συμπεριφορά μπορεί να αλλάξει είτε με κάποια από τις ανωτέρω εντολές, ή με την οδηγία διαμόρφωσης `PRINTOUT` στο αρχείο `CONFIG` (βλέπε παράγραφο 11.11).

5.5 Μετρητές κλήσεων. Εντολή RESET

Το Merlin περιέχει ενσωματωμένους τέσερεις μετρητές οι τιμές των οποίων εμφανίζονται με χρήση των εντολών `SHORTDIS` και `VALDIS`:

α) Ο αριθμός τον κλήσεων της αντικειμενικής συνάρτησης από την αρχή της λειτουργίας του Merlin.
β) Ο αριθμός τον κλήσεων της αντικειμενικής συνάρτησης από την πιό πρόσφατη εντολή `RESET`.
γ) Ο αριθμός τον κλήσεων του υποπρογράμματος granal που υπολογίζει αναλυτικά τις παραγώγους, από την αρχή της λειτουργίας του Merlin.
δ) Ο αριθμός τον κλήσεων του υποπρογράμματος granal που υπολογίζει αναλυτικά τις παραγώγους, από την πιό πρόσφατη εντολή `RESET`.
ε) Ο αριθμός τον κλήσεων του υποπρογράμματος hanal που υπολογίζει αναλυτικά τις δεύτερες παραγώγους, από την αρχή της λειτουργίας του Merlin.
στ) Ο αριθμός τον κλήσεων του υποπρογράμματος hanal που υπολογίζει αναλυτικά τις δεύτερες παραγώγους, από την πιό πρόσφατη εντολή `RESET`.

Η εντολή `RESET` μηδενίζει τους μετρητές (β), (δ) και (στ) επιτρέποντας την καταμέτρηση επιμέρους κλήσεων από διάφορες μεθόδους ελαχιστοποίησης.

5.6 Παρουσίαση των πινάκων ελέγχου
Εντολές PANELON και PANELOFF

Η εμφάνιση των πινάκων ελέγχου για τις εντολές ελαχιστοποίησης (αλλά και άλλες εντολές που χρησιμοποιούν πίνακες ελέγχου) καθορίζεται από τις εντολές:

PANELON Οι πίνακες ελέγχου εμφανίζονται και ο χρήστης μπορεί να εισάγει αλλαγές στις παραμέτρους ελέγχου. Αυτή είναι η προκαθορισμένη συμπεριφορά.

PANELOFF Οι πίνακες ελέγχου δεν εμφανίζονται, αλλά η μέθοδος προχωρεί με τις ήδη υπάρχουσες τιμές των παραμέτρων ελέγχου.

Σημειώστε ότι αν έχει δοθεί η εντολή NOPRINT, δεν παράγεται έξοδος, αλλά ο πίνακας ελέγχου εξακολουθεί να αναμένει την είσοδο αλλαγών, όπως περιγράφεται στην παράγραφο 3.5.

5.7 Μετονομασίες εντολών Εντολές ALIAS, UNALIAS και ALIASDIS

Είναι δυνατόν να καθοριστούν δεύτερα ονόματα γιά οποιαδήποτε από τις εντολές του Merlin. Τα δεύτερα ονόματα έχουν μήκος μέχρι 5 χαρακτήρες και ελέγχονται από τις εντολές:

ALIAS Καθορίζει το δεύτερο όνομα μιας εντολής.

UNALIAS Ακυρώνει το δεύτερο όνομα μιας εντολής.

ALIASDIS Εμφανίζει τα ήδη καθορισμένα δεύτερα ονόματα.

Τυπικό παράδειγμα των ανωτέρω εντολών όπου το GD ορίζεται σαν δεύτερο όνομα της εντολής GRADDIS φαίνεται στον πίνακα 16.

Πίνακας 16: Χρήση των εντολών ALIAS, ALIASDIS και UNALIAS.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
alias  <cr>
ALIAS

Enter alias:
gd  <cr>

Enter actual command:
graddis  <cr>

Alias "gd" has been defined as the command "graddis".

    /\/\/\/\/\/\     Merlin is at your command !!!
aliasdis  <cr>
ALIASDIS

Current aliases are:
--------------------
gd    = graddis

    /\/\/\/\/\/\     Merlin is at your command !!!
unalias  <cr>
UNALIAS

Enter alias to remove:
gd

Alias "gd" has been removed.

    /\/\/\/\/\/\     Merlin is at your command !!!
```

5.8 Απαγόρευση υπολογισμού της συνάρτησης Εντολές NOEVAL και EVALUATE

Κάθε φορά που γίνεται χρήση των εντολών POINT, MARGIN κα. το Merlin καλεί την αντικειμενική συνάρτηση. Πολλές φορές είναι επιθυμητό (ιδιαίτερα μέσω προγραμμάτων MCL, βλέπε κεφάλαιο 10) να αλλάξουμε τις τιμές πολλών παραμέτρων, ορίων κλπ., υπολογίζοντας τη συνάρτηση μία μόνο φορά στο τέλος, αφού ολοκληρώσουμε τις αλλαγές. Αυτό επιτυγχάνεται με τις εντολές:

NOEVAL Η αντικειμενική συνάρτηση δεν υπολογίζεται (δεν καλείται η FUNCTION FUNMIN ή η SUBROUTINE SUBSUM). Όλες οι εντολές του Merlin που απαιτούν υπολογισμό της αντικειμενικής συνάρτησης (πχ. εντολές ελαχιστοποίησης) δεν μπορούν να εκτελεστούν.

EVALUATE Η αντικειμενική συνάρτηση υπολογίζεται όποτε χρειαστεί. Αυτός είναι ο προκαθορισμένος τρόπος λειτουργίας. Επιπλέον, κάθε φορά που δίνεται η εντολή EVALUATE το Merlin εξαναγκάζεται να κάνει μία κλήση της αντικειμενικής συνάρτησης.

5.9 Ανάθεση συμβολικών ονομάτων Η εντολές GODFATHER και NONAME

Η ανάθεση συμβολικών ονομάτων στις παραμέτρους της αντικειμενικής συνάρτησης γίνεται με χρήση της εντολής GODFATHER. Η εντολή GODFATHER αναμένει είσοδο μιάς γραμμής της μορφής:

```
δείκτης όνομα δείκτης όνομα ...
```

Πχ.

```
1 Rho 2 Sigma
```

Τα συμβολικά ονόματα έχουν μήκος μέχρι 10 χαρακτήρες και η επιλογή τους αφήνεται στην απόλυτη διάθεση του χρήστη. Δεν είναι απαραίτητο να ανατεθούν συμβολικά ονόματα σε όλες τις παραμέτρους. Η εντολή NONAME αφαιρεί τα συμβολικά ανόματα. Αναμένει είσοδο μιας γραμμής με τους δείκτες των παραμέτρων των οποίων τα ονόματα πρόκειται να αφαιρεθούν.

5.10 Ανακατεύθυνση της εξόδου Εντολές HIDEOUT και REVEAL

Προσωρινή ανακατεύθυνση της εξόδου του Merlin μπορεί να γίνει με χρήση των εντολών:

`HIDEOUT` Ζητείται το όνομα ενός αρχείου που θα δεχτεί προσωρινά την έξοδο του Merlin. Από τη χρήση της εντολής `HIDEOUT`, μέχρι τη χρήση της εντολής `REVEAL`, όλη η έξοδος του Merlin γράφεται στο συγκεκριμένο αρχείο.

`REVEAL` Επαναφέρει την έξοδο του Merlin στον προκαθορισμένο προορισμό αναιρώντας τη λειτουργία της προηγούμενης εντολής `HIDEOUT`.

Σημειώστε ότι επιτρέπεται η χρήση κιβωτισμένων εντολών `HIDEOUT – REVEAL`.

5.11 Έλεγχος βημάτων Εντολές STEP, STEPDIS, STEPALL και ADJUST

Ο ελαχιστοποιητής `ROLL` γιά κάθε παράμετρο x_i χρησιμοποιεί (και τροποποιεί) τα βήματα s_i (βλέπε παράγραφο 9.7). Οι παρακάτω εντολές σχετίζονται με τον έλεγχο των βημάτων s_i.

`STEP` Ζητείται είσοδος μιάς γραμμής με τις τιμές των βημάτων. Εάν αυτές δεν καθοριστούν με χρήση της εντολής `STEP`, η προκαθορισμένη τιμή τους είναι 1.

`STEPDIS` Εμφανίζονται οι τρέχουσες τιμές των βημάτων s_i.

`STEPALL` Καθορίζονται τα βήματα γιά όλες τις παραμέτρους x_i, με μιά γραμμική αναζήτηση στη διεύθυνση που δείχνουν οι παράγωγοι της συνάρτησης.

`ADJUST` Καθορίζονται τα βήματα γιά όλες τις παραμέτρους x_i, χωρίς να υπολογιστούν οι παράγωγοι της συνάρτησης.

Τυπικός διάλογος χρήσης των εντολών `STEP` και `STEPDIS` φαίνεται στον πίνακα 17.

Πίνακας 17: Χρήση των εντολών `STEP` και `STEPDIS`.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
step  <cr>
STEP
1 1.5 2 1.1

    /\/\/\/\/\/\     Merlin is at your command !!!
stepdis  <cr>
STEPDIS

Index  Name      Fix       Step
-------------------------------------
    1) Rho       .......   1.5000
    2) Sigma     .......   1.1000

    /\/\/\/\/\/\     Merlin is at your command !!!
```

Κεφάλαιο 6

Άλλες δυνατότητες

6.1 Όρια στις παραμέτρους. Εντολές LMARGIN, RMARGIN, LDEMARGIN και RDEMARGIN

Όρια του τύπου:

$$a_i \leq x_i \leq b_i \qquad i = 1, 2, \ldots N$$

μπορεί να αντιμετωπιστούν από τους ελαχιστοποιητές του Merlin. Κάθε μία από τις παραμέτρους x_i μπορεί να έχει αριστερό όριο a_i , δεξί όριο b_i , ή και τα δύο. Τα όρια τίθενται με τις εντολές RMARGIN και LMARGIN από τις οποίες ζητείται είσοδος μιάς γραμμής με τους δείκτες των παραμέτρων και τις αντίστοιχες τιμές των ορίων. Τυπικό παράδειγμα των εντολέων RMARGIN και LMARGIN φαίνεται στον πίνακα 18. Οι τιμές των ορίων εμφανίζονται με χρήση της εντολής SHORTDIS.

Ακύρωση των ορίων γίνεται με τις εντολές LDEMARGIN και RDEMARGIN από τις οποίες ζητείται είσοδος μιας γραμμής για τις παραμέτρους των οποίων επιθυμούμε να ακυρώσουμε τα όρια. Τυπικό παράδειγμα φαίνεται στον πίνακα 19.

Πίνακας 18: Παράδειγμα χρήσης των εντολών LMARGIN και RMARGIN καθώς και εξέταση των ορίων αμέσως μετά με την εντολή SHORTDIS.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
lmargin  <cr>
LMARGIN

Enter left margins sequence:
1  -5  2  -7  <cr>

    /\/\/\/\/\/\     Merlin is at your command !!!
rmargin  <cr>
RMARGIN

Enter right margins sequence:
1  8  <cr>

    /\/\/\/\/\/\     Merlin is at your command !!!
shortdis  <cr>
SHORTDIS

------------------------------------------------------------------------------
Number of evaluations:     Function        Gradient        Hessian
Total:                          132               1              0
Since last reset:               132               1              0
------------------------------------------------------------------------------
Index  Name        Fix        Parameter value       Left margin   Right margin
    1) Rho         .......    1.40000000000000        -5.000        8.000
    2) Sigma       .......   -2.50000000000000        -7.000
       Value .......      1989.32000000000

    /\/\/\/\/\/\     Merlin is at your command !!!
```

Πίνακας 19: Παράδειγμα χρήσης των εντολών LDEMARGIN και RDEMARGIN καθώς και εξέταση των ορίων αμέσως μετά με την εντολή SHORTDIS.

```
/\/\/\/\/\/\     Merlin is at your command !!!
ldemargin  <cr>
LDEMARGIN

Enter left margins sequence:
2  <cr>

    /\/\/\/\/\/\     Merlin is at your command !!!
rdemargin  <cr>
RDEMARGIN

Enter right margins sequence:
1  <cr>

    /\/\/\/\/\/\     Merlin is at your command !!!
shortdis  <cr>
SHORTDIS

-------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                         132               1              0
Since last reset:              132               1              0
-------------------------------------------------------------------------------
Index  Name       Fix       Parameter value       Left margin   Right margin
    1) Rho        .......    1.40000000000000        -5.000
    2) Sigma      .......   -2.50000000000000
       Value .......      1989.32000000000

    /\/\/\/\/\/\     Merlin is at your command !!!
```

6.2 Σταθεροποίηση των παραμέτρων Εντολές FIX, LOOSE, FIXALL και LOOSALL

Πολλές φορές είναι επιθυμητό να σταθεροποιήσουμε την τιμή ορισμένων παραμέτρων σε κάποια συγκεκριμένη τιμή και να προχωρήσουμε σε ελαχιστοποίηση χρησιμοποιώντας τις υπόλοιπες. Η σταθεροποίηση/αποσταθεροποίηση των παραμέτρων ελέγχεται από τις εντολές:

FIX Ζητείται είσοδος μιάς γραμμής για να καθοριστούν οι παράμετροι προς σταθεροποίηση. Οι παράμετροι που προσδιορίζουμε σταθεροποιούνται στην τρέχουσα τιμή τους ενώ με τη χρήση της SHORTDIS εμφανίζεται η ένδειξη . . * . . αριστερά της τιμής τους.

LOOSE Ακυρώνει τη λειτουργία της εντολής FIX. Οι παράμετροι γιά τις οποίες ακυρώνεται η λειτουργία της σταθεροποίησης δίνονται ως είσοδος μιάς γραμμής.

LOOSALL Ακυρώνει τη σταθεροποίηση όλων των παραμέτρων.

FIXALL Σταθεροποιεί όλες τις παραμέτρους στην τρέχουσα τιμή τους. Προφανώς πρέπει να προηγηθεί εντολή LOOSE πριν από οποιαδήποτε ελαχιστοποίηση, ειδάλως κατάλληλα διαγνωστικά μηνύματα θα δείξουν ότι όλες οι παράμετροι είναι καθηλωμένες.

Τυπικά παραδείγματα χρήσης των εντολών FIX και LOOSE φαίνονται στους πίνακες 20 και 21.

Πίνακας 20: Παράδειγμα χρήσης της εντολής FIX καθώς και εξέταση των παραμέτρων αμέσως μετά με την εντολή SHORTDIS.

```
    /\/\/\/\/\     Merlin is at your command !!!
fix  <cr>
FIX
1  <cr>

    /\/\/\/\/\     Merlin is at your command !!!
shortdis  <cr>
SHORTDIS

-------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                         132               1              0
Since last reset:              132               1              0
-------------------------------------------------------------------------------
Index  Name       Fix       Parameter value       Left margin   Right margin
    1) Rho        .. * ..    1.40000000000000
    2) Sigma      .......   -2.50000000000000
       Value .......      1989.32000000000

    /\/\/\/\/\     Merlin is at your command !!!
```

Πίνακας 21: Παράδειγμα χρήσης της εντολής LOOSE καθώς και εξέταση των παραμέτρων αμέσως μετά με την εντολή SHORTDIS.

```
    /\/\/\/\/\     Merlin is at your command !!!
loose  <cr>
LOOSE
1  <cr>

    /\/\/\/\/\     Merlin is at your command !!!
shortdis  <cr>
SHORTDIS

-------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                         132               1              0
Since last reset:              132               1              0
-------------------------------------------------------------------------------
Index  Name       Fix       Parameter value       Left margin   Right margin
    1) Rho        .......    1.40000000000000
    2) Sigma      .......   -2.50000000000000
       Value .......      1989.32000000000

    /\/\/\/\/\     Merlin is at your command !!!
```

6.3 Συσσώρευση αρχικών σημείων. Η εντολή ACCUM

Τυχαία αρχικά σημεία εκκίνησης για τους ελαχιστοποιητές μπορούν να παραχθούν με την εντολή ACCUM. Για το σκοπό αυτό κάθε παράμετρος x_i θα πρέπει είτε να έχει άνω και κάτω όριο (μετά από χρήση των εντολών LMARGIN και RMARGIN), ή η τιμή της να έχει σταθεροποιηθεί (με την εντολή FIX). Η ACCUM παράγει τυχαία δοκιμαστικά σημεία, και συσσωρεύει σε ένα καθορισμένο αρχείο (βλέπε παράγραφο 6.3) όσα δίνουν τιμή συνάρτησης μικρότερη από μιά τιμή στόχο. Η ACCUM χρησιμοποιεί ένα πίνακα ελέγχου για την εισαγωγή των απαραίτητων στοιχείων.

Οι παράμετροι του πίνακα ελέγχου της ACCUM είναι:

NUMBER OF CALLS	Μέγιστος επιτρεπτός αριθμός κλήσεων της αντικειμενικής συνάρτησης.
NUMBER OF POINTS	Ζητούμενος αριθμός σημείων με τιμή μικρότερη της τιμής-στόχου. Κάθε τέτοιο σημείο προστίθεται στο αρχείο που καθορίζεται από την παράμετρο FILE.
TARGET VALUE	Η τιμή-στόχος με την οποία συγκρίνονται τα δοκιμαστικά σημεία που δειγματίζει η ACCUM.
FILE	Το αρχείο στο οποίο θα γραφούν τα σημεία με τιμή μικρότερη από την τιμή-στόχο.
CANCEL-BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Η λειτουργία της ACCUM τερματίζεται όταν είτε ο μέγιστος επιτρεπτός αριθμός κλήσεων εξαντληθεί, ή βρεθούν τόσα σημεία όσα ζητούνται με την παράμετρο NUMBER OF POINTS. Μετά το τέλος της ACCUM, και αφού έχουν συσσωρευθεί σημεία στο αρχείο, ο χρήστης μπορεί να επιθεωρήσει το αρχείο με την εντολή INSPECT και να επιλέξει οποιοδήποτε από τα σημεία με την εντολή PICK (βλέπε παράγραφο 7.3). Τυπικό παράδειγμα φαίνεται στον πίνακα 22.

Πίνακας 22: Δείγμα χρήσης της εντολής ACCUM.

```
    /\/\/\/\/\/     Merlin is at your command !!!
accum  <cr>
ACCUM
-------------------------------------------------------------------------------
                                  ACCUM Panel
-------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) NOC        Number of calls                  300     Any int >= 1
 2) NOP        No. of Points                      1     Any int >= 1
 3) TARGET     Target Value                  3375.6     Any real
 4) FILE       File name                      INIPO     Any string
 5) CANCEL     Cancel button                      1     {0,1}
-------------------------------------------------------------------------------
Enter changes:
3 3000 2 5 <cr>

Number of points appended to file "INIPO": 5
Function calls: 30

    /\/\/\/\/\/     Merlin is at your command !!!
```

6.4 Γραφική απεικόνιση. Η εντολή GRAPH

Γραφικές απεικονίσεις της συνάρτησης με τη χρήση χαρακτήρων μπορούν να γίνουν με την εντολή GRAPH. Η εντολή GRAPH χρησιμοποιεί ένα πίνακα ελέγχου για την εισαγωγή των απαραίτητων στοιχείων.

Οι παράμετροι του πίνακα ελέγχου της GRAPH είναι:

VARIABLE INDEX	Ο δείκτης της παραμέτρου της αντικειμενικής συνάρτησης ως προς την οποία ζητείται γραφική απεικόνιση. Όλες οι υπόλοιπες παράμετροι θεωρούνται σταθερές.
NO OF POINTS	Ο αριθμός των σημείων (και κατά συνέπεια κλήσεων της συνάρτησης) που θα χρησιμοποιηθούν.
FROM	Μαζί με την παράμετρο TO καθορίζουν το διάστημα στο οποίο ζητείται η απεικόνιση.
TO	Μαζί με την παράμετρο FROM καθορίζουν το διάστημα στο οποίο ζητείται η απεικόνιση.
LINES	Καθορίζει τον αριθμό των γραμμών που θα καταλάβει η απεικόνιση στην οθόνη.
COLOUMNS	Καθορίζει τον αριθμό των στηλών που θα καταλάβει η απεικόνιση στην οθόνη.
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Τυπικό παράδειγμα χρήσης της GRAPH φαίνεται στον πίνακα 23.

Πίνακας 23: Τυπικό παράδειγμα χρήσης της GRAPH.

```
/\/\/\/\/\/\     Merlin is at your command !!!
graph  <cr>
GRAPH
------------------------------------------------------------------------------
                                GRAPH Panel
------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) INDEX      Variable Index                   1       Any int >= 1
 2) NOP        No. of Points                   30       Any int >= 1
 3) FROM       From                         -10000.     Any real
 4) TO         To                           10000.      Any real
 5) LINES      Lines                           17       Any int >= 1
 6) COLS       Coloumns                        64       Any int >= 1
 7) CANCEL     Cancel button                    1       {0,1}
------------------------------------------------------------------------------
Enter changes:
1 2 3 -1.5 4 5 <cr>


             .+....+....+....+....+....+....+....+....+....+....+....+....+....+....
    1.600E+03+               .                                                  *+
             .               .                                                   .
 F  1.400E+03+               .                                                *  +
 U           .               .                                              *    .
 N  1.200E+03+               .                                                   +
 C           .               .                                          *        .
 T  1.000E+03+               .                                        *          +
 I           .               .                                                   .
 O   800.    +               .                                   *               +
 N           .               .                                *                  .
     600.    +*              .                             *                     +
             .  *            .                          *                        .
 V   400.    +    *          .                       *                           +
 A           .       * *     .                    *                              .
 L   200.    +           *   .                 * *                               +
 U           .             * *               ***                                 .
 E  0.119    +               . *  * * X * * *                                     +
             .+....+....+....+....+....+....+....+....+....+....+....+....+....+....
       -1.500000E+00   I    5.634921E-01   I    2.626984E+00   I
                -4.682540E-01        1.595238E+00        3.658730E+00
                VARIABLE :   2)


    //////     MERLIN   IS AT YOUR COMMAND !!!
```

6.5 Μαζική ανάθεση αρχικών τιμών. Η εντολή INIT

Η εντολή INIT επιτρέπει στο χρήστη να δώσει αρχικές τιμές στις παραμέτρους της συνάρτησης, διαβάζοντάς τις από κάποιο αρχείο. Επιπλέον μπορούν να αποδωθούν αρχικές τιμές στα αριστερά ή δεξιά όρια, στα βήματα της εντολής ROLL (βλέπε παράγραφο 5.11), στα συμβολικά ονόματα των παραμέτρων και στην κατάσταση σταθεροποίησης κάθε παραμέτρου (βλέπε παράγραφο 6.2). Σε αυτή την τελευταία περίπτωση, στο αρχείο πρέπει να υπάρχει ο αριθμός 0 ή 1 για μιά σταθεροποιημένη ή ελεύθερη μεταβλητή αντίστοιχα.

Οι παράμετροι του πίνακα ελέγχου της INIT είναι:

FILE	Το αρχείο από το οποίο θα διαβαστούν οι αρχικές τιμές.
WHAT	Καθορίζει που θα αποδοθούν αρχικές τιμές:
	X Τιμές των παραμέτρων.
	L Αριστερά (κάτω) όρια των παραμέτρων.
	R Δεξιά (άνω) όρια των παραμέτρων.
	F Κατάσταση σταθεροποίησης των παραμέτρων.
	S Βήματα της εντολής ROLL.
	N Συμβολικά ονόματα των παραμέτρων.
NPL	Αριθμός δεδομένων σε κάθε γραμμή του αρχείου.
SKIP	Αριθμός γραμμών που θα υπερπηδηθούν πρίν αρχίσει η ανάγνωση των δεδομένων.
FROM	Η απόδοση αρχικών τιμών θα ξεκινήσει από την παράμετρο με αυτό το δείκτη.
TO	Η απόδοση αρχικών τιμών θα σταματήση στην παράμετρο με αυτό το δείκτη.
FORMAT	Ο τρόπος ανάγνωσης των δεδομένων. Αναφέρεται στην αντίστοιχη εντολή Fortran που θα πραγματοποιήσει την ανάγνωση από το αρχείο. Αν το FORMAT είναι κενό τότε η ανάγνωση των δεδομένων γίνεται με τη χρήση μιας free-format εντολής read.
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Παράδειγμα χρήσης της εντολής INIT φαίνεται στον πίνακα 24.

Πίνακας 24: Ανάθεση αρχικών τιμών με την εντολή INIT.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
init  <cr>
INIT
-------------------------------------------------------------------------------
                                INIT Panel
-------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) FILE       File to read from           rinit.txt    Any string
 2) WHAT       What to initialize              r        {x,l,r,f,s,n}
 3) NPL        Numbers per line (0=free)       0        Any int >= 0
 4) SKIP       Lines to skip                   0        Any int >= 0
 5) FROM       Index to start from             1        Any int >= 1
 6) TO         Index to end to                 2        Any int >= 1
 7) FORMAT     Format, eg: E20.10                       Any string
 8) CANCEL     Cancel/Proceed Init             1        {0,1}
-------------------------------------------------------------------------------
Enter changes:
file rinit what r <cr>

2 right margin(s) have been initialized from file "rinit".

    /\/\/\/\/\/\     Merlin is at your command !!!
```

6.6 Επικοινωνία με τον κώδικα του χρήστη. Εντολές FLAG, FLAGDIS και τα υποπρογράμματα GETFLA, SETFLA

Το Merlin διαθέτει μηχανισμό επικοινωνίας μεταξύ του λειτουργικού συστήματος και των τμημάτων κώδικα που γράφει ο χρήστης (function funmin, subroutine granal κλπ.). Η επικοινωνία πραγματοποιείται μέσω 20 προκαθορισμένων μεταβλητών (flags), η τιμή των οποίων μπορεί να αλλάξει με την εντολή FLAG. Η εντολή FLAG αναμένει είσοδο μιας γραμμής με τους δείκτες και τις τιμές των flags που επιθυμούμε να αλλάξουμε. Οι τιμές όλων των flags εμφανίζονται με την εντολή FLAGDIS. Τυπικό παράδειγμα φαίνεται στον πίνακα 25.

Τις τιμές των flags μπορεί να εκμεταλευθεί ο κώδικας του χρήστη (πχ. η function funmin). Η αλλαγή τιμής σε ένα flag γίνεται με κλήση του υποπρογράμματος SETFLA:

```
CALL SETFLA ( I, F )
```

Ο δείκτης του flag που θα αλλαχτεί δίνεται ως είσοδος στη μεταβλητή I, ενώ η τιμή στη μεταβλητή F. Επιεώρηση της τιμής ενός flag γίνεται με κλήση του υποπρογράμματος GETFLA:

```
CALL GETFLA ( I, F )
```

Η μεταβλητή I καθορίζει το δείκτη του flag του οποίου ζητούμε την τιμή, ενώ η μεταβλητή F επιστρέφει την τιμή αυτή.

Πίνακας 25: Τυπικός τρόπος χρήσης των εντολών FLAG και FLAGDIS.

```
    /\/\/\/\/\/\     Merlin is at your command !!!
flag  <cr>
FLAG
1 -2 4 7.5

    /\/\/\/\/\/\     Merlin is at your command !!!
flagdis  <cr>
FLAGDIS

                            Current flag values
                            -------------------
Flag( 1)=  -2.00   Flag( 6)=    .00   Flag(11)=    .00   Flag(16)=    .00
Flag( 2)=    .00   Flag( 7)=    .00   Flag(12)=    .00   Flag(17)=    .00
Flag( 3)=    .00   Flag( 8)=    .00   Flag(13)=    .00   Flag(18)=    .00
Flag( 4)=   7.50   Flag( 9)=    .00   Flag(14)=    .00   Flag(19)=    .00
Flag( 5)=    .00   Flag(10)=    .00   Flag(15)=    .00   Flag(20)=    .00

    /\/\/\/\/\/\     Merlin is at your command !!!
```

Κεφάλαιο 7

Αρχεία

Το Merlin είναι δυνατόν να αποθηκεύσει τις τρέχουσες τιμές των παραμέτρων, τα συμβολικά τους ονόματα, τα όρια και την κατάσταση σταθεροποίησης, σε αρχεία. Στην περιγραφή που ακολουθεί, τα ονόματα των αρχείων αναφέρονται με κεφαλαίους χαρακτήρες, όπως είναι προγραμματισμένα στον πηγαίο κώδικα Fortran. Πρέπει να χρησιμοποιούνται με αυτό τον τρόπο, εάν το λειτουργικό σύστημα κάτω από το οποίο λειτουργεί το Merlin θεωρεί διαφορετικά ονόματα που εισάγονται με κεφαλαίους ή πεζούς χαρακτήρες (πχ. unix). Η μικρότερη μονάδα αποθήκευσης σε ένα αρχείο είναι η "εγγραφή". Τυπικά περιεχόμενα μιάς εγγραφής φαίνονται στον πίνακα 26.

Πίνακας 26: Τρόπος αποθήκευσης μιάς εγγραφής σε αρχείο.

```
  1) Rho       .......   1.4000000000000      -10.00        10.00
  2) Sigma     .......  -2.5000000000000      -40.00        40.00
       VALUE.......       1989.3200000000
```

7.1 Οι εντολές BACKUP, NOBACK, LASTBACK και FULLBACK

Κατά την εκτέλεση του Merlin είναι πολλές φορές είναι επιθυμητό να προστατεύσουμε τα αποτελέσματα μιας χρονοβόρας ελαχιστοποίησης από έναν απρόοπτο (μη ομαλό) τερματισμό του προγράμματος. Το Merlin έχει τη δνατότητα να αποθηκεύει τις τιμές των παραμέτρων σε αρχείο, σε προκαθορισμένες περιπτώσεις, για λόγους ασφαλείας. Οι σχετικές λεπτομέρειες ρυθμίζονται από την εντολή `BACKUP` η οποία χρησιμοποιεί ένα πίνακα ελέγχου για την εισαγωγή των απαραίτητων παραμέτρων.

Οι παράμετροι του πίνακα ελέγχου της εντολής `BACKUP` είναι:

`FILE` Το αρχείο ασφαλείας στο οποίο θα αποθηκεύονται οι εγγραφές.

`MODE` Ο τρόπος με τον οποίο θα αποθηκευτούν οι εγραφές. Επιτρεπτές τιμές είναι:

- `FULL` Κάθε εγγραφή αποθηκεύεται στο αρχείο.
- `LAST` Μόνο η πιό πρόσφατη εγγραφή αποθηκεύεται στο αρχείο. Με τον τρόπο αυτό το αρχείο ασφαλείας περιέχει πάντα μία μόνο εγγραφή.
- `NO` Δεν αποθηκεύονται εγγραφές στο αρχείο ασφαλείας.

`WHEN` Καθορίζει πότε θα αποθηκεύεται μία εγγραφή στο αρχείο ασφαλείας. Επιτρεπτές τιμές είναι:

- `M` Η εγγραφή αποθηκεύεται μετά το πέρας λειτουργίας των

ελαχιστοποιητών.

X Η εγγραφή αποθηκεύεται πριν από τυχόν αλλαγές στις παραμέτρους (όπως για παράδειγμα από την εντολή POINT).

D Η εγγραφή αποθηκεύεται πριν από τυχόν αλλαγές στα όρια, ονόματα ή κατάσταση σταθεροποίησης των παραμέτρων.

L Η εγγραφή αποθηκεύεται κάθε φορά που οι ελαχιστοποιητές βρίσκουν μια χαμηλότερη τιμή της συνάρτησης.

SAFE Επιτρεπτές τιμές είναι:

YES Το αρχείο ασφαλείας μετά από κάθε εγγραφή κλείνεται και ξαναανοίγεται, έτσι ώστε τα περιεχόμενά του να γράφονται πάντα στο δίσκο. Ο τρόπος αυτός είναι πιό ασφαλής, αλλά και πιό αργός.

NO Το αρχείο ασφαλείας παραμένει συνέχεια ανοικτό.

CANCEL BUTTON Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Επιπλέον της εντολής BACKUP οι παρακάτω εντολές χρησιμοποιούνται για τον έλεγχο της αποθήκευσης εγγραφών ασφαλείας:

NOBACK Τίποτε δεν αποθηκεύεται στο αρχείο ασφαλείας. Αυτή είναι η προκαθορισμένη συμπεριφορά. Είναι ισοδύναμη με MODE NO στην εντολή BACKUP.

LASTBACK Κάθε εγγραφή αντικαθιστά πλήρως τα περιεχόμενα του αρχείου ασφαλείας. Με τον τρόπο αυτό μόνο η πιό πρόσφατη παραμέτρων μένει κάθε φορά στο αρχείο. Είναι ισοδύναμη με MODE LAST στην εντολή BACKUP.

FULLBACK Κάθε εγγραφή προστίθεται στο τέλος του αρχείου ασφαλείας. Το μέγεθος του αρχείου μεγαλώνει καθώς ο χρήστης εργάζεται με το Merlin. Είναι ισοδύναμη με MODE FULL στην εντολή BACKUP.

7.2 Επιλεκτική αποθήκευση σημείων. Η εντολή MEMO

Η τρέχουσες τιμές των παραμέτρων μπορούν να αποθηκευτούν σε οποιοδήποτε αρχείο με την εντολή MEMO η οποία χρησιμοποιεί ένα πίνακα ελέγχου για την εισαγωγή των απαραίτητων παραμέτρων.

Οι παράμετροι του πίνακα ελέγχου της εντολής MEMO είναι:

FILE Το αρχείο στο οποίο θα προστεθεί μια εγγραφή με τις τρέχουσες τιμές των παραμέτρων.

OPEN Για λόγους ταχύτητας είναι δυνατό το αρχείο που καθορίζεται με την παράμετρο FILE να παραμένει ανοικτό μετά τη χρήση της εντολής MEMO. Επιτρεπτές τιμές για την παράμετρο OPEN είναι:

- 0 Το αρχείο κλείνεται μετά την εκτέλεση της εντολής MEMO.
- 1 Το αρχείο παραμένει ανοικτό μετά την εκτέλεση της εντολής MEMO.

CANCEL BUTTON Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

7.3 Επιθεώρηση των αρχείων Οι εντολές INSPECT και PICK

Ο χρήστης μπορεί να επιθεωρήσει αρχεία που έχουν γραφεί από τις εντολές BACKUP και MEMO με την εντολή INSPECT. Στη συνέχεια οι αποθηκευμένες τιμές μπορούν να διαβαστούν από το Merlin και να γίνουν οι τρέχουσες τιμές των παραμέτρων με τη χρήση της εντολής PICK.

Οι παράμετροι του πίνακα ελέγχου της εντολής INSPECT είναι:

FILE Το αρχείο προς επιθεώρηση.

DIR Κατεύθυνση της επιθεώρησης. Επιτρεπτές τιμές είναι:

- 1 Η επιθεώρηση του αρχείου γίνεται από την αρχή πρός το τέλος.
- -1 Η επιθεώρηση γίνεται από το τέλος προς την αρχή.

PRINT Καθορίζει τι θα τυπωθεί κατά την επιθεώρηση. Επιτρεπτές τιμές είναι:

- 0 Δεν τυπώνεται τίποτε.
- 1 Τυπώνονται μόνο οι τιμές της συνάρτησης.
- 2 Τυπώνονται οι τιμές των παραμέτρων και οι τιμές της συνάρτησης.

FROM Ο αριθμός εγγραφής από τον οποίο θα αρχίσει η επιθεώρηση.

TO Ο αριθμός εγγραφής στον οποίο θα τελειώσει η επιθεώρηση.

CANCEL BUTTON Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Οι παράμετροι του πίνακα ελέγχου της εντολής PICK είναι:

FILE Το αρχείο απο το οποίο θα διαβαστεί η εγγραφή.

REC Ο αριθμός εγγραφής που θα διαβαστεί

CANCEL BUTTON Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

Τυπικό παράδειγμα χρήσης των εντολών INSPECT και PICK φαίνεται στον πίνακα 27.

Πίνακας 27: Χρήση των εντολών INSPECT και PICK. Το επιθεωρούμενο αρχείο BACKUP έχει 2 εγγραφές.

```
    /\/\/\/\/\     Merlin is at your command !!!
inspect  <cr>
INSPECT
-------------------------------------------------------------------------------
                                INSPECT Panel
-------------------------------------------------------------------------------
Ind Keyword    Description                    Value     Menu
 1) FILE       File name to inspect          BACKUP     Any string
 2) DIR        Direction (Forw./Backw.)         1       {1,-1}
 3) PRINT      Print Nothing/Value/Point        1       {0,1,2}
 4) FROM       Record to start from (0...       0       Any int >= 0
 5) TO         Record to end to (0=ign...       0       Any int >= 0
 6) CANCEL     Cancel/Proceed inspection        1       {0,1}
-------------------------------------------------------------------------------
Enter changes:
<cr>

    Record    1      Value .......       2.07032095136927
    Record    2      Value .......       1.36034602045084

    /\/\/\/\/\     Merlin is at your command !!!
```

7.4 Η εντολές DISCARD και DELETE

Τα περιεχόμενα οποιουδήποτε αρχείου εγγραφών που έχει δημιουργηθεί από το Merlin, μπορούν να καταστραφούν με την εντολή DISCARD. Η εντολή DISCARD αφού ζητήσει το όνομα του αρχείου, καταστρέφει τα περιεχόμενά του γράφοντας μία μοναδική εγγραφή με τις πληροφορίες του τρέχοντος σημείου. Η εντολή DELETE αφού ζητήσει το όνομα ενός αρχείου, το διαγράφει πλήρως από το δίσκο. Μπορεί να χρησιμοποιηθεί όχι μόνο για αρχεία του Merlin αλλά και για οποιοδήποτε άλλο αρχείο.

7.5 Το αρχείο HELP
Οι εντολές HELP και LIST

Το αρχείο HELP περιέχει τις πληροφορίες που εμφανίζονται όταν χρησιμοποιηθεί η εντολή HELP. Απόσπασμα του αρχείου φαίνεται στον πίνακα 28. Η εντολή LIST εμφανίζει ένα συνοπτικό κατάλογο με όλες τις εντολές Merlin.

Πίνακας 28: Απόσπασμα του αρχείου HELP.

```
***MACRO
INITIATES MACRO CONSTRUCTION.  IT PROMPTS THE USER FOR A MACRO-NAME
AND CONSECUTIVELY FOR MERLIN COMMANDS.  A MACRO - NAME MUST ALWAYS
START WITH A PERIOD AND SHOULD NOT BE LONGER THAN TEN CHARACTERS.
TO END A MACRO CONSTRUCTION SESSION ENTER THE COMMAND CLEAR.

***CLEAR
INSTRUCTS MERLIN TO TERMINATE A MACRO - CONSTRUCTION SESSION.

***SIMPLEX
INVOKES THE SIMPLEX METHOD OF MINIMIZATION.  PANEL PROMPTING.
```

7.6 Τα αρχεία MACROF και CONFIG

Σχετίζονται το πρώτο με τα παραμετρικά εντολοπακέτα και το δεύτερο με τη διαμόρφωση του συστήματος. Περιγράφονται αναλυτικά στα κεφάλαια 8 και 11 αντίστοιχα.

Κεφάλαιο 8

Παραμετρικά εντολοπακέτα

Το Merlin αναγνωρίζει και εκτελεί παραμετρικά εντολοπακέτα. Ένα εντολοπακέτο είναι μια συλλογή από εντολές Merlin και κλήσεις αλλών εντολοπακέτων αποθηκευμένων σε κάποιο αρχείο. Το όνομα του κάθε εντολοπακέτου δεν πρέπει να υπερβαίνει τους 10 χαρακτήρες.

8.1 Είδη εντολοπακέτων

Υπάρχουν δύο είδη εντολοπακέτων:

α) Εντολοπακέτα δίσκου.
Βρίσκονται το καθένα σε ένα ξεχωριστό αρχείο. Το όνομα του εντολοπακέτου ταυτίζεται με το όνομα του αρχείου στο οποίο είναι αποθηκευμένο. Τυπικό παράδειγμα τέτοιου εντολοπακέτου φαίνεται στον πίνακα 29. Κατασκευάζονται από το χρήστη με τη βοήθεια κάποιου συντάκτη κειμένου.

Πίνακας 29: Τυπικό παράδειγμα εντολοπακέτου δίσκου.

```
point
1 1.4 2 -2.5
<cr>
shortdis
graddis
```

β) Εντολοπακέτα βιβλιοθήκης.
Το Merlin διατηρεί ένα ειδικό αρχείο με το όνομα MACROF στο οποίο είναι δυνατό να συνυπάρξουν πολλά εντολοπακέτα μαζί. Τα εντολοπακέτα είναι αποθηκευμένα το ένα μετά το άλλο ώς εξής:

```
*
.όνομα_εντολοπακέτου
εντολή
εντολή
...
clear
```

Η εντολή clear χρησιμοποιείται για να δηλώσει το τέλος του εντολοπακέτου. Τυπικό παράδειγμα του αρχείου MACROF με δύο εντολοπακέτα φαίνεται στον πίνακα 30. Τα εντολοπακέτα του

αρχείου MACROF μπορούν να κατασκευαστούν είτε με τον συντάκτη κειμένου που διαθέτει το λειτουργικό σύστημα, ή με την εντολή MACRO του Merlin.

Σε ένα εντολοπακέτο μπορούν να υπάρχουν κλήσεις άλλων εντολοπακέτων, αναδρομικές κλήσεις του ιδίου ή άλλων εντολοπακέτων δεν επιτρέπονται.

Πίνακας 30: Τυπικό παράδειγμα του αρχείου MACROF με δύο εντολοπακέτα.

```
*
.adis
anal
graddis
valdis
clear
*
.na
godfather
1 rho 2 sigma
shortdis
clear
```

8.2 Κατασκευή εντολοπακέτων. Η εντολή MACRO

Η εντολή MACRO βοηθά το χρήστη να κατασκευάσει ένα εντολοπακέτο και να το τοποθετήσει στο αρχείο MACROF. Αρχικά ζητείται το όνομα του εντολοπακέτου και κατόπιν οι εντολές που το αποτελούν. Ο τερματισμός της διαδικασίας γίνεται με την εισαγωγή της εντολής CLEAR. Τυπικό παράδειγμα κατασκευής εντολοπακέτου φαίνεται στον πίνακα 31.

Πίνακας 31: Κατασκευή εντολοπακέτου με την εντολή MACRO.

```
    /\/\/\/\/\     Merlin is at your command !!!
macro  <cr>
MACRO

** Enter macro name.  Exit with a blank line **
.na  <cr>

** Enter desired command.  Exit with CLEAR **
godfather  <cr>

** Enter desired command.  Exit with CLEAR **
1  Rho  2  Sigma  <cr>

** Enter desired command.  Exit with CLEAR **
shortdis  <cr>

** Enter desired command.  Exit with CLEAR **
clear  <cr>

** Macro editing complete **

    /\/\/\/\/\     Merlin is at your command !!!
```

8.3 Κλήση των εντολοπακέτων. Ορίσματα

Οι κλήσεις των εντολοπακέτων ξεχωρίζουν από τις εντολές Merlin από την αρχική τελεία. Τυπικό παράδειγμα χρήσης εντολοπακέτου φαίνεται στον πίνακα 32.

Τα εντολοπακέτα είναι δυνατόν να δέχονται ορίσματα κατά την κλήση τους. Τα ορίσματα αυτά αναφέρονται ως [1], [2] κλπ. μέσα στο εντολοπακέτο και αντικαθίστανται από τα πραγματικά ορίσματα που έχει εισάγει ο χρήστης κατά την κλήση του. Πχ. το παρακάτω εντολοπακέτο με όνομα .set θέτει ώς τιμή στην παράμετρο x_2 της συνάρτησης το πρώτο όρισμα που εισάγει ο χρήστης.

```
point
2 [1]
<cr>
shortdis
```

Παραδείγματα κλήσης του παραπάνω εντολοπακέτου είναι:

```
.set 7
.set -2.5
.set 68.23
```

Το σύμβολο [0] αντιπροσωπεύει το όνομα του εντολοπακέτου, ενώ το [#] τον αριθμό των ορισμάτων που έχει εισάγει ο χρήστης κατά την κλήση του εντολοπακέτου. Στο ανωτέρω παράδειγμα το [0] θα ήταν set, ενώ το [#] θα ήταν 1. Αν ο χρήστης εισάγει λιγότερα ορίσματα από αυτά που χρειάζεται το εντολοπακέτο, τότε η θέση των επιπλέον ορισμάτων μένει κενή. Ο αριθμός των ορισμάτων που μπορεί να έχει ένα εντολοπακέτο περιορίζεται μόνο από το μέγεθος της γραμμής εισόδου, που είναι 80 χαρακτήρες.

Πίνακας 32: Κλήση εντολοπακέτου.

```
/\/\/\/\/\     Merlin is at your command !!!
.adis  <cr>

            ...  Execution of macro .adis has been initiated  ...
ANAL
GRADDIS

Index  Name      Fix       Gradient                Gradient-Mode
-------------------------------------------------------------
    1)           .......  1.920600000000000E+04        ANAL
    2)           ....... -2.400000000000000E+03        ANAL
VALDIS
-------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                           1               1              0
Since last reset:                1               1              0
-------------------------------------------------------------------------------

               Current value:   14409.0000000000
CLEAR

            ...  Execution of macro .adis has been completed  ...

    /\/\/\/\/\     Merlin is at your command !!!
```

Κεφάλαιο 9

Οι αλγόριθμοι ελαχιστοποίησης

9.1 Συμβολισμός

Παρακάτω δίνεται μία περιληπτική περιγραφή των αλγορίθμων ελαχιστοποίησης του Merlin καθώς και η σημασία των παραμέτρων στους πίνακες ελέγχου. Τα μικρά έντονα γράμματα δηλώνουν διανύσματα, ενώ τα έντονα κεφαλαία δηλώνουν πίνακες. Επειδή όλες οι μέθοδοι είναι επαναληπτικές, οι ποσότητες που σχετίζονται με την επανάληψη k δηλώνονται με ένα άνω δείκτη σε παρενθέσεις. Πχ. $\mathbf{x}^{(k)}$. Τα χρησιμοποιούμενα σύμβολα και η σημασία τους είναι:

g Το διάνυσμα των πρώτων παραγώγων της συνάρτησης με στοιχεία: $g_i = \frac{\partial f}{\partial x_i}$

G Ο πίνακας των δεύτερων παραγώγων της συνάρτησης με στοιχεία $G_{ij} = \frac{\partial^2 f}{\partial x_i \, \partial x_i}$

H Ο αντίστροφος του πίνακα των δευτέρων παραγώγων με στοιχεία $H_{ij} = \left[\mathbf{G}^{-1}\right]_{ij}$

9.2 Τερματισμός των ελαχιστοποιητών Εντολές TARGET και NOTARGET

Όλοι οι αλγόριθμοι ελαχιστοποίησης τερματίζουν τη λειτουργία τους όταν ξεπεράσουν ένα δοσμένο αριθμό κλήσεων της αντικειμενικής συνάρτησης. Σε αυτή την περίπτωση ο χρήστης ενημερώνεται από διαγνωστικό της μορφής:

```
-------------------------------------------------------------------------------
                BFGS: All function evaluations have been used
-------------------------------------------------------------------------------
```

Οι μέθοδοι BFGS, DFP, CONGRA και ROLL τερματίζουν τη λειτουργία τους, όταν φτάσουν σε μιά τιμή-στόχο. Ο έλεγχος αυτής της τιμής γίνεται με τις εντολές:

TARGET Θέτει την τιμή-στόχο γιά τις μεθόδους BFGS, DFP, CONGRA και ROLL

NOTARGET Ακυρώνει την τιμή-στόχο.

Ο χρήστης ενημερώνεται για την επίτευξη της τιμής-στόχου με το διαγνωστικό:

```
-------------------------------------------------------------------------------
                BFGS: Target value has been reached
-------------------------------------------------------------------------------
```

Ορισμένοι από τους ελαχιστοποιητές χρησιμοποιούν και επιπλέον κριτήρια για τερματισμό. Αυτά αναφέρονται όπου χρειάζεται.

9.3 Κοινές παράμετροι

Στους πίνακες ελέγχου όλων των ελαχιστοποιητών υπάρχουν οι παρακάτω παράμετροι:

NUMBER OF CALLS — Προσεγγιστικός μέγιστος επιτρεπτός αριθμός κλήσεων της αντικειμενικής συνάρτησης. Στον αριθμό αυτό συμπεριλαμβάνονται και οι κλήσεις που θα δαπανηθούν για τον υπολογισμό των παραγώγων όταν αυτές πρέπει να προσεγγιστούν αριθμητικά.

ITERATIONS — Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου. Η τιμή -1 υπονοεί ένα άπειρο αριθμό επαναλήψεων (έως ότου η μέθοδος τερματίσει για κάποιο άλλο λόγο)

PRINTOUT LEVEL — Επιλογή της εξόδου από τη μέθοδο. Επιτρεπτές τιμές είναι:

- 0 Η μέθοδος δεν παράγει καθόλου έξοδο.
- 1 Η μέθοδος αναφέρει την τιμή της αντικειμενικής συνάρτησης κάθε φορά που βρίσκει ένα χαμηλότερο σημείο όπως φαίνεται στον πίνακα 12. Αυτή είναι η προκαθορισμένη συμπεριφορά.
- 2 Η μέθοδος αναφέρει την τιμή της αντικειμενικής συνάρτησης και τις τιμές όλων των παραμέτρων κάθε φορά που βρίσκει χαμηλότερο σημείο.

CANCEL-BUTTON — Ελέγχει την εκτέλεση της μεθόδου μετά από τις αλλαγές των παραμέτρων στον πίνακα ελέγχου. Επιτρεπτές τιμές είναι:

- 0 Οι αλλαγές του χρήστη στις παραμέτρους του πίνακα ελέγχου καταχωρούνται, αλλά η μέθοδος ελαχιστοποίησης δεν εκτελείται.
- 1 Η μέθοδος ελαχιστοποίησης εκτελείται. Αυτή είναι η προκαθορισμένη συμπεριφορά.

Οι μέθοδοι ελαχιστοποίησης που χρησιμοποιούν πρώτες παραγώγους (BFGS, DFP, CONGRA, TRUST και TOLMIN) περιέχουν τις παραμέτρους ελέγχου:

GRADIENT-MODE — Ο τρόπος υπολογισμού των παραγώγων. Επιτρεπτές τιμές είναι:

- 0 Κατά την εκτέλεση της μεθόδου, οι παράγωγοι υπολογίζονται διαρκώς με τον τρόπο που έχει προκαθορίσει ο χρήστης (FAST, QUAD, NUMER, ANAL ή MIXED). Αυτή είναι η προκαθορισμένη συμπεριφορά.
- 1 Καθώς το τοπικό ελάχιστο προσεγγίζεται απαιτούνται παράγωγοι μεγαλύτερης ακρίβειας. Χρησιμοποιείται η αυτόματη αλλαγή παραγώγων όπως περιγράφεται στην παράγραφο 4.3.

Οι μέθοδοι ελαχιστοποίησης που χρησιμοποιούν γραμμικές αναζητήσεις (οι μέθοδοι γραμικής αναζήτησης που χρησιμοποιούνται από το Merlin περιγράφονται στην αναφορά [24]) περιέχουν τις παράμετρους ελέγχου:

`LS METHOD`	Μέθοδος γραμμικής αναζήτησης. Οι επιτρεπτές τιμές είναι: 1 Εφαρμόζονται οι ασθενείς συνθήκες Wolfe-Powell. 2 Εφαρμόζονται οι ισχυρές συνθήκες Wolfe-Powell.
`MAX. LS ITERATIONS`	Μέγιστος αριθμός εσωτερικών επαναλήψεων της γραμμικής αναζήτησης.
`LINE SEARCH RHO`	Η παράμετρος ρ της γραμμικής αναζήτησης.
`LINE SEARCH SIGMA`	Η παράμετρος σ της γραμμικής αναζήτησης.

9.4 BFGS

0) Δίνεται αρχική τιμή στον πίνακα $\mathbf{H}$ και υπολογίζεται το διάνυσμα των πρώτων παραγώγων $\mathbf{g}^{(1)}$.

Κατόπιν για την επανάληψη k:

1) Υπολογισμός της διεύθυνσης $\mathbf{s}^{(k)} = -\mathbf{H}^{(k)}\,\mathbf{g}^{(k)}$
2) Γραμμική αναζήτηση στη διεύθυνση $\mathbf{s}^{(k)}$ που δίνει $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$
3) Ενημέρωση του $\mathbf{H}^{(k)}$ από τον τύπο (οι άνω δείκτες (k) έχουν παραλειφθεί χάριν απλότητος):

$$\mathbf{H}^{(k+1)} = \mathbf{H} + \left(1 + \frac{\gamma^T \mathbf{H} \gamma}{\delta^T \gamma}\right)\frac{\delta\delta^T}{\delta^T\gamma} - \frac{\delta\gamma^T\mathbf{H} + \mathbf{H}\gamma\delta^T}{\delta^T\gamma}$$

με $\boldsymbol{\gamma}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$

Ο πίνακας ελέγχου της `BFGS` φαίνεται στον πίνακα 12. Οι παράμετροι που περιέχει είναι:

`NUMBER OF CALLS`	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
`USE/RECALCULATE GRADIENT`	Ελέγχει τον υπολογισμό ή όχι των πρώτων παραγώγων πρίν την έναρξη των επαναλήψεων, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι: 0 Οι παράγωγοι $\mathbf{g}^{(1)}$ υπολογίζονται πρίν την έναρξη των επαναλήψεων. 1 Οι παράγωγοι δεν υπολογίζονται πρίν την έναρξη των επαναλήψεων, αλλά χρησιμοποιείται η τευλευταία υπολογεισθείσα τιμή.
`USE/RECALCULATE HESSIAN`	Ελέγχει την αρχική τιμή που αποδίδεται στον πίνακα $\mathbf{H}$, πρίν αρχίσουν οι επαναλήψεις, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι:

0 Ο μοναδιαίος πίνακας δίνεται σαν αρχική τιμή του πίνακα **H**.

1 Ο πίνακας **H** διατηρεί τις τιμές που είχε μετά το τέλος της προηγούμενης ελαχιστοποίησης με BFGS ή DFP.

ITERATIONS	Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου.
GRADIENT MODE	Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).
X-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραμέτρων.
G-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραγώγων.
F-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές της συνάρτησης.
LS METHOD	Μέθοδος γραμμικής αναζήτησης.
MAX. LS ITERATIONS	Μέγιστος αριθμός εσωτερικών επαναλήψεων της γραμμικής αναζήτησης.
LINE SEARCH RHO	Η παράμετρος ρ της γραμμικής αναζήτησης.
LINE SEARCH SIGMA	Η παράμετρος σ της γραμμικής αναζήτησης.
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.5 DFP

0) Δίνεται αρχική τιμή στον πίνακα **H** και υπολογίζεται το διάνυσμα των πρώτων παραγώγων $\mathbf{g}^{(1)}$.

Κατόπιν για την επανάληψη k:

1) Υπολογισμός της διεύθυνσης $\mathbf{s}^{(k)} = -\mathbf{H}^{(k)}\,\mathbf{g}^{(k)}$
2) Γραμμική αναζήτηση στη διεύθυνση $\mathbf{s}^{(k)}$ που δίνει $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$
3) Ενημέρωση του $\mathbf{H}^{(k)}$ από τον τύπο (οι άνω δείκτες (k) έχουν παραλειφθεί χάριν απλότητος):

$$\mathbf{H}^{(k+1)} = \mathbf{H} + \frac{\boldsymbol{\delta}\,\boldsymbol{\delta}^{T}}{\boldsymbol{\delta}^{T}\,\boldsymbol{\gamma}} - \frac{\mathbf{H}\,\boldsymbol{\gamma}\,\boldsymbol{\gamma}^{T}\,\mathbf{H}}{\boldsymbol{\gamma}^{T}\,\mathbf{H}\,\boldsymbol{\gamma}}$$

με $\boldsymbol{\gamma}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της DFP είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
USE/RECALCULATE GRADIENT	Ελέγχει τον υπολογισμό ή όχι των πρώτων παραγώγων πρίν την έναρξη των επαναλήψεων, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι:

0 Οι παράγωγοι $\mathbf{g}^{(1)}$ υπολογίζονται πρίν την έναρξη των επαναλήψεων.

1 Οι παράγωγοι δεν υπολογίζονται πρίν την έναρξη των επαναλήψεων, αλλά χρησιμοποιείται η τευλευταία υπολογεισθείσα τιμή.

`USE/RECALCULATE HESSIAN` Ελέγχει την αρχική τιμή που αποδίδεται στον πίνακα **H**, πρίν αρχίσουν οι επαναλήψεις, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι:

0 Ο μοναδιαίος πίνακας δίνεται σαν αρχική τιμή του πίνακα **H**.

1 Ο πίνακας **H** διατηρεί τις τιμές που είχε μετά το τέλος της προηγούμενης ελαχιστοποίησης με `BFGS` ή `DFP`.

`ITERATIONS` Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου.

`GRADIENT MODE` Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).

`X-CONVERGENCE CRITERION` Κριτήριο τερματισμού για τις τιμές των παραμέτρων.

`G-CONVERGENCE CRITERION` Κριτήριο τερματισμού για τις τιμές των παραγώγων.

`F-CONVERGENCE CRITERION` Κριτήριο τερματισμού για τις τιμές της συνάρτησης.

`LS METHOD` Μέθοδος γραμμικής αναζήτησης.

`MAX. LS ITERATIONS` Μέγιστος αριθμός εσωτερικών επαναλήψεων της γραμμικής αναζήτησης.

`LINE SEARCH RHO` Η παράμετρος ρ της γραμμικής αναζήτησης.

`LINE SEARCH SIGMA` Η παράμετρος σ της γραμμικής αναζήτησης.

`PRINTOUT LEVEL` Έλεγχος εξόδου (βλέπε παράγραφο 9.3).

`CANCEL BUTTON` Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.6 CONGRA

0) Υπολογίζεται το διάνυσμα των πρώτων παραγώγων $\mathbf{g}^{(1)}$ και τίθεται $\mathbf{s}^{(1)} = -\mathbf{g}^{(1)}$.

Γιά την επανάληψη k:

1) Υπολογισμός της διεύθυνσης $\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} + \beta^{(k)}\,\mathbf{s}^{(k-1)}$
όπου εάν χρησιμοποιηθεί ο τύπος των Polak και Ribiere [8], είναι:

$$\beta^{(k)} = \frac{(\mathbf{g}^{(k)} - \mathbf{g}^{(k-1)})^{T}\,\mathbf{g}^{(k)}}{\mathbf{g}^{(k)T}\,\mathbf{g}^{(k-1)}}$$

ενώ αν χρησιμοποιηθεί ο τύπος των Fletcher και Reeves [16], είναι:

$$\beta^{(k)} = \frac{\mathbf{g}^{(k)T}\,\mathbf{g}^{(k)}}{\mathbf{g}^{(k-1)T}\,\mathbf{g}^{(k-1)}}$$

2) Γραμμική αναζήτηση στη διεύθυνση $\mathbf{s}^{(k)}$ που θα δώσει το $\mathbf{x}^{(k+1)}$

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της CONGRA είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
USE/RECALCULATE GRADIENT	Όπως γιά την BFGS (βλέπε παράγραφο 9.4).
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
GRADIENT MODE	Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).
F-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές της συνάρτησης.
METHOD	Επιλογή του τύπου γιά το $\beta^{(k)}$. Επιτρεπτές τιμές είναι: 0 Χρησιμοποιείται ο τύπος των Polak και Ribiere. Αυτή είναι η προκαθορισμένη τιμή. 1 Χρησιμοποιείται ο τύπος των Fletcher και Reeves.
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).
X-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραμέτρων.
G-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραγώγων.
MAX. LS ITERATIONS	Μέγιστος αριθμός εσωτερικών επαναλήψεων της γραμμικής αναζήτησης.
LINE SEARCH RHO	Η παράμετρος ρ της γραμμικής αναζήτησης.
LINE SEARCH SIGMA	Η παράμετρος σ της γραμμικής αναζήτησης.
ITERATIONS	Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου.
RESTART	Καθορίζει αν η διεύθυνση αναζήτησης θα επαναπροσδιορίζεται στη διεύθυνση μέγιστης καθόδου κάθε N επαναλήψεις. Επιτρεπτές τιμές είναι: 0 Δεν γίνεται επαναπροσδιορισμός της διεύθυνσης αναζήτησης. 1 Η διεύθυνση αναζήτησης επαναπροσδιορίζεται στη διεύθυνση μέγιστης καθόδου κάθε N επαναλήψεις.

9.7 ROLL

Έστω $\mathbf{x}^c$ το τρέχων σημείο και $f_c = f(\mathbf{x}^c)$ η τιμή της συνάρτησης. Έστω $\mathbf{s}$ ένα διάνυσμα δοκιμαστικών βημάτων σε κάθε διεύθυνση. Παρακάτω περιγράφεται ένας κύκλος λειτουργίας της ROLL που επαναλαμβάνεται γιά όλες τις παραμέτρους της συνάρτησης (i=1,N)

1) Επιλογή ενός δοκιμαστικού σημείου $\mathbf{x}^t$, έτσι ώστε $x_j^t = x_j^c$ γιά όλα τα $j \neq i$ και $x_i^t = x_i^c + s_i$
2) Υπολογισμός της $f_+ = f(\mathbf{x}^t)$
3) Εάν $f_+ < f_c$ θέτουμε $\mathbf{x}^c = \mathbf{x}^t$, $f_c = f_+$, $s_i = \alpha\, s_i$ και συνεχίζουμε από το βήμα 1 με το επόμενο i.

4) Εάν $f_+ \geq f_c$ επιλέγουμε ένα άλλο δοκιμαστικό σημείο $\mathbf{x}^t$, με $x_j^t = x_j^c$ γιά όλα τα $j \neq i$ και $x_i^t = x_i^c - s_i$
5) Υπολογισμός της $f_- = f(\mathbf{x}^t)$
6) Εάν $f_- < f_c$ θέτουμε $\mathbf{x}^c = \mathbf{x}^t$, $f_c = f_-$, $s_i = -\alpha\, s_i$ και συνεχίζουμε από το βήμα 1 με το επόμενο i.
7) Υπολογισμός ενός κατάλληλου βήματος ως: $s_i = \frac{1}{2} \frac{f_+ - f_-}{f_+ + f_- - 2f_c} s_i$

Άν μετά ένα πλήρη κύκλο δεν παρατηρηθεί πρόοδος, πραγματοποιείται μιά γραμμική αναζήτηση στη διεύθυνση **s**.

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της ROLL είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
LINE SEARCH TOLERANCE	Ακρίβεια της γραμμικής αναζήτησης (βλέπε παράγραφο 9.3).
STEP FACTOR	Η παράμετρος α που ρυθμίζει το μέγεθος των βημάτων στα σημεία 3 και 6 του αλγορίθμου.
FAILURES ALLOWED	Ο επιτρεπτός αριθμός διαδοχικών αποτυχημένων κύκλων της μεθόδου. Η μέθοδος τερματίζει τη λειτουργία της όταν υπάρξουν τόσες διαδοχικές αποτυχίες.
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
MIN. RELATIVE DROP	Ελάχιστη επιτρεπτή μείωση της συνάρτησης σε ένα κύκλο (βλέπε παράγραφο 9.3).
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).
LINE SEARCH CALLS	Μέγιστος αριθμός κλήσεων από τη γραμμική αναζήτηση.

9.8 SIMPLEX

1) Ξεκινώντας από το τρέχον σημείο $\mathbf{x}^c$ κατασκευάζουμε τις κορυφές simplex $\mathbf{p}_i$, i=0,N (με $\mathbf{p}_0 = \mathbf{x}^c$) και τις αντίστοιχες τιμές της συνάρτησης $y_i = f(\mathbf{p}_i)$.
2) Εύρεση της χαμηλότερης και υψηλότερης τιμής y_{low} και y_{high} και των αντίστοιχων κορυφών $\mathbf{p}_{low}$ και $\mathbf{p}_{high}$:
   $y_{low} = \min\{ y_i, i=0,N \}$ $y_{high} = \max\{ y_i, i=0,N \}$
   Υπολογισμός του κεντροειδούς $\mathbf{p}^c$ όλων των $\mathbf{p}_i$, εκτός από το $\mathbf{p}_{high}$.
   Θέτουμε $\mathbf{p}^* = (1+\alpha)\,\mathbf{p}^c - \alpha\,\mathbf{p}_{high}$ και υπολογίζουμε το $y^* = f(\mathbf{p}^*)$
3) Εάν $y^* < y_{low}$ τότε θέτουμε $\mathbf{p}^{**} = (1+\gamma)\,\mathbf{p}^* - \gamma\,\mathbf{p}^c$ και υπολογίζουμε το $y^{**} = f(\mathbf{p}^{**})$
   Εάν $y^{**} < y^*$ τότε αντικαθιστούμε το $\mathbf{p}_{high}$ από το $\mathbf{p}^{**}$ ειδάλως αντικαθιστούμε το $\mathbf{p}_{high}$ από το $\mathbf{p}^*$.

Η διαδικασία συνεχίζεται στο βήμα 5 με έλεγχο του κριτηρίου σύγκλισης.

4) Εάν $y^{*} > y_i$ γιά όλα τα i εκτός από το y_{high} τότε

   4α) Εάν $y^{*} < y_{high}$ αντικαθιστούμε το $\mathbf{p}_{high}$ από το $\mathbf{p}^{*}$.
   Θέτουμε $\mathbf{p}^{**} = (1-\beta)\,\mathbf{p}^{c} - \beta\,\mathbf{p}_{high}$ και υπολογίζουμε το $y^{**} = f(\mathbf{p}^{**})$.

   4β) Εάν $y^{**} > y_{high}$ τότε αντικαθιστούμε τα $\mathbf{p}_i$ από το $\frac{\mathbf{p}_i + \mathbf{p}_{low}}{2}$ γιά i=0,N
   αλλιώς αντικαθιστούμε το $\mathbf{p}_{high}$ από το $\mathbf{p}^{**}$.

   Ειδάλως αντικαθιστούμε το $\mathbf{p}_{high}$ από το $\mathbf{p}^{*}$.

5) Ελέγχουμε το κριτήριο σύγκλισης. Εάν ικανοποιείται επιστρέφουμε τα y_{low} , $\mathbf{p}_{low}$ αλλιώς επαναλαμβάνουμε από το βήμα 2. Χρησιμοποιείται το κριτήριο σύγκλισης:

$$\frac{1}{N+1}\sum_{i=0}^{N}\left|y_i - f(\mathbf{p}^{c})\right| \quad < \varepsilon$$

όπου ε είναι ένας προκαθορισμένος αριθμός. Οι τιμές των σταθερών α, β και γ είναι α=1, $\beta=\frac{1}{2}$ και γ=2 όπως προτείνεται στην αναφορά [17].

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της SIMPLEX είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
SIMPLEX TOLERANCE	Καθορίζει την παράμετρο ε στο κριτήριο σύγκλισης.
X-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραμέτρων.
ITERATIONS	Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου.
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
REFLECTION FACTOR	Η σταθερά α.
CONTRACTION FACTOR	Η σταθερά β.
EXPANSION FACTOR	Η σταθερά γ.
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.9 TOLMIN

Περιγράφεται αναλυτικά στις αναφορές [19,20].

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της TOLMIN είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
TERMINATION ACCURACY	Η απαιτούμενη ακρίβεια για τον τερματισμό της μεθόδου [20].
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
GRADIENT MODE	Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.10 LEVE

Περιγράφεται αναλυτικά στις αναφορές [21,22]. Γιά να χρησιμοποιηθεί πρέπει να έχουμε θέσει το Merlin στον τρόπο λειτουργίας SOS (βλέπε παράγραφο 5.2) και να έχουμε προγραμματίσει την `SUBROUTINE SUBSUM` (παράγραφος 2.3).

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της `LEVE` είναι:

`NUMBER OF CALLS`	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
`F-TOLERANCE`	Κριτήριο τερματισμού βασισμένο στις τιμές της συνάρτησης [22].
`X-TOLERANCE`	Κριτήριο τερματισμού βασισμένο στις τιμές των παραμέτρων [22].
`G-TOLERANCE`	Κριτήριο τερματισμού βασισμένο στις τιμές των παραγώγων [22].
`STEP FACTOR`	Συντελεστής αλλαγής βήματος [22].
`FUNCTION ACCURACY`	Η ακρίβεια με την οποία υπολογίζεται η συνάρτηση του χρήστη (`SUBROUTINE SUBSUM`).
`PRINTOUT LEVEL`	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
`CANCEL BUTTON`	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.11 TRUST

Πρόκειται για μέθοδο που χρησιμοποιεί διαστήματα εμπιστοσύνης και ενημερώνει τον πίνακα **H** χρησιμοποιώντας τον τύπο BFGS.

0) Δίνεται αρχική τιμή στον πίνακα **H** και υπολογίζεται το διάνυσμα των πρώτων παραγώγων $\mathbf{g}^{(1)}$. Επίσης δίνεται αρχική τιμή στο διάστημα εμπιστοσύνης $R^{(1)}$.

Κατόπιν για την επανάληψη k:

1) Υπολογισμός του βήματος $\boldsymbol{\delta}^{(k)}$ χρησιμοποιώντας την τεχνική double dogleg (δες αναφορά [7, σελ139).
2) Υπολογισμός του λόγου εμπιστοσύνης $r^{(k)} = \dfrac{f^{(k)} - f(\mathbf{x}^{(k)} + \delta^{(k)})}{f^{(k)} - q^{(k)}(\delta^{(k)})}$ της πραγματικής προς την αναμενόμενη πτώση της συνάρτησης χρησιμοποιώντας το τετραγωνικό μοντέλο:
$$q(\delta) = f(\mathbf{x}) + \delta^T \mathbf{g} + \frac{1}{2}\delta^T \mathbf{G} \delta^T$$
3) Εάν $r^{(k)} < 1/4$ τότε
$R^{(k+1)} = || \delta^{(k)}|| / 4$
αλλοιώς εάν $r^{(k)} > 3/4$ και $|| \delta^{(k)}|| = R^{(k)}$ τότε
$R^{(k+1)} = 2\, R^{(k)}$
αλλοιώς
$R^{(k+1)} = R^{(k)}$
4) Εάν $r^{(k)} \le 0$ τότε
$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$

αλλοιώς

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$$

5) Ενημέρωση του $\mathbf{H}^{(k)}$ από τον τύπο BFGS (δες παράγραφο 9.4).

Οι παράμετροι που περιέχονται στον πίνακα ελέγχου της TRUST είναι:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
USE/RECALCULATE GRADIENT	Ελέγχει τον υπολογισμό ή όχι των πρώτων παραγώγων πρίν την έναρξη των επαναλήψεων, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι: 0 Οι παράγωγοι $\mathbf{g}^{(1)}$ υπολογίζονται πρίν την έναρξη των επαναλήψεων. 1 Οι παράγωγοι δεν υπολογίζονται πρίν την έναρξη των επαναλήψεων, αλλά χρησιμοποιείται η τευλευταία υπολογεισθείσα τιμή.
USE/RECALCULATE HESSIAN	Ελέγχει την αρχική τιμή που αποδίδεται στον πίνακα $\mathbf{H}$, πρίν αρχίσουν οι επαναλήψεις, στο βήμα 0 του αλγορίθμου. Επιτρεπτές τιμές είναι: 0 Ο μοναδιαίος πίνακας δίνεται σαν αρχική τιμή του πίνακα $\mathbf{H}$. 1 Ο πίνακας $\mathbf{H}$ διατηρεί τις τιμές που είχε μετά το τέλος της προηγούμενης ελαχιστοποίησης με BFGS ή DFP.
USE/RECALCULATE RADIUS	Ελέγχει την αρχική τιμή που αποδίδεται στο διάστημα εμπιστοσύνης R, πριν αρχίσουν οι επαναλήψεις. 0 1 Χρησιμοποιείται η τιμή του R από την τελευταία κλήση της μεθόδου.
ITERATIONS	Ο επιτρεπτός αριθμός επαναλήψεων της μεθόδου.
GRADIENT MODE	Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).
X-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραμέτρων.
G-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές των παραγώγων.
F-CONVERGENCE CRITERION	Κριτήριο τερματισμού για τις τιμές της συνάρτησης.
LS METHOD	Μέθοδος γραμμικής αναζήτησης.
MAX. LS ITERATIONS	Μέγιστος αριθμός εσωτερικών επαναλήψεων της γραμμικής αναζήτησης.
PRINTOUT LEVEL	Έλεγχος εξόδου (βλέπε παράγραφο 9.3).
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.12 AUTO

Εκτός από τους ήδη αναφερθέντες αλγόριθμους ελαχιστοποίησης, το Merlin περιέχει την εντολή AUTO που είναι ένας συνδιασμός των ROLL, SIMPLEX, BFGS και TRUST. Η AUTO δοκιμάζει διαδοχικά τους τέσσερεις ελαχιστοποιητές και χρησιμοποιεί κάθε φορά αυτόν με την καλύτερη απόδοση.

Ο πίνακας ελέγχου της AUTO περιέχει τις παραμέτρους:

NUMBER OF CALLS	Προσεγγιστικός μέγιστος αριθμός κλήσεων (βλέπε παράγραφο 9.3).
TARGET VALUE	Τιμή στόχος. Όταν η τιμή της συνάρτησης γίνει μικρότερη από την τιμή στόχο η διαδικασία AUTO θα τερματιστεί.
GRADIENT MODE	Τρόπος υπολογισμού παραγώγων (βλέπε παράγραφο 9.3).
CANCEL BUTTON	Ακύρωση εκτέλεσης (βλέπε παράγραφο 9.3).

9.13 Οι γραμμικοί περιορισμοί

Ο αλγόριθμος TOLMIN μέσω του υποπρογράμματος UCLDAT ενημερώνεται για τους γραμμικούς περιορισμούς και χρησιμοποιεί τη μέθοδο του Powell (active set method) [19,20]. Οι υπόλοιποι αλγόριθμοι αντιμετωπίζουν τους περιορισμούς με μεθόδους penalty και barrier (για ανισότητες μόνο) και υλοποιούνται μέσω προγραμμάτων MCL.

9.13.1 Μέθοδοι penalty

Η συνάρτηση κόστους γράφεται:

$$F(\mathbf{x}) = f(\mathbf{x}) + r\left[\sum_{i=1}^{meq} \Psi_i(c_i(\mathbf{x})) + \sum_{i=meq+1}^{m} \Phi(c_i(\mathbf{x}))\right]$$

όπου συμβολίζουμε με $c_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, meq$ τους περιορισμούς ισότητας (στην περίπτωσή μας $c_i(\mathbf{x}) = \mathbf{a}_i^T\mathbf{x} - b_i$ και με $c_i(x) \geq 0 \quad i=meq+1, \ldots, m$ τους ανισοτικούς $c_i(\mathbf{x}) = \mathbf{a}_i^T\mathbf{x} - b_i$

Οι συναρτήσεις Ψ και Φ πρέπει να έχουν επιλεχθεί κατάλληλα. Δύο είναι οι συνήθεις επιλογές κατ' αρχήν για την Ψ:

$$\Psi(c_i(\mathbf{x})) = [c_i(\mathbf{x})]^2$$

ή

$$\Psi(c_i(\mathbf{x})) = |c_i(\mathbf{x})|$$

και για την Φ αντίστοιχα:

$$\Phi(c_i(\mathbf{x})) = \{\min[0, c_i(\mathbf{x})]\}^2$$

ή

$$\Phi(c_i(\mathbf{x})) = -\min[0, c_i(\mathbf{x})]$$

Η παράμετρος r ρυθμίζεται από τον αλγόριθμο όπως φαίνεται παρακάτω.
Δοθέντος ενός αρχικού σημείου $x^{(0)}$, θέτουμε r = 1 και k = 0, s > 1

1) Ελαχιστοποιείται η συνάρτηση:

$$F(\mathbf{x}) = f(\mathbf{x}) + r\left[\sum_{i=1}^{meq} \Psi_i(c_i(\mathbf{x})) + \sum_{i=meq+1}^{m} \Phi(c_i(\mathbf{x}))\right]$$

και εντοπίζεται η θέση του ελαχίστου x^*
$k = k+1, \quad r = r * s \quad x^{(k)} = x^*$

2) Ελέγχονται τα κριτήρια τερματισμού.

3) Συνέχεια από το βήμα 1.

9.13.2 Μέθοδοι barrier

Στην περίπτωση των ανισοτικών περιορισμών μπορούμε να εφαρμόσουμε τεχνικές φραγμάτων (Barrier).
Η συνάρτηση κόστους γράφεται:

$$F(\mathbf{x}) = f(\mathbf{x}) + r\sum_{i=1}^{m} b_i(c_i(\mathbf{x}))$$

όπου $b_i(t) \rightarrow \infty$
$t \rightarrow 0^+$

Παραδείγματα συναρτήσεων για το b_i είναι:

$$b_i(c_i(\mathbf{x})) = \frac{1}{c_i(\mathbf{x})} \qquad \text{(Carrol)}$$

$$b_i(c_i(\mathbf{x})) = -\log[c_i(\mathbf{x})] \qquad \text{(Frisch)}$$

Οι παραπάνω τεχνικές Penalty + Barrier μπορούν να υλοποιηθούν άμεσα μέσω της MCL. Η παράμετρος r μπορεί να ρυθμίζεται μέσω της εντολής `FLAG` που θέτει την επιθυμούμενη τιμή σε μιά "flag" παράμετρο. (Το Merlin αυτή τη στιγμή διαθέτει 20 REAL "flag" παραμέτρους).
Η συνάρτηση funmin πρέπει κατάλληλα να γραφτεί για να χρησιμοποιεί την παράμετρο "flag".

```
    Function funmin (x, n)
    Dimension x (n)
→   call getflag (1, R)
    Funmin = f (x) + R * C (x)
    End
```

Οπου η συνάρτηση f(x) είναι η αντικειμενική συνάρτηση ανεξάρτητη από τους περιορισμούς, και C(x) είναι η συνάρτηση

$$C(x) = \left[\sum_{i=1}^{meq} \Psi_i(c_i(\mathbf{x})) + \sum_{i=meq+1}^{m} \Phi(c_i(\mathbf{x})) \right]$$

για την περίπτωση των τεχνικών penalty και αντίστοιχα για τις τεχνικές τύπου Barrier. Σημειώστε ότι οι τεχνικές Barrier απαιτούν το αρχικό σημείο $x^{(0)}$ να είναι εφικτό (feasible), ενώ αυτό δεν είναι απαραίτητο για τις τεχνικές τύπου Penalty.

9.13.3 Αντιμετώπιση περιορισμών με MCL

Παρακάτω δίνουμε τη συνάρτηση funmin και πλήρες πρόγραμμα MCL για την επίλυση του προβλήματος:

Ελαχιστοποίηση της συνάρτησης:

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2$$

υπό τους περιορισμούς:

$$x_1 + x_2 + x_3 + x_4 = 1$$
$$-(x_1 + x_2) \le -2$$

```
      function funmin(x,n)
      implicit double precision (a-h,o-z)
      dimension x(n)
*
      call getfla(1,r)
*
      sum = -1
      funmin = 0
*
      do 1 i = 1,n
        sum = sum + x(i)
        funmin = funmin + x(i)**2
1     continue
*
      funmin = funmin + r*(sum**2 + min(0.d0,x(1)+x(2)-2.d0)**2)
*
      end
```

```
program
var i

point(x.1=2.1;x.2=-99.1;x.3=0.001;x.4=-77.0001)
loop i from 0 to 10
   flag(f.1=10.**i)
   evaluate
   bfgs(noc=1000; print=0)
endloop
shortdis
end
```

Κεφάλαιο 10

Η γλώσσα προγραμματισμού MCL

10.1 Τι είναι η γλώσσα MCL

Όπως έχει ήδη αναφερθεί ο κύριος τρόπος χρήσης του Merlin είναι διαλογικός. Ο χρήστης εισάγει τις εντολές του από κάποιο τερματικό και παρατηρεί το αποτέλεσμα στην οθόνη. Γιά παράδειγμα, ο χρήστης μπορεί να διαπιστώσει ότι κάποιος ελαχιστοποιητής υπερτερεί έναντι των άλλων και να αποφασίσει να τον χρησιμοποιεί συνεχώς, ή (αφού οι αλγόριθμοι ελαχιστοποίησης ελέγχονται από παραμέτρους) να βρεί τις τιμές των παραμέτρων ελέγχου που δίνουν τα καλύτερα αποτελέσματα. Η διαδικασία αυτή είναι χρονοβόρα διότι συνήθως απαιτείται αρκετός πειραματισμός, κάτι που δεν είναι επιθυμητό.

Η γλώσσα προγραμματισμού MCL (Merlin Control Language) έχει φτιαχτεί κυρίως για να αντιμετωπίσει αυτό το πρόβλημα. Είναι μία κατανοητή καί εύκολη στη μάθηση γλώσσα που ελέγχει το Merlin δυναμικά, όπως και ο χρήστης από το τερματικό του, όμως δεν απαιτεί την ανάμιξη ή την παρέμβαση του χρήστη. Η MCL υποστηρίζει αριθμητικές πράξεις και συναρτήσεις (όπως η Fortran), περιέχει εντολές ελέγχου και επανάληψης, πολυδιάστατους πίνακες, εντολές εισόδου εξόδου, κα. Η δομή της γλώσσας περιγράφεται πλήρως στο [2].

10.2 Ο μεταγλωτιστής MCL

Για είναι σε θέση το Merlin να εκτελέσει ένα πρόγραμμα MCL, αυτό πρέπει να μεταγλωτιστεί σε μιά σειρά εντολών που το Merlin αναγνωρίζει και ονομάζονται "αντικειμενικός κώδικας Merlin" (Merlin Object Code). Τη λειτουργία αυτή αναλαμβάνει ο μεταγλωτιστής MCL. Ο μεταγλωτιστής MCL, αναμένει γιά είσοδο μιά σειρά από παραμέτρους στη μορφή:

```
παράμετρος=τιμή, παράμετρος=τιμή ...
```

Τυπικό παράδειγμα χρήσης του μεταγλωτιστή φαίνεται στον πίνακα 33.

Οι παράμετροι που δέχεται ο μεταγλωτιστής είναι:

`I` Το όνομα του αρχείου με το πρόγραμμα MCL που πρόκειται να μεταγλωτιστεί.

`B` Το όνομα του αρχείου που θα δεχτεί τον αντικειμενικό κώδικα Merlin. Η προκαθορισμένη τιμή είναι MOC.

`E` Το όνομα του αρχείου που θα δεχτεί τον κατάλογο με τα λάθη που πιθανώς θα βρεί ο μεταγλωτιστής στο πρόγραμμα MCL. Εάν δεν εισάγουμε τιμή για την παράμετρο E, τα λάθη θα εμφανιστούν στην οθόνη.

DEBUG Καθορίζει αν στον αντικειμενικό κώδικα Merlin θα συμπεριληφθούν πληροφορίες εκσφαλμάτωσης. Επί του παρόντος η μόνη πληροφορία εκσφαλμάτωσης που μπορεί να συμπεριληφθεί στον αντικειμενικό κώδικα Merlin (και που εμφανίζεται αν προκύψει κάποιο λάθος κατά την εκτέλεση του από το Merlin) είναι οι αριθμοί γραμμών του πηγαίου κώδικα MCL. Οι δύο επιτρεπτές τιμές για την παράμετρο αυτή είναι:

- TRUE Συμπεριλαμβάνονται οδηγίες εκσφαλμάτωσης στον αντικειμενικό κώδικα Merlin.
- FALSE Δεν συμπεριλαμβάνονται οδηγίες εκσφαλμάτωσης. Αυτή είναι η προκαθορισμένη τιμή.

BOUNDS Καθορίζει αν στον παραγόμενο αντικειμενικό κώδικα Merlin θα συμπεριληφθούν επιπλέον εντολές που θα ελέγχουν την ώρα της εκτέλεσης, αν οι δείκτες των πινάκων είναι μέσα στα δηλωμένα όρια. Επιτρεπτές τιμές για αυτή την παράμετρο είναι:

- TRUE Συμπεριλαμβάνεται κώδικας γιά τον έλεγχο των δεικτών των πινάκων.
- FALSE Δεν συμπεριλαμβάνεται κώδικας γιά τον έλεγχο των δεικτών των πινάκων. Αυτή είναι η προκαθορισμένη τιμή.

Ο μεταγλωτιστής MCL είναι γραμμένος σε ANSI-Fortran-77. Πρόγραμμα MCL και τμήμα του παραγόμενου αντικειμενικού κώδικα Merlin φαίνεται στον πίνακα 34. Ο αντικειμενικός κώδικας Merlin και η λειτουργία του μεταγλωτιστή MCL περιγράφεται στο εγχειρίδιο [23] που συνοδεύει την [2].

Πίνακας 33: Τυπικό παράδειγμα χρήσης του μεταγλωτιστή MCL.

```
-------------------------------------------------------------------------------
                              M C L - 1.0  LC
-------------------------------------------------------------------------------
  Enter Compilation parameters   - I, B, E, DEBUG, BOUNDS -
i=test  <cr>
Complilation succesful, output is in file "MOC".
```

Πίνακας 34: Πρόγραμμα MCL και τμήμα του παραγόμενου αντικειμενικού κώδικα Merlin.

```
var i; a                                $LENGTH
                                          7
a = 2                                   $PUSHC
                                          2
loop i from 1 to dim                    $PUSHC
    point (x.i=a)                           2.00000
end loop                                $POPCONT
                                        $PUSHC
shortdis                                    1.00000
```

10.3 Εκτέλεση προγραμμάτων MCL
Η εντολή RUNMCL

Προγράμματα MCL που έχουν μεταφραστεί σε αντικειμενικό κώδικα Merlin, εκτελούνται με χρήση της εντολής RUNMCL. Δείγμα χρήσης της RUNMCL και αποτελέσματα από πρόγραμμα MCL φαίνονται στον πίνακα 35.

Πίνακας 35: Αποτελέσματα από την εκτέλεση του προγράμματος MCL του πίνακα 34, με χρήση της εντολής RUNMCL.

```
    /\/\/\/\/\     Merlin is at your command !!!
runmcl  <cr>
RUNMCL

Enter file name:
MOC  <cr>
POINT
POINT
SHORTDIS

-------------------------------------------------------------------------------
Number of evaluations:    Function        Gradient        Hessian
Total:                           3               0              0
Since last reset:                3               0              0
-------------------------------------------------------------------------------
Index  Name       Fix       Parameter value       Left margin   Right margin
    1)            .......   2.00000000000000
    2)            .......   2.00000000000000
       Value .......     401.000000000000

    /\/\/\/\/\     Merlin is at your command !!!
```

Κεφάλαιο 11

Διαμόρφωση του συστήματος Το αρχείο CONFIG

Το Merlin είναι γραμμένο σε ANSI-Fortran-77 και δεν περιέχει ενσωματωμένα χαρακτηριστικά που να εξαρτώνται από κάποιο συγκεκριμένο υπολογιστή. Όμως την πρώτη φορά που το Merlin εγκαθίσταται σε ένα σύστημα ορισμένες παράμετροι πρέπει να καθοριστούν ορθά. Τέτοιες παράμετροι είναι οι αριθμοί των μονάδων εισόδου/εξόδου, ο μεγαλύτερος πραγματικός αριθμός που μπορεί να διαπραγματευτεί το σύστημα, κα. Ο μηχανισμός αλλαγής αυτών των παραμέτρων έχει ως εξής:

Όταν το Merlin ξεκινά, ελέγχει γιά την ύπαρξη ενός αρχείου με το όνομα `CONFIG`. Εάν το αρχείο βρεθεί, τότε αυτό ανοίγεται και διαβάζεται γραμμή προς γραμμή. Κάθε γραμμή του `CONFIG` περιέχει μία οδηγία διαμόρφωσης που αλλάζει την προκαθορισμένη συμπεριφορά του Merlin. Οι οδηγίες διαμόρφωσης, η δράση τους και οι αντίστοιχες προκαθορισμένες τιμές περιγράφονται παρακάτω. Στην περιγραφή που ακολουθεί το <x> αναφέρεται σε ένα πραγματικό αριθμό, τα <n> και <m> σε ακέραιο, ενώ το <filename> σε όνομα αρχείου.

11.1 BIGGER

Σύνταξη:	`BIGGER <x>`
Σκοπός:	Καθορίζει (κατά απόλυτη τιμή) τον μέγιστο πραγματικό αριθμό που μπορεί να χειριστεί το σύστημα.
Προκαθορισμένη τιμή:	1.E+38
Σχόλια:	Η προκαθορισμένη τιμή είναι κατάλληλη για αριθμούς απλής ακρίβειας για τους περισσότερους μεταγλωτιστές.

11.2 SMALLER

Σύνταξη:	`SMALLER <x>`
Σκοπός:	Καθορίζει (κατά απόλυτη τιμή) τον μικρότερο πραγματικό αριθμό που μπορεί να χειριστεί το σύστημα χωρίς να προκύψει κατάσταση underflow.
Προκαθορισμένη τιμή:	1.E-36
Σχόλια:	Η προκαθορισμένη τιμή είναι κατάλληλη για αριθμούς απλής ακρίβειας για τους περισσότερους μεταγλωτιστές.

11.3 INPUT_UNIT

Σύνταξη: `INPUT_UNIT <n>`

Σκοπός: Καθορίζει τον αριθμό που θα χρησιμοποιηθεί σαν αριθμός μονάδος εισόδου στις εντολές εισόδου Fortran.

Προκαθορισμένη τιμή: 5

Σχόλια: Ο αριθμός αυτός πρέπει να τεθεί έτσι ώστε εντολές read από την αντίστοιχη μονάδα <n> να αναφέρονται στο πληκτρολόγιο. Ο αριθμός αυτός δεν χρησιμοποιείται όταν καθοριστεί αρχείο εισόδου με την οδηγία διαμόρφωσης INPUT_FILE.

11.4 OUTPUT_UNIT

Σύνταξη: `OUTPUT_UNIT <n>`

Σκοπός: Καθορίζει τον αριθμό που θα χρησιμοποιηθεί σαν αριθμός μονάδος εξόδου στις εντολές εξόδου Fortran.

Προκαθορισμένη τιμή: 6

Σχόλια: Ο αριθμός αυτός πρέπει να τεθεί έτσι ώστε εντολές write στην αντίστοιχη μονάδα <n> να αναφέρονται στην οθόνη. Ο αριθμός αυτός δεν χρησιμοποιείται όταν καθοριστεί αρχείο εξόδου με την οδηγία διαμόρφωσης `OUTPUT_FILE`.

11.5 INPUT_FILE

Σύνταξη: `INPUT_FILE <filename>`

Σκοπός: Καθορίζει το όνομα του αρχείου από όπου το Merlin θα διαβάσει την είσοδο.

Προκαθορισμένη τιμή: Καμία. Η είσοδος διαβάζεται από την μονάδα που καθορίστηκε με την οδηγία διαμόρφωσης `INPUT_UNIT`. Η μονάδα αυτή πρέπει να αντιστοιχεί στο πληκτρολόγιο.

11.6 OUTPUT_FILE

Σύνταξη: `OUTPUT_FILE <filename>`

Σκοπός: Καθορίζει το όνομα του αρχείου όπου το Merlin θα γράψει την έξοδο.

Προκαθορισμένη τιμή: Καμία. Η έξοδος γράφεται στην μονάδα που καθορίστηκε με την οδηγία διαμόρφωσης `OUTPUT_UNIT`. Η μονάδα αυτή πρέπει να αντιστοιχεί στην οθόνη.

11.7 INPUT_PRECONN

Σύνταξη: `INPUT_PRECONN yes ή no`

Σκοπός: Καθορίζει αν η είσοδος είναι προσυνδεμένη από το λειτουργικό σύστημα στην αντίστοιχη μονάδα εισόδου. Σαν αποτέλεσμα το Merlin δεν θα χρησιμοποιήσει εντολές open ή close γιά αυτή τη μονάδα. Αν όμως έχει καθοριστεί το όνομα κάποιου αρχείου για είσοδο ή έξοδο (με την οδηγία διαμόρφωσης `INPUT_FILE`), το οποίο δεν είναι ήδη ανοικτό κατά την κλήση του Merlin, τότε το Merlin θα προσδιορίσει ένα κατάλληλο αριθμό μονάδος (διαφορετικό από αυτό που δίνεται στην οδηγία `INPUT_UNIT`) και θα ανοίξει το αρχείο. Όταν το Merlin τελειώσει, θα κλείσει το αρχείο, αφήνοντάς το στην αρχική κατάσταση.

Προκαθορισμένη τιμή: Η είσοδος είναι προσυνδεμένη από το λειτουργικό σύστημα στην μονάδα εισόδου που καθορίζεται από την οδηγία διαμόρφωσης `INPUT_UNIT`.

11.8 OUTPUT_PRECONN

Σύνταξη: `OUTPUT_PRECONN yes ή no`

Σκοπός: Καθορίζει αν η έξοδος είναι προσυνδεμένη από το λειτουργικό σύστημα στην αντίστοιχη μονάδα εξόδου. Σαν αποτέλεσμα το Merlin δεν θα χρησιμοποιήσει εντολές open ή close γιά αυτή τη μονάδα. Αν όμως έχει καθοριστεί το όνομα κάποιου αρχείου για έξοδο (με την οδηγία διαμόρφωσης `OUTPUT_FILE`), το οποίο δεν είναι ήδη ανοικτό κατά την κλήση του Merlin, τότε το Merlin θα προσδιορίσει ένα κατάλληλο αριθμό μονάδος (διαφορετικό από αυτό που δίνεται στην οδηγία `OUTPUT_UNIT`) και θα ανοίξει το αρχείο. Όταν το Merlin τελειώσει, θα κλείσει το αρχείο, αφήνοντάς το στην αρχική κατάσταση.

Προκαθορισμένη τιμή: Η έξοδος είναι προσυνδεμένη από το λειτουργικό σύστημα στην μονάδα εισόδου που καθορίζεται από την οδηγία διαμόρφωσης `OUTPUT_UNIT`.

11.9 UNIT_RANGE

Σύνταξη: `UNIT_RANGE <n> <m>`

Σκοπός: Καθορίζει την περιοχή των αριθμών μονάδων που επιτρέπεται να χρησιμοποιήσει το Merlin για εσωτερικές διαδικασίες εισόδου/εξόδου.

Προκαθορισμένη τιμή: 50 80

Σχόλια: Αν ένας αριθμός στην ανωτέρω περιοχή έχει ανατεθεί σε κάποιο αρχείο που είναι ήδη ανοικτό κατά την κλήση του υποπρογράμματος Merlin, ο αριθμός αυτός δεν θα χρησιμοποιηθεί.

11.10 HEADER

Σύνταξη: `HEADER yes ή no`

Σκοπός: Καθορίζει αν θα εμφανιστούν οι αρχικές πληροφορίες που δίνονται κατά την εκκίνηση του Merlin (έκδοση, e-mail κλπ).

Προκαθορισμένη τιμή: Οι πληροφορίες εμφανίζονται.

11.11 PRINTOUT

Σύνταξη: `PRINTOUT <print-option>`

Σκοπός: Καθορίζει το αρχικό επίπεδο εξόδου από το πρόγραμμα. Επιτρεπτές τιμές για το <print-option> είναι:

- `FULL` Παράγεται η μέγιστη δυνατή έξοδος με πλήθος διαγνωστικών μηνυμάτων και προτροπών.
- `HALF` Παράγεται μερική έξοδος με τα πιό σημαντικά διαγνωστικά μηνύματα και προτροπές.
- `NO` Δεν παράγεται καθόλου έξοδος.

Προκαθορισμένη τιμή: `FULL`

Σχόλια: Βλέπε επίσης και την παράγραφο 5.4.

11.12 PROLOG

Σύνταξη: `PROLOG <command>`

Σκοπός: Καθορίζει ότι η <command> (που υποτίθεται ότι είναι μία σωστή εντολή Merlin ή κλήση εντολοπακέτου), θα εκτελεστεί μόλις το Merlin ξεκινήσει.

Προκαθορισμένη τιμή: Καμία. Δεν εκτελείται πρόλογος.

11.13 EPILOG

Σύνταξη: `EPILOG <command>`

Σκοπός: Καθορίζει ότι η <command> (που υποτίθεται ότι είναι μία σωστή εντολή Merlin ή κλήση εντολοπακέτου), θα εκτελεστεί μόλις δοθεί στο Merlin μία από τις εντολές τερματισμού `STOP`, `RETURN` ή `QUIT` και πριν ο έλεγχος επιστρέψει στο καλών πρόγραμμα. Αν κάποια από αυτές τις εντολές δοθεί με μία οποιαδήποτε παράμετρο (π.χ. `STOP XX`), η <command> δεν θα εκτελεστεί.

Προκαθορισμένη τιμή: Καμία. Δεν εκτελείται επίλογος.

Σχόλια: Αν στη θέση της <command> δοθεί η κλήση ενός εντολοπακέτου (π.χ. `EPILOG .ENDIT`), είναι δυνατόν να δοθεί μία δεύτερη εντολή τερματισμού από το εντολοπακέτο. Σε αυτή την περίπτωση η <command> δεν εκτελείται ξανά, αλλά το Merlin τερματίζει τη λειτουργία του ή επιστρέφει στο καλών πρόγραμμα (ανάλογα με την αντίστοιχη εντολή τερματισμού) αμέσως.

11.14 MODE

Σύνταξη: `MODE <mode>`

Σκοπός: Καθορίζει ότι η λειτουργία του πακέτου θα γίνει με τον τρόπο <mode>. Επιτρεπτές τιμες είναι:

- `IAF` Διαλογικός τρόπος χρήσης. Οποιοδήποτε λάθος δεν τερματίζει το πρόγραμμα, αλλά επιστρέφει τον έλεγχο στον χρήστη γιά την εισαγωγή μιας νέας εντολής.
- `BATCH` Μη διαλογικός τρόπος χρήσης. Οποιοδήποτε λάθος τερματίζει τη λειτουργία του προγράμματος.

Προκαθορισμένη τιμή: `IAF`

Σχόλια: Εάν στο αρχείο `CONFIG` δεν υπάρχει η οδηγία διαμόρφωσης `MODE`, τότε κατά την εκκίνηση του Merlin, ο χρήστης θα ερωτηθεί διαλογικά για τον επιθυμητό τρόπο χρήσης.

11.15 FILE

Σύνταξη: `FILE <filename>`

Σκοπός: Τα περιεχόμενα του αρχείου `CONFIG`, από την επόμενη γραμμή και έως μιά γραμμή που θα περιέχει τους χαρακτήρες `...END` αντιγράφονται κατά την εκκίνηση του Merlin στο αρχείο <filename>.

Προκαθορισμένη τιμή: Καμία.

11.16 PLUG

Σύνταξη: `PLUG <n> <name>`

Σκοπός: Καθορίζει ότι στην προσαρτούμενη εντολή <n>, που αντιστοιχεί στην `SUBROUTINE PLUG<n>`, έχει προστεθεί κώδικας από το χρήστη, ο οποίος στο εξής θα είναι γνωστός με το όνομα `<name>`.

Προκαθορισμένη τιμή: Καμία.

11.17 Σχόλια

Στο αρχείο `CONFIG` μπορούν να εισαχθούν επεξηγηματικά σχόλια. Αυτά εισάγονται με τη χρήση του συμβόλου % σαν πρώτου χαρακτήρα μιάς γραμμής.

Κεφάλαιο 12

Επεκτάσεις

12.1 Μέγιστος αριθμός παραμέτρων και όρων

Ο μέγιστος αριθμός παραμέτρων της αντικειμενικής συνάρτησης καθορίζεται στον πηγαίο κώδικα Fortran με τη δήλωση:

```
PARAMETER ( MXV = 200 )
```

ενώ ο μέγιστος αριθμός όρων όταν η συνάρτηση είναι άθροισμα τετραγώνων καθορίζεται από τη δήλωση:

```
PARAMETER ( MXT = 200 )
```

Οι παραπάνω δηλώσεις εμφανίζονται σε διάφορα σημεία του πηγαίου κώδικα και χρησιμοποιούνται για τον ορισμό βοηθητικών μονοδιάστατων πινάκων.

12.2 Προσαρτούμενες εντολές

Είναι πολλές φορές επιθυμητό ο χρήστης να ενσωματώσει δικά του τμήματα κώδικα στο πακέτο, τα οποία θα καλούνται όπως οι υπόλοιπες εντολές του Merlin. Για το σκοπό αυτό χρησιμοποιείται ένας συγκεκριμένος μηχανισμός, που επιτρέπει στο χρήστη να ενσωματώσει ομοιόμορφα δικά του τμήματα κώδικα, χωρίς να χρειάζεται να γνωρίζει τις εσωτερικές λεπτομέρειες του πακέτου και κυρίως χωρίς να χρειάζεται να επέμβει στον πηγαίο κώδικα. Επεκτάσεις που προστίθενται με τον τρόπο αυτό ονομάζονται "*προσαρτούμενες εντολές*" και αναγνωρίζονται αυτόματα από την MCL. Η διαδικασία ενσωμάτωσης μιας προσαρτούμενης εντολής έχει ως εξής:

- Ο χρήστης πρέπει να προγραμματίσει ένα υποπρόγραμμα που θα διεκπεραιώνει την επιθυμητή λειτουργία.
- Το αρχείο διαμόρφωσης `CONFIG` θα πρέπει να ενημερωθεί κατάλληλα έτσι ώστε να αναγνωρίζεται η νέα εντολή.

12.3 Γράφοντας την προσαρτούμενη εντολή

Ας υποθέσουμε ότι θέλουμε να δημιουργήσουμε μια εναλλακτική μορφή της εντολής `SHORTDIS`. Δηλαδή μια προσαρτούμενη εντολή που θα τυπώνει τις τιμές των παραμέτρων, τα συμβολικά ονόματά τους (εάν υπάρχουν) και τις πρώτες παραγώγους της συνάρτησης. Ένα υποπρόγραμμα που εκτελεί τις λειτουργίες αυτές φαίνεται στον πίνακα 36.

Όπως είναι σαφές από το υπόδειγμα, για να χρησιμοποιήσει κανείς τις εσωτερικές μεταβλητές του Merlin (τιμές των παραμέτρων και παραγώγων, μονάδες εισόδου - εξόδου κλπ.) πρέπει να χρησιμοποιήσει ένα καθορισμένο σύνολο *υποπρογραμμάτων προσαρμογής*. Πλήρης κατάλογος των υποπρογραμμάτων προσαρμογής δίνεται στο Παράρτημα Α.

Τέλος πρέπει να εισαχθεί μια κλήση στο υποπρόγραμμα του χρήστη σε ένα από τα 50 κενά υποπρογράμματα (`PLUG1` έως `PLUG50`) που διατίθενται για το σκοπό αυτό.

```
SUBROUTINE PLUG1
CALL LIKESH
END
```

Πίνακας 36: Υπόδειγμα προσαρτούμενης εντολής.

```
       SUBROUTINE LIKESH
       CHARACTER*10 NAME
       CALL GETDIM(N,M)
       CALL GETIOU(IUINP,IUOUT)
       DO 10,I=1,N
           CALL GETX1(I,XI)
           CALL GETNM1(I,NAME)
           CALL GETG1(I,GI)
           WRITE (IUOUT,20) I, NAME, XI, GI
10     CONTINUE
20     FORMAT (2X,I5,2X,A,2X,1PE14.7,2X,1PE14.7)
       END
```

12.4 Ενημέρωση του αρχείου CONFIG

Προκειμένου να αναγνωρίζονται αυτόματα οι προσαρτούμενες εντολές, πρέπει να τους δοθεί ένα όνομα, το οποίο θα δηλωθεί στο αρχείο διαμόρφωσης `CONFIG`. Για το παράδειγμα της προηγούμενης παραγράφου η αντίστοιχη οδηγία διαμόρφωσης θα μπορύσε να είναι:

```
PLUG 1 NEWSH
```

Βιβλιογραφία

1. G.A. Evangelakis, J.P. Rizos, I.E. Lagaris and I.N. Demetropoulos, Merlin - A Portable System for Multidimensional Minimization, Comp. Phys. Commun. 46 (1987) 401-415

2. C.S. Chassapis, D.G. Papageorgiou and I.E. Lagaris, MCL - Optimization Oriented Programming Language, Comp. Phys. Commun. 52 (1989) 223-239

3. D.G. Papageorgiou, C.S. Chassapis and I.E. Lagaris, Merlin-2.0 - Enhanced and Programmable Version, Comp. Phys. Commun. 52 (1989) 241-247

4. D.G. Papageorgiou and I.E. Lagaris, Merlin-2.1 - Double Precision, Comp. Phys. Commun. 58 (1990) 119-125

5. P.E. Gill, W. Murray and M.H. Wright, Practical Optimization, Academic Press, 1989

6. R. Fletcher, Practical Methodes of Optimization, Volume 1, Unconstrained Optimization, John Wiley, 1980

7. J.E Dennis jr. and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice Hall, 1983

8. E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, 1971

9. M.A. Wolfe, Numerical Methods for Unconstrained Optimization, Van Nostrand Reinhold Company Ltd., 1987

10. Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute, New York

11. J.J More, B.S. Garbow and K.E. Hillstrom, Testing Unconstrainded Optimization Software, ACM Trans. Math. Softw. 7, (1981) 17-41

12. A Fortran-to-C Converter, S.I. Feldman, D.M. Gray, M.W. Maimone, N.L. Schryer, Computing Science Technical Report No. 149, AT&T Bell Laboratories

13. H.H Rosenbrock, An Automatic Method for Finding the Greatest or Least Value of a Function, Computer J. 3 (1960) 175-184

14. C.G. Broyden, The Convergence of a Class of Double Rank Minimization Algorithms, parts I and II, J. Inst. Maths. Applns., 6 (1970) 76-90 and 222-231
R. Fletcher, A New Approach to Variable Metric Algorithms, Computer J., 13 (1970) 317-322
D. Goldfarb, A Family of Variable Metric Methods Derived by Variational Means, Maths. Comput. 24 (1970) 23-26
D. Shanno, Conditioning of Quasi-Newton Methods for Function Minimization, Maths Comput., 24 (1970) 647-656

15. W.C. Davidon, Variable Metric Method for Minimization, AEC Res. and Dev. Report ANL-5990 (revised), 1959
R. Fletcher and M.J.D. Powell, A Rapidly Convergent Descent Method for Minimization, Computer J. 6 (1963) 163-168

16. R. Fletcher and C.M. Reeves, Function Minimization by Conjugate Gradients, Comp. J., 7 (1964) 149-154

17. J. Nedler and R. Mead, Computer J. 7 (1965) 308

18. K. Mika and Th. Chaves, Bericte der KFA, Julich No 1643 (1980)

19. M.J.D. Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming, 45, 1989,547-566

20. M.J.D. Powell, TOLMIN: A Fortran Package for Linearly Constrained Optimization Calculations, DAMTP 1989/NA2, Dept. of Applied Mathematics and Theoretical Physics, Univ. of Cambridge, June 1989.

21. M.D. Hebden, An Algorithm for Minimization Using Exact Second Derivatives, Atomic Energy Research Establishment report TP 515, Harwell, England.
J.J. More, the Levenberg-Marquardt algoritm: Implementation and Theory, Numerical Analysis, G.A. Watson ed., Lecture Notes in mathematics 630, Springer-Verlag.

22. J.J. More, B.S. Garbow, K.E. Hillstrom, User Guide for Minpack-1, ANL-80-74, August 1980

23. C.S. Chassapis, D.G. Papageorgiou and I.E. Lagaris, MCL and MCLCOM 1.0 Developers Manual, Comp. Phys. Commun. program ABHA (1989)

24. Al-Baali M. and Fletcher R., An efficient Line Search for nonlinear Least Squares, JOTA 48 (1986) 359-377.

Παράρτημα Α

Υποπρογράμματα προσαρμογής

```
SUBROUTINE GETDIM ( NOVAR, NOTERM )
INTEGER NOVAR, NOTERM
```
Επιστρέφει τον αριθμό των παραμέτρων (NOVAR) της αντικειμενικής συνάρτησης. Στην περίπτωση που η συνάρτηση έχει τη μορφή αθροίσματος τετραγώνων, επιστρέφεται και ο αριθμός των επιμέρους όρων (NOTERM).

```
SUBROUTINE GETX ( X )
REAL X(*)
```
Επιστρέφει τις τρέχουσες τιμές των παραμέτρων x_i. Το στοιχείο X(I) αντιστοιχεί στην παράμετρο x_i.

```
SUBROUTINE GETFIX ( IX )
INTEGER IX(*)
```
Επιστρέφει την κατάσταση σταθεροποίησης των παραμέτρων x_i. Το στοιχείο IX(I) αντιστοιχεί στην παράμετρο x_i και η επιστρεφόμενη τιμή είναι 0, αν η παράμετρος x_i έχει σταθεροποιηθεί. Αλλιώς επιστρέφεται η τιμή 1.

```
SUBROUTINE GETMAR ( XL, XU, MF )
REAL XL(*), XU(*)
INTEGER MF(*)
```
Επιστρέφει τις τιμές των άνω (XU(I)) και κάτω (XL(I)) ορίων γιά κάθε παράμετρο x_i. Τα στοιχεία του πίνακα MF δείχνουν ποιά όρια έχουν τεθεί. Συγκεκριμένα MF(I)=0, εάν δεν έχουν τεθεί όρια στην παράμετρο x_i. MF(I)=−1, εάν έχει τεθεί μόνο το κάτω όριο. MF(I)=1, εάν έχει τεθεί μόνο το άνω όριο. MF(I)=2, εάν έχουν τεθεί άνω και κάτω όριο.

```
SUBROUTINE GETACC ( ACC )
REAL ACC
```
Επιστρέφει τη σχετική ακρίβεια του υπολογιστή.

```
SUBROUTINE GETFLA ( I, F )
```
Επιστρέφει την τιμή (F) της flag που ο δείκτης της καθορίζεται από τη μεταβλητή I.

```
SUBROUTINE SETFLA ( I, F )
```
Θέτει την τιμή F στην flag που ο δείκτης της καθορίζεται από τη μεταβλητή I.

```
INTEGER FUNCTION ISMCL ( )
```
Επιστρέφει την τιμή 1 εάν εκτελείται πρόγραμμα MCL, διαφορετικά επιστρέφει 0.

```
SUBROUTINE GETX1 ( I, X )
```
Επιστρέφει την τιμή (X) της παραμέτρου που ο δείκτης της καθορίζεται από τη μεταβλητή I.

```
SUBROUTINE SETX1 ( I, X )
```
Θέτει την τιμή X στην παράμετρο που ο δείκτης της καθορίζεται από τη μεταβλητή I.

```
SUBROUTINE GETFX1 ( I, IX )
```
Επιστρέφει την κατάσταση σταθεροποίησης (IX) για τη παράμετρο που ο δείκτης της καθορίζεται από τη μεταβλητή I.

```
SUBROUTINE GETMAR1 ( I, XL, XU, MF )
```
Επιστρέφει τα άνω (XU) και κάτω (XL) όρια για την παράμετρο που ο δείκτης της καθορίζεται από τη μεταβλητή I. Η τιμές της μεταβλητής MF είναι όπως για την SUBROUTINE GETMAR.

```
SUBROUTINE GETVAL ( VALUE )
```
Επιστρέφει την τρέχουσα τιμή της αντικειμενικής συνάρτησης.

```
SUBROUTINE GETNAM ( NAME )
CHARACTER*(*) NAME(*)
```
Επιστρέφει στον πίνακα NAME τα συμβολικά ονόματα των παραμέτρων, όπως αυτά έχουν τεθεί με την εντολή GODFATHER.

```
INTEGER FUNCTION ISIAF ( )
```
Επιστρέφει την τιμή 1 εάν έχει επιλεγεί ο διαλογικός τρόπος επικοινωνίας (IAF). Διαφορετικά επιστρέφει την τιμή 0.

```
SUBROUTINE GETTER ( TERM )
REAL TERM(*)
```
Εάν η συνάρτηση είναι της μορφής αθροίσματος τετραγώνων, επιστρέφονται οι τιμές των επιμέρους όρων.

```
SUBROUTINE GETIOU ( NIN, NOUT )
INTEGER NIN, NOUT
```
Επιστρέφει τις μονάδες εισόδου (NIN) και εξόδου (NOUT) που χρησιμοποιεί το Merlin.

```
SUBROUTINE GETG ( GRAD )
REAL GRAD(*)
```
Επιστρέφει τις πρώτες παραγώγους της αντικειμενικής συνάρτησης, υπολογισμένες με τον τρόπο που έχει καθοριστεί (δες παράγραφο 4.1).

```
SUBROUTINE GETG1 ( I, G )
INTEGER I
REAL G
```
Επιστρέφει την πρώτη παράγωγο της αντικειμενικής συνάρτησης ως προς τη μεταβλητή `I`.