Computer Physics Communications 46 (1987) 401–415
North-Holland, Amsterdam

MERLIN – A PORTABLE SYSTEM FOR MULTIDIMENSIONAL MINIMIZATION

G.A. EVANGELAKIS, J.P. RIZOS, I.E. LAGARIS
Physics Department, University of Ioannina, Ioannina, Greece

and

I.N. DEMETROPOULOS
Department of Chemistry, University of Ioannina, Ioannina, Greece

Received 5 January 1987; in revised form 5 May 1987

PROGRAM SUMMARY

Title of the program: MERLIN

Catalogue number: AAXW

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: CDC CYBER-171; *Installation*: University of Ioannina, Ioannina, Greece

Operating system: NOS 2.4.1 630/628

Programming language used: ANSI FORTRAN 77

High speed storage required: depending upon the maximum number of variables the object can handle (28000 words for 30 variables, 50000 words for 150 variables)

Number of bits in a word: 60

Peripherals used: terminal, line printer, card reader, disc

Number of lines in combined program and test deck: 4370

Separate documentation available: manual (54 pages)

Keywords: minimization, data fitting, SIMPLEX, DFP, BFGS, variable metric, stochastic search, line search, numerical differentiation

Nature of physical problem
A lot of problems in physics, chemistry, applied mathematics as well as in engineering and in other fields, are quite often reduced to minimizing a function of several variables. MERLIN is a system designed to minimize a multidimensional function.

Method of solution
Six algorithms are implemented. Three of them make use of the function's gradient and hence are suitable for minimizing differentiable functions, the rest three do not use derivatives at all and so are applicable to non-differentiable functions as well.
The first three are the quasi-Newton methods known as DFP [1], BFGS [2] and the conjugate gradient method of Polak and Ribiere [3]. The SIMPLEX method of Nedler and Mead [4], corrected as suggested by Mike and Chaves [5], a random search and an ad hoc method, which use only function values, are implemented as well.

Restriction of the complexity of the problem
Currently MERLIN is dimensioned to handle up to 150 variables. However by redimensioning a few arrays it can easily be enhanced or reduced according to user's needs, as described in the provided manual.

Typical running time
Since housekeeping operations are quite fast, running time heavily depends on the complexity of the objective function. The provided test run, took 7.12 CPU seconds on a CDC CYBER-171.

Unusual features of the problem
Apart from being a minimization program, MERLIN is designed to be easily further developed by the user and evolve in different directions. Hence it can serve as a testing ground of different trial algorithms and can be seen as an optimization development system.

References
[1] W.C. Davidon, AEC R&D report ANL 5990 (1959).
R. Fletcher and M.J.D. Powell, Computer J. 6 (1963) 163.

[2] R. Fletcher, Computer J. 13 (1970) 317.
D. Goldfarb, Maths. Comput. 24 (1970) 23.
C.G. Broyden, J. Inst. Maths. Appl. 6 (1970) 76 and 222.
D.F. Shanno, Maths. Comput. 24 (1970) 647.
[3] E. Polak, Computational Methods in Optimization: A Unified Approach (Academic Press, New York, 1971).
[4] J.A. Nedler and R. Mead, Computer J. 7 (1965) 308.
[5] K. Mika and Th. Chaves, Berichte der KFA Jülich Nr. 1643 (1980).

LONG WRITE-UP

1. Introduction

Minimizing a function of many variables is a problem common to many fields. A lot of problems in engineering, chemistry, physics etc. are quite often reduced to minimizing a function of several variables. As examples we refer to nuclear-interaction modeling by potentials [1], to computational fitting of ab initio potential energy surfaces [2], to curve fitting (an every-day need in experimental work), and to all kinds of variational methods that are applied either to classical or to quantum mechanical problems.

The difficulties of multidimensional minimization are many and diverse. The function may not be analytically known, so that derivatives have to be estimated numerically, which costs both in computing time and in accuracy. Problems also arise when many local minima exist, since there is no algorithm that guarantees convergence to the global solution. We do not know of any method that deals satisfactorily with this problem. When physical limits are imposed on some parameters, the problem becomes constrained and thus more difficult to handle.

MERLIN is a system constructed in an attempt to solve the above mentioned problems, aiming in addition to be user-friendly and easily portable to different machines. It has been under continuous development since 1983 and it has been benefited from our experience with the program MINUIT [3].

2. General description

The ease of use is achieved by means of an operating system residing in SUBROUTINE MERLIN. The user enters commands to instruct the program for the route of action. There is an on-line command directory to remind the user of the commands in case of uncertainty, and an on-line HELP that offers a brief description for each command.

The commands can be separated into categories corresponding to their action as: Minimization, Information, Administration, Editing and User commands.

Minimization commands instruct MERLIN to call the appropriate minimization routines.

Information commands give information about MERLIN's operating system.

Administration commands can be mode-commands, display commands or control commands.

Editing commands refer to MACROS. A MACRO is a package of commands that the user may create interactively. To modify a MACRO a simple editor is provided which is operated by the above mentioned editing commands. User commands are dummy commands. They do nothing. Simply transfer the control to a point in the program where a future call to an associated subroutine may appear and then return the control to the command prompting point. MERLIN recognizes them as valid commands i.e. they are built into MERLIN's operating architecture, in order to ease further development by the interested user.

MERLIN is portable in the sense that care has been taken to follow closely the FORTRAN 77 ANSI standards. Assembly routines are not used and open/close statements are made optional since many compilers treat them in different manner. There is only one exception. The random number generator function (used only in one place, in subroutine RANDOM) is chosen to be the CDC's intrinsic function RANF( ), that has to be replaced in order to run on other machines.

The rest of this paper is organized in the following way.

In section 3, MERLIN's operating system is

presented. Therein, a description of MERLIN-modes and all MERLIN commands is given.

In section 4, the minimization algorithms are sketched.

In section 5, the structure of the user supplied routines is presented, along with an example.

In section 6, some of the Merlin's special features are discussed.

3. MERLIN – operating system

3.1. MERLIN – files and associated units

MERLIN uses the following twelve units for I/O operations:

31, 32, 33, 34, 35, 36 and 41, 42, 43, 44, 45, 46.

Units 31 and 32 are the input and output files and when running interactively they should be assigned to the terminal by the appropriate system command. Unit 45 is a scratch file necessary for picking records of information by the command PICK described later on. Unit 42 is reserved for a HELP-file. Unit 41 is the unit where the MACROS are written and MERLIN assignes to it the name MACROF. Units 33, 34, 35, 43 and 44 are assigned the names: DATA, INIPO, BACKUP, STORE and DISPO. In these files reside one or more records, each with the following information:

The following line appears N times, where N is the number of variables of the objective function:

index, (name), fix status, value, (left margin), (right margin).

Quantities in parentheses may or may not appear. For example the user may assign a name to each variable in which case, names will appear, also margins may or may not appear depending upon whether the user wishes to frame or not, some of the variables in a certain range. For fixed variables an asterisk appears in between dots, in the opposite case only dots appear.

The last line of the record contains the function's value, evaluated at the point defined by the variable-values listed in the record. An example for a three-variable case of such a record is shown in table 1.

The DATA file contains only one record, while files: INIPO, BACKUP, STORE and DISPO may contain none or more than one records.

File BACKUP contains information about the current point and about previous points and is updated every time a minimization routine terminates by appending to it a record with the current point information.

File DATA is rewound and updated every time prior to the execution of a minimization routine, so that it always contains information about the previous and not about the current point.

File STORE contains information about points selected by the user and is updated by appending records to the end of it every time the command MEMO is issued.

File DISPO contains information about points which the user wishes to temporarily reject in order to try processing the minimization from a different point. It is updated by appending records to the end of it every time either of the commands PICK, POINT, INIT or ACCUM are issued. Units 36 and 46 are associated with the commands REWIND and INIT only when the optional OPEN statements are used.

All the above units are considered reserved by MERLIN.

3.2. MERLIN – operating modes

There are seven categories of behavior that are globally followed. These are called modes of operation. Each mode has its options. The modes and the associated options are shown in table 2. The first option-line contains the default options.

The BACKUP mode options, arrange the level of the protection of the intermediate results, which is implemented by updating the file BACKUP. For example the option FULLBACK, corre-

Table 1

1) WO	.....	50.000000000000	50.00	70.00
2) AS	.. * ..	87.000000000000	80.00	
3) RO	.....	6.6147969121794		
VALUE	.....	9913.4824639511		

Table 2
MERLIN modes and associated options

BACKUP	DERIVA	PRINTO	USERSO	CALLBY	PROCES	PANEL
FULLBACK	NUMER	FULLPRINT	EXPERT	INDEX	IAF	PANEL-ON
LASTBACK	ANAL	HALFPRINT	ROOKIE	NAME	BATCH	PANEL-OFF
NOBACK		NOPRINT				

sponds to updating that file without erasing earlier information, while the option LASTBACK, corresponds to backing-up only the latest results (earlier points are deleted) and the NOBACK option offers no protection at all.

Changing a mode option is done by issuing an appropriate command (see next section).

The DERIVA mode options instruct MERLIN how to calculate the gradient of the objective function. If the NUMER option is chosen, the calculation proceeds via a numerical algorithm (described in the program's manual), while if the ANAL option is chosen, MERLIN calls a user-supplied routine.

The PRINTO mode options arrange the level of MERLIN's output. The FULLPRINT option corresponds to the highest level (i.e. to the maximum output information), while the HALFPRINT option corresponds to an intermediate level (parts of the output are suppressed) and the NOPRINT option corresponds to totally suppressing the output information.

The USERSO mode options set the user sophistication level. In the EXPERT option, minimization proceeds through the minimization commands, while in the ROOKIE option there is only one minimization command (the command AUTO) which invokes an automatic minimizer to ease the operation for the inexperienced user.

The CALLBY mode options set the way of referencing the parameters of the objective function. In INDEX option, each parameter is referenced by its index, while in the NAME option by its name. To operate in the NAME option, the user must have previously assigned parameter names (see command GODFATHER).

The PROCES mode options set the way of error handling.

In IAF option MERLIN is tolerant to errors, issues informative messages and does not abort. This option is very convenient for interactive usage.

In BATCH option, upon an error in input, execution is terminated and a diagnostic message is issued. This option is to be selected for batch processing, where user intervention is not possible.

The PANEL mode options PANEL-ON, PANEL-OFF activate or deactivate the Panel-prompting (a way to input information described later).

Inspection of the current mode-options, is facilitated by the command MODEDIS.

3.3. MERLIN – operating commands

The description of the MERLIN commands, reveals the capabilities of the program and clarifies the possibilities offered to the user.

MERLIN's prompt for command entry is:
/\/\/\/\/\ MERLIN IS AT YOUR COMMAND!!!
ADMINISTRATION COMMANDS:
Mode commands:
FULLBACK, LASTBACK, NOBACK
These commands set the BACKUP mode options in an obvious correspondence.
ANAL, NUMER
These commands set the DERIVA mode options.
FULLPRINT, HALFPRINT, NOPRINT
These commands set the PRINTO mode options.
EXPERT, ROOKIE
These commands set the USERSO mode options.
INDEX, NAME
These commands set the CALLBY mode options.
IAF, BATCH
These commands set the PROCES mode options.
PANELOFF, PANELON
These commands set the PANEL mode options.

Display commands:
MODEDIS
Displays current mode options.
SHORTDIS
Displays a record of information for the current point and in addition the values of two counters. The first counter informs the user of the total number of evaluations of the objective function since the beginning of the run, the second, issues the number of calls to the objective function since the last time the RESET command was used.
GRADDIS
If in NUMER option, the user is prompt to enter an error-bound, required for the derivative calculation. The values of the first partial derivatives are then displayed along with their error codes. A zero error code indicates satisfactory derivative calculation. (Within the required accuracy). If in ANAL option, no further prompting takes place and no error codes are displayed.
GRAPH
Permits the user to obtain a one-dimensional rough graph of the objective function, versus a specified variable, in a specified range. A minor minimization is performed. Prompting is done via a panel.
GRADCHECK
Checks the values of the user-supplied derivatives against those calculated numerically by MERLIN. This command works only if the ANAL option is selected, if not, a reminder to the user is issued. The user is prompt to enter an error-bound for the numerical calculation of the gradient.
CATALOG
Lists the names of MERLIN files that the user can manipulate. An asterisk below the file's name indicates that the file is being used, otherwise the indication 'EMPTY' appears instead.

Control commands:
STOP
Terminates execution of the program.
RETURN
Terminates MERLIN minimization and transfers the control to the main program. If one uses the main program provided here, (Program MASTER) the STOP and RETURN commands are equivalent. However the user may call MERLIN from his own main program, work with it, and upon reaching at a satisfactory point to terminate the minimization session and transfer the control back to his main program for further processing.
QUIT
Prompts the user to enter a value for MERLIN's return flag output parameter IQUIT and returns control to the main program. Through this return-flag, additional control is offered to the user's main program.
GODFATHER
Assigns names to variables. This command when issued for the first time, invokes prompting and the user is expected to enter a name for the variable whose index is displayed. Prompting stops when all variables are assigned names or when a carriage return is entered immediately after the prompt.
If this command is used more than once, MERLIN assumes that the user needs only to modify one or more names and prompting stops after the first entry which may be a string of the form:
indexi,namei,indexj,namej,indexk,namek,...,indexq,nameq and can be 79 characters long at most.
MERLIN does no checks for duplicate names so the user should be careful while assigning them.
MARGIN
Initiates prompting for assigning left and right margins to variables. By that is understood that a variable must be less or equal to its right and greater or equal to its left margin. The first prompt asks for left margins the second prompt asks for right margins. The entry to both prompts may be a string of the form:
indexi,margini,indexj,marginj,...,indexk,margink and can be 79 characters long at most.
If the NAME option is selected for the CALLBY mode, indices are replaced by names.
DEMARGIN
Removes margins from specified variables. It first prompts for removal of left margins and consecutively for removal of right margins.
The input strings required are of the form:
index1,index2,index3,...,indexk
If in INDEX option and
name1,name2,name3,...,namek
if in NAME option of the CALLBY mode.
FIX

Fixes specified variables to their current values.
If option INDEX is selected for the CALLBY mode, the user is prompt to enter the indices of the variables to the fixed; otherwise if option NAME is selected, the user should enter the names of the variables instead.
The required input is as in the DEMARGIN command.

LOOSE

Looses previously fixed variables. Prompting is same as for the FIX command.

LOOSALL

Looses all fixed variables. No prompting.

STEPALL

Calculates search steps for all non-fixed variables. It invokes a step-size procedure that uses the gradient of the objective function. It also performs a minor minimization. No prompting.

ADJUST

Calculates search steps for all non-fixed variables. It invokes a step-size procedure that does not need the gradient components. Hence it is preferable when the gradient is either not provided by the user or when its calculation is time consuming. It also performs a minor minimization. No prompting.

STEP

Initiates prompting for assigning values to the search steps for each variable. The input string should be of the form:
indexi,stepi,indexj,stepj,...,indexk,stepk
(INDEX option)
namei,stepi,namej,stepj,...,namek,stepk
(NAME option)

POINT

Initiates prompting for assigning values to the variables. The entry is as in STEP command. Prompting stops if either a single carriage return is entered on prompt or a zero is entered in place of an index (or name). The previous point information is appended in file DISPO.

REWIND

Rewinds specified unit. Unit specification is prompt. If OPEN statements are used, a file-name is prompt associated with unit 36.

HIDEOUT

Assigns MERLIN's output to specified unit. Unit specification is prompt. If OPEN statements are used, connects the unit to the file:
HIDE(unitnumber). For example, unit 16 is connected to file: HIDE16.

REVEAL

Reassigns MERLIN's output to the default unit.

RESET

Resets the partial call-counter to zero.

MEMO

Appends the current point information to file STORE.

INIT

This command permits the user to initialize the variables (or to reset them) with values stored in some unit. This is done through a free-format READ statement, so no specific format is required. It prompts the user for the unit number. If OPEN statements are used, a filename is prompt associated with unit 46.

ACCUM

This command accumulates at random points in the N-dimensional space that result to an objective function's value less than a target value. It prompts the user for the number of desired points, the maximum number of calls to be spent and the target value.
To use this command all non-fixed variables must be framed, and values for all the variables must exist (i.e. this command will not work if initially no values are assigned in all variables). The accumulated points are stored in MERLIN's file INIPO.
Note that if all calls are spent and the desired number of points is not reached, no action is taken by MERLIN and the control is transfered to the command prompting point.

INSPECT

Initiates inspection of any of the files: DATA, BACKUP,DISPO,STORE,INIPO. The user is prompt to enter one of these filenames. The inspection is done record by record of information. To interrupt inspection (in order to subsequently use the PICK command to select a new point) any character besides a space or a $ sign is required.

PICK

Picks the last point inspected and makes it the current point. The previous point is appended to file DISPO.
PICK works only if issued immediately after an

INSPECT command.
DISCARD
Discards any one of the files: DISPO,STORE, BACKUP.
By this, it is understood that the specified file is rewound and overwritten with the current point information. File specification is prompt.

INFORMATION COMMANDS:
LIST
Displays all MERLIN commands.
HELP
Initiates a HELP session.
INTRODUCE
Informs the user of the extensions (if any) to the standard version of MERLIN that are implemented by other users.
EDITING COMMANDS:
MACRO
Initiates MACRO construction. It prompts the user for a MACRO-name and consecutive 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 MACRO-construction session. It is an otherwise inert command.
MEDIT
Initiates editing of the file MACROF, where in the various MACROs reside, by invoking the MEDIT editor.
The editor's commands are described in the user's manual (deactivated if the BATCH-option has been selected).

MINIMIZATION COMMANDS:
All minimization commands are using panel-prompting, if the option PANEL-ON is selected, and no prompting at all otherwise.
The underlying algorithms are described in the section that follows.
SIMPLEX
Invokes the SIMPLEX method of minimization, based on refs. [6,7].
ROLL
Invokes an adhoc but robust no-derivative method of minimization.
RANDOM
Invokes a random search no-derivative routine.
BFGS
Invokes a routine that uses the BFGS formula to update the inverse Hessian matrix [4].
DFP
Invokes a routine that uses the DFP formula to update the inverse Hessian matrix [4].
CONGRA
Invokes a routine that uses the conjugate gradient method of Polak and Ribiere [5].
The rest of the commands which appear when a LIST command is issued are dummy commands. The purpose of their existence is to ease the development a user may wish to pursue. Commands MIN10–MIN20 are meant to be minimization commands and updating of the files BACKUP and DATA is being performed, while commands USCOM13-USCOM20 are meant to be used as utility commands.

4. Minimization algorithms

a. The following algorithm is invoked by issuing the command SIMPLEX:

1) Starting from the initial point (P_0) calculate initial simplex vertices P_i, $i=1, N$ and the corresponding function values: $y_i = f(P_i)$, $i = 0, N$.
2) Determine $y_{low} = \min\{y_i, i=0, N\}$, $y_{high} = \max\{y_i, i=0, N\}$ and calculate (P^c) the centroid of all P_i excluding P_{high}.
   Set $P^* = (1+\alpha)P^c - \alpha P_{high}$ and calculate $y^* = y(P^*)$.
3) If $y^* < y_{low}$ then form $P^{**} = (1.+\gamma)P^* - \gamma P^c$, calculate $y^{**} = y(P^{**})$ and go to (4); else go to (5).
4) If $y^{**} < y^*$ then replace P_{high} by P^{**} and go to (6).
5) If $y^* > y_i$ for all y_i except y_{high} go to (7).
6) Replace P_{high} by P^* and go to (9).
7) If $y^* < y_{high}$ replace P_{high} by P^*. Form $P^{**} = (1.-\beta)P^c + \beta P_{high}$ and calculate $y^{**} = y(P^{**})$.

8) If $y^{**} > y_{high}$ then replace P_i by $(P_i + P_{low})/2.$, $i = 0, N$; else replace P_{high} by P^{**}.
9) Check if convergence criterion is satisfied. If so return y_{low}, P_{low} as the optimum value and point; else go to (2).

Initially the simplex vertices are taken as the minima in each of the N directions through the starting point P_0. If both margins exist for all non-fixed variables, there is an option to select the simplex vertices at random. Convergence is assumed when:

$$\frac{1}{N+1}\sum_{i=0}^{N} |y_i - f(P^c)| < \epsilon,$$

where ϵ, is a preset tolerance.

The values for α, β, γ are 1, 1/2, 2 as suggested in ref. [6].

b. The following algorithm is followed when the command RANDOM is issued.

Let the currently best point be $X^c = (X_1^c, X_2^c, \ldots, X_n^c)$ and let $f_c = f(X^c)$ be the corresponding value of the objective function. A box is constructed in the hyperspace of n dimensions, centered around the current point X^c. This is facilitated by picking for each variable, a step size $S_i > 0$ and the box is defined by the n intervals $(X_i^c - S_i, X_i^c + S_i)$.

A trial point $X^t = (X_1^t, X_2^t, \ldots, X_n^t)$ is sampled at random from inside the box and is accepted or rejected according to whether $f_t = f(X^t)$ is lower or not than f_c. If $f(X^t) < f_c$ the trial is successful; else it is said to be a failure.

In an attempt to gain in efficiency we include an optional line search in the direction of progress and an optional volume excluding technique for reshaping the box after every failure, which is described next.

Suppose that the box is defined by the set of the n intervals (l_j, r_i) and suppose that the sampled trial point X^t results to a failure. After the failure the box is redefined by the set of the following n-intervals:

$$(l_i', r_i') = \begin{cases} (l_i, X_i^t) & \text{if} \quad X_i^c < X_i^t \\ (X_i^t, r_i) & \text{if} \quad X_i^c > X_i^t \\ (l_i, r_i) & \text{if} \quad X_i^c = X_i^t \end{cases}$$

When the above option is not selected, the reshaping is done by merely multiplying the box sides with a reducing factor, whenever a preset number of consecutive failures occurs. This is called a failure cycle. One can instruct the routine to return control to the calling program if a preset number of consecutive failure cycles is reached.

c. The following algorithm is the basis of the procedure invoked by the command ROLL.

For each variable X_i there exists an associated step S_i.

For a problem where the objective function depends on n variables the procedure described below is repeated n times, as i takes on the values $1, 2, 3, \ldots, n$.

Let $X^c = (X_1^c, X_2^c, \ldots, X_n^c)$ be the current point and let $f_c = f(X^c)$:

1) Pick a trial point:

   $X_j^t = X_j^c$ *for all* $j \neq i$ and

   $X_i^t = X_i^c + S_i$.

2) Calculate $f_+ = f(X^t)$.
3) If $f_+ < f_c$ set $X^c = X^t$, $f_c = f_+$ and $S_i = aS_i$
   proceed from step 1) for the next value of i.
4) If $f_+ \geq f_c$ pick another trial point as:

   $X_j^t = X_j^c$ for all $j \neq i$ and

   $X_i^t = X_i^c - S_i$.

5) Calculate $f_- = f(X^t)$.
6) If $f_- < f_c$ set $X^c = X^t$, $f_c = f_-$ and $S_i = -aS$
   proceed from step 1) for the next value of i.
7) if $f_- \geq f_c$ calculate an appropriate step by:

   $S_i = -\frac{1}{2}(f_+ - f_-)/(f_+ + f_- - 2f_c)S_i$.

8) Proceed from step 1) for the next value of i.

In the above, a, is an enhancement factor chosen by the user.

If after looping over all variables there is no

progress, a line search is performed in the direction $S = (S_1, S_2, \ldots, S_n)$.

The above procedure is repeated until a preset number of calls to the objective function is reached, whereupon control is transfered to the calling program.

The routine terminates also when a preset number of consecutive failures is reached. We consider as a failure the case where either looping over all variables or after having performed the line search, the relative progress made (i.e. $[f_{\text{initial}} - f_{\text{final}}]/[f_{\text{initial}}]$) is less than a preset small number.

d. To describe the general algorithm followed by all the so-called quasi-Newton methods we need to define a few quantities. Following Fletcher [4], let $f(x)$ be the objective function with $x = (x_1, x_2, \ldots, x_n)$,let g be its gradient and G be its Hessian matrix, i.e.:

$$g_i = \frac{\partial f(x)}{\partial x_i} \quad \text{and} \quad G_{ij} = \frac{\partial^2 f(x)}{\partial x_i \, \partial x_j}$$

and let $H_{ij} = [G^{-1}]_{ij}$.

Since all these methods are iterative, we denote the kth iteration related quantities with a superscript in parentheses.

For instance $x^{(k)}$ is the vector x at the kth iteration, $H^{(k)}$ is the inverse Hessian at the kth iteration etc.

Let $\delta^{(k)} = x^{(k+1)} - x^{(k)}$ and $\gamma^{(k)} = g^{(k+1)} - g^{(k)}$. Then the algorithm is as:

1) Calculate the direction $s^{(k)} = -H^{(k)} g^{(k)}$
2) Perform a line-search along $s^{(k)}$ and obtain so $x^{(k+1)} = x^{(k)} + \delta^{(k)}$
3) Update $H^{(k)}$ to obtain $H^{(k+1)}$, calculate $g^{(k+1)}$, go to 1.

Initially one takes $H^{(1)} = I$, the unit matrix.

The updating, is the operation that distinguishes between methods.

For the DFP method the updating formula is:

$$H_{ij}^{(k+1)} = H_{ij} + \frac{\delta_i \delta_j}{\delta_m \gamma_m} - \frac{H_{ip}\gamma_p \gamma_q H_{qj}}{\gamma_m H_{mn} \gamma_n},$$

while for the BFGS it is given by:

$$H_{ij}^{(k+1)} = H_{ij} + \left(1 + \frac{\gamma_m H_{mn} \gamma_n}{\delta_\mu \gamma_\mu}\right) \frac{\delta_i \delta_j}{\delta_v \gamma_v} - \left(\frac{\delta_i \gamma_p H_{pj} + H_{ip} \gamma_p \delta_j}{\delta_v \gamma_v}\right).$$

Where in both cases we used the summation convention over repeated indices and we dropped the superscript k on the quantities on the right hand side.

e. The conjugate gradient methods do not use the Hessian matrix, they use only the gradient of the objective function. They create directions of search conjugate to each other. Among the many algorithms of that type the one due to Polak and Ribiere has displayed a staggering improvement in performance when solving problems with a large number of variables.

A sketch of the algorithm follows:

At the kth iteration:

1) Calculate: $S^{(k+1)} = -g^{(k+1)} + \beta^{(k)} S^{(k)}$, where:

$$\beta^{(k)} = \left(g^{(k+1)} - g^{(k)}\right) \cdot g^{(k+1)} / g^{(k)} \cdot g^{(k)}$$

2) Perform a line-search along the direction $S^{(k+1)}$ obtaining so the new point, calculate the new gradient and go to 1.

Initially we take the steepest descent direction: $S^{(1)} = -g^{(1)}$.

5. User-supplied routines

A) SUBROUTINE GRANAL(N,X,VAL,GRAD)
INPUT: N,X,VAL
OUTPUT: GRAD
X is a real array dimensioned as X(N).
X(N) contains the values of the variables.
VAL the value of the objective function.
GRAD is a real array dimensioned as GRAD(N).
GRAD(N) upon return, should contain the components of the gradient.

This subroutine is optional. MERLIN calculates the gradient numerically. If however, the partial derivatives are analytically known or they can be computed in some other convenient way, the user should take advantage of it. The above subroutine, when provided, is meant to serve this purpose.

B) FUNCTION FUNMIN(X,N)
INPUT: N,X
X is an array, dimensioned as X(N) and contains the values of the variables. N is the dimensionality of the problem.

This function subprogram is the implementation of the objective function i.e., upon return FUNMIN(X,N) should be equal to the objective function evaluated at the input point X(N):

As an example consider as objective function the following:

$$f(x_1, x_2, x_3) = (x_1 - 3)^2 + 5x_2^2(x_3 - x_1)^4 + 10x_3^2(100 - x_1x_3)^2$$

with partial derivatives:

$$\partial f/\partial x_1 = 2(x_1 - 3) - 20x_2^2(x_3 - x_1)^3 - 20x_3^3(100 - x_1x_3),$$

$$\partial f/\partial x_2 = 10x_2(x_3 - x_1)^4,$$

$$\partial f/\partial x_3 = 20x_2^2(x_3 - x_1)^3 + 20x_3(100 - x_1x_3)^2 - 20x_1x_3^2(100 - x_1x_3).$$

The two routines appearing in table 3, implement the objective function and the gradient calculation.

As one can see from this example, the VAL input-argument to the GRANAL subroutine may or may not be used. Its presence serves the purpose of saving additional calls to the objective function, if in the calculation of the gradient the function's value is needed.

6. Special features

Merlin provides a friendly and convenient environment to work with, and at the same time, powerful optimization routines. Our attitude, based on the fact that there does not exist a single algorithm to be used in all cases as a panacea, is that the user should be able to try easily the various different methods, in order to find the ones most suitable to his problem. Very important is also the capability to store points considered to be potentially useful and pick points that have previously been stored.

In cases where there exist a lot of local minima inside a known region, one may be able to reach them by starting from different initial points. So accumulating various initial points at random in the specified region that correspond to objective fucntion's values lower than a user-set upper bound, can be quite helpful.

Table 3

```
      FUNCTION FUNMIN(X,N)
      DIMENSION X(N)
      X1 = X(1)
      X2 = X(2)
      X3 = X(3)
      FUNMIN = (X1-3)**2+5.*X2**2*(X3-X1)**4+10.*X3**2*(100-X1*X3)**2
      RETURN
      END

      SUBROUTINE GRANAL(N,X,VAL,GRAD)
      DIMENSION X(N),GRAD(N)
      X1 = X(1)
      X2 = X(2)
      X3 = X(3)
      GRAD(1) = 2.*(X1-3.)-20.*X2**2*(X3-X1)**3-20.*X3**3*(100-X1*X3)
      GRAD(2) =10.*X2*(X3-X1)**4
      GRAD(3) = 20.*X2**2*(X3-X1)**3+20.*X3*(100-X1*X3)**2-20.*X3**2
     > *X1*(100.-X1*X3)
      RETURN
      END
```

Creation of command packages (Macros) is quite convenient and important especially when one deals often with the same type of functions. In a macro one can store a successful strategy for the type of problems he usually faces and use it over and over. A collection of such macros is therefore useful since they accelerate the process of minimization.

The input operations are handled in a natural way which we call the "one-line-input". The maximum length of the input record is 79 characters.

When only variable specification is required (as for example in the FIX command) one enters the indices (or names) of the associated variables, separated by blanks or commas.

When assigning values to variables (as for example in the STEP command), the input line consists of pairs, containing first the index (or name) of the variable, and then the corresponding value.

For the minimization routines (as well as for the graph-routine) the input is assisted by panels. The panels show a list of variables (that are input to the subprogram to be called) along with their current values, a short explanation of their role and an indicative menu. These variables do not have names but they are associated with indices that are displayed on the panel. To change their values, one follows the philosophy described above of the one-line-input. Representative panels are sown in table 4.

A full description of all Merlin's special features can be found in the user's manual.

Merlin has been successfully used in several calculations. As examples we refer to X-ray and ionic conductivity data fits to the theoretical models, study of nuclear interactions via potential modeling, Compton profile calculations, solution of linear systems when other methods suffer from numerical instabilities, study of chemical reactions and reaction path fitting, variational calculation of the ground state for simple systems etc. We list collectively under reference [8], several works in which MERLIN was used to solve the relevant optimization problems that came-up.

In the near future we intent to present a special version of this program appropriate to CDC-machines that will take advantage of the NOS operat-

Table 4

RANDOM-PANEL

INDEX	DESCRIPTION	VALUE	MENU
1)	NUMBER OF CALLS	1000	ANY INTEGER
2)	ACTIVATE/DEACTIVATE VOLUME EXCLUSION	0	(0/1)
3)	STEP FACTOR	0.70	ANY REAL IN (0, 1)
4)	CYCLE-SIZE	30	ANY INTEGER
5)	NUMBER OF CONSECUTIVE CYCLE FAILURES ALLOWED	5	ANY INTEGER
6)	LINE-SEARCH PERIOD	3	ANY INTEGER
7)	PRINTOUT SELECTION	1	0/1/2
8)	CANCEL-BUTTON	1	(0/1)

ENTER CHANGES
? 1 3000 3 0.4 4 20 5 6 5

SIMPLEX-PANEL

INDEX	DESCRIPTION	VALUE	MENU
1)	INITIALIZATION SCHEME (SYSTEMATIC/RANDOM)	1	(1/2)
2)	INITIALIZATION TOLERANCE	0.0	ANY REAL
3)	INIT. CALLS/VARIABLE	100	ANY INTEGER
4)	SIMPLEX TOLERANCE	0.0	ANY REAL
5)	SIMPLEX CALLS	1000	ANY INTEGER
6)	PRINTOUT SELECTION	0	(0/1/2)
7)	CANCEL-BUTTON	1	(0/1)

ENTER CHANGES
? 2 0.001 3 50 5 500

ing system through non-standard FORTRAN and assembly language routines [9]. This will be quite convenient since it will permit issuing of NOS commands familiar to the CYBER users, will contain additional file manipulation commands, priority control etc.

A MERLIN control language is currently under intensive development [10], which will enable the user to program MERLIN according to his particular needs.

Finally, we report that following the installation instructions given in the user's manual, we were able to run this program on APPLE's 512k Macintosh, using the Microsoft FORTRAN'77 compiler.

Acknowledgements

We would like to acknowledge the contributions made in the course of the development of this work by several colleagues and friends. Special thanks are due to Dr. I. Demetriou, V. Voutsadakis and D. Papageorgiou for illuminating discussions, Dr. T. Bakas, Dr. C. Kotsis, D. Katsanos, C. Hasapis, T. Liakopoulos and C. Hatzigeorgiou for useful suggestions after running innumerous times and reading the several preliminary versions of the manuscript, the computer center of the University of Ioannina for generous provision of computer time on a CDC-CYBER 171 and to professors N.G. Alexandropoulos and C. Sakarelos for constant support and encouragement. We also thank Th. Fresta, for carrying out efficiently the difficult task of typing the various different versions of the manuscript.

References

[1] T. Hamada and I.D. Johnston, Nucl. Phys. 34 (1962) 382.
R.V. Reid, Ann. Phys. 50 (1968) 411.
M. Laccombe, B. Loiseau, J.M. Richard, R. Vinh Mau, J. Cote, P. Pires and R. de Tourreil, Phys. Rev. C21 (1980) 861.
I.E. Lagaris and V.R. Pandharipande, Nucl. Phys. A359 (1981) 331.
R.B. Wiringa, R.A. Smith and T.L. Ainsworth, Phys. Rev. C29 (1984) 1207.

[2] N. Sathyamurthy, Comput. Phys. Rep. 3 (1985) 1.
S. Bell and J.S. Crighton, Chem. Phys. 80 (1984) 2465.

[3] F. James and M. Roos, Comput. Phys. Commun. 10 (1975) 343.

[4] R. Fletcher, Practical Methods of Optimization (John Wiley, New York).
R. Fletcher and M.J.D. Powell, Computer J. 6 (1963) 163.

[5] E. Polak, Computational Methods in Optimization: A Unified Approach (Academic Press, New York, 1971).

[6] J.A. Nedler and R. Mead, Computer J. 7 (1965) 308.

[7] K. Mika and Th. Chaves, Berichte der KFA Jülich No. 1643 (1980).

[8] N.I. Papanicolau, N.C. Bacalis and D.A. Papaconstantopoulis, Z. Phys. B 65 (1987) 453.
G. Pantis and I.E. Lagaris, Z. Phys. A 321 (1985) 149.
I.N. Demetropoulos, S.C. Plaskasovitis and I.E. Lagaris, Abstracts, 2nd Intern. Symp. on Kinetics in Analytical Chemistry, PIII-18 (1986).
C. Kotsis, Doctoral Dissertation, University of Ioannina (1986).

[9] D.G. Papageorgiou, C.S. Hassapis and I.E. Lagaris, in preparation.

[10] C.S. Hassapis, D.G. Papageorgiou and I.E. Lagaris, in preparation.

INPUT DECK

```
3                                   1   ROLL                    14
POINT                               2   5 0 1 800 2 0           15
1 30 2 30 3 33.88                   3   .S                      16
0                                   4   SIMPLEX                 17
GODFATHER                           5   5 2000                  18
X                                   6   .S                      19
Y                                   7   ANAL                    20
Z                                   8   BFGS                    21
MACRO                               9   1 2000 6 0              22
.S                                 10   DFP                     23
SHORTDIS                           11   1 2000 6 0              24
CLEAR                              12   .S                      25
.S                                 13   STOP                    26
```

OUTPUT

```
ENTER # OF VARIABLES
-------------------------------------------------------------------------
.............................          MERLIN - 1.0  *** STANDARD VERSION
.............................     G.A.EVANGELAKIS,J.P.RIZOS,I.E.LAGARIS
.............................            PHYSICS DEPARTMENT
.............................     I.N.DEMETROPOULOS
.............................            DEPARTMENT OF CHEMISTRY
 UNIVERSITY OF IOANNINA. OCTOBER 1986
    IOANNINA - G R E E C E
-------------------------------------------------------------------------
    ANOTHER  MINIMIZATION SESSION STARTS.
    THERE IS A HELP FACILITY ASSOCIATED WITH A
    HELP-FILE ASSIGNED TO UNIT  42
    MAKE SURE THIS FILE IS PRESENT TO PROVIDE ON-LINE  INFORMATION
-------------------------------------------------------------------------
ESTIMATED MACHINE'S ACCURACY: 1.E-14


  ....  W A R N I N G  ....
...  INITIALIZE VARIABLES  ...


   /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
POINT
POINT


   /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
GODFATHER
VARIABLE  1  :
VARIABLE  2  :
VARIABLE  3  :
          1) .... X
          2) .... Y
          3) .... Z

   /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
MACRO
** ENTER MACRO - NAME **
** MACRO EDITING STARTS **
** ENTER DESIRED COMMAND **
** ENTER DESIRED COMMAND **
 ** MACRO EDITING COMPLETE **
```

```
    /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
SHORTDIS
TOTAL NU. OF CALLS = 1
NU. OF CALLS SINCE LAST RESET  1
  1) X          .......    30.000000000000
  2) Y          .......    30.000000000000
  3) Z          .......    33.880000000000
       VALUE.....          9640575114.3915

                *** END OF MACRO PROCESSING ***

    /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
ROLL
-------------------------------------------------------------------------------
                          ROLL - PANEL
-------------------------------------------------------------------------------
 INDEX    DESCRIPTION                VALUE       MENU
   1)   NUMBER OF CALLS               300        ANY INTEGER
   2)   TOLERANCE                    .100E-02    ANY REAL IN (0,1)
   3)   STEP FACTOR                  .30E+01     ANY REAL > 1
   4)   FAILURES ALLOWED              4          ANY INTEGER
   5)   PRINTOUT SELECTION            1          0/1/2
   6)   WALL-PARAMETER               3          ANY INTEGER
   7)   CANCEL-BUTTON                 1          ( 0/1 )
-------------------------------------------------------------------------------
   ENTER CHANGES
       NUMBER OF      ROLL  CALLS:        801

    /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
SHORTDIS
TOTAL NU. OF CALLS = 802
NU. OF CALLS SINCE LAST RESET  802
  1) X          .......    37.950240707030
  2) Y          .......    -.568434188608086E-13
  3) Z          .......    2.6351484637018
       VALUE.....          1221.5207432240

                *** END OF MACRO PROCESSING ***

    /\/\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
SIMPLEX
-------------------------------------------------------------------------------
                        SIMPLEX - PANEL
-------------------------------------------------------------------------------
 INDEX    DESCRIPTION                VALUE       MENU
   1)   INITIALIZATION SCHEME           1        (1/2)
        (SYSTEMATIC/RANDOM)
   2)   INITIALIZATION TOLERANCE      100E-01    ANY REAL
   3)   INIT. CALLS/VARIABLE          100        ANY INTEGER
   4)   SIMPLEX TOLERANCE            0.         ANY REAL
   5)   SIMPLEX CALLS                 100        ANY INTEGER
   6)   PRINTOUT SELECTION            0         (0/1/2)
   7)   CANCEL-BUTTON                 1          ( 0/1 )
-------------------------------------------------------------------------------
   ENTER CHANGES
       NUMBER OF   SIMPLEX  CALLS:       2118
```

```
    /\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
SHORTDIS
TOTAL NU. OF CALLS = 2920
NU. OF CALLS SINCE LAST RESET  2920
  1) X          .......   3.8716684404021
  2) Y          .......  -.568434188608E-13
  3) Z          .......   25.828400351577
       VALUE.....         .76644352226356

               *** END OF MACRO PROCESSING ***

    /\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
ANAL

    /\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
BFGS
-------------------------------------------------------------------------------
                          BFGS - PANEL
-------------------------------------------------------------------------------
 INDEX    DESCRIPTION                 VALUE       MENU
   1)   NUMBER OF CALLS                300         ANY INTEGER
   2)   TOLERANCE                     .100E-02     ANY REAL IN (0,1)
   3)   ERROR-BOUND                  .10E-01      ANY REAL < 1
   4)   USE/RECALCULATE GRADIENT       0           (1/0)
   5)   USE/RESET  HESSIAN            0           (1/0)
   6)   PRINTOUT SELECTION            1           0/1/2
   7)   WALL-PARAMETER                3           ANY INTEGER
   8)   CANCEL-BUTTON                 1           ( 0/1 )
-------------------------------------------------------------------------------
   ENTER CHANGES
       NUMBER OF      BFGS  CALLS:       2001

    /\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
DFP
-------------------------------------------------------------------------------
                          DFP - PANEL
-------------------------------------------------------------------------------
 INDEX    DESCRIPTION                 VALUE       MENU
   1)   NUMBER OF CALLS                300         ANY INTEGER
   2)   TOLERANCE                     .100E-02     ANY REAL IN (0,1)
   3)   ERROR-BOUND                  .100E-01     ANY REAL IN (0,1)
   4)   USE/RECALCULATE GRADIENT       0           (1/0)
   5)   USE/RESET  HESSIAN            0           (1/0)
   6)   PRINTOUT SELECTION            1           0/1/2
   7)   WALL-PARAMETER                3           ANY INTEGER
   8)   CANCEL-BUTTON                 1           ( 0/1 )
-------------------------------------------------------------------------------
   ENTER CHANGES
       NUMBER OF       DFP  CALLS:       1188

    /\/\/\/\/\     MERLIN   IS AT YOUR COMMAND !!!
SHORTDIS
 TOTAL NU. OF CALLS = 6109
 NU. OF CALLS SINCE LAST RESET  6109
   1) X          .......   3.0000000000132
   2) Y          .......   .2135117241379 3E-14
   3) Z          .......   33.333333333187
        VALUE.....         .19321191098585E-21

               *** END OF MACRO PROCESSING ***

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