# FMCGA - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS

J.W. Bandler and M.A. El-Gamal

SOS-84-7-U

June 1984

# FMCGA - A FORTRAN PACKAGE FOR UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS

J.W. Bandler and M.A. El-Gamal

Abstract

FMCGA is a package of six subroutines for minimization of an unconstrained function by the new three term conjugate gradient method. The package implements Nazareth's conjugate gradient formula for generating conjugate search directions even with inexact line search. It also incorporates Dixon's gradient prediction method to retain the finite termination property when applied to quadratic functions. The peformance results for this combination outperforms a wide subset of current conjugate gradient codes. The package has been tested succesfully on problems involving both a small and a large number of variables, on problems with varying degrees of non-linearity, and on problems evolving from practical applications such as the solution of sets of non-linear equations. The package and user-oriented documentation have been developed for the CDC 170/815 system with the NOS 2.1 level 558 operating system and the Fortran Extended (FTN) version 4.8 compiler.

---

This work was supported by the Natural Sciences and Engineering Reseach Council of Canada under Grant A7239.

The authors are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

# I. INTRODUCTION

This report gives a user-oriented description of the package FMCGA of six Fortran subroutines to minimize an unconstrained function by conjugate gradients with inexact line search. The package and documentation have been prepared for the CDC 170/815 system with the NOS 2.1 level 558 operating system and the Fortran Extended (FTN) version 4.8 compiler.

The purpose of producing this package is to provide a simple user-oriented program of unconstrained function minimization. The user has to supply the following

a) the main segment that prepares arguments and calls the main subroutine of the package,

b) two subroutines which evaluate the objective function and its gradients w.r.t. the variables.

The minimization algorithm which is implemented by the package generates conjugate search directions that remain conjugate even with approximate line searches and maintains finite termination when applied to quadratic functions.

In this report we will first review the properties of conjugate gradient algorithms. We will then present the algorithm implemented by the package [1-3] illustrating its advantages over existing algorithms.

Information on the structure of the FMCGA package and how to access it on the CDC 170/815 system at McMaster University is given.

Different examples to familiarize the user with the package are presented.

# II. GENERAL DESCRIPTION

Introduction

Conjugate gradient methods minimize a function $F(\mathbf{x})$ of n variables starting from an initial point $\mathbf{x}^1$, by searching in a sequence along directions $\mathbf{p}^1$, $\mathbf{p}^2$, ..., $\mathbf{p}^j$, ... and obtaining

successive approximations $\mathbf{x}^2$, $\mathbf{x}^3$, ..., $\mathbf{x}^j$, ... to the minimum. They are described by the relations

$$\mathbf{p}^1 = -\mathbf{g}^1, \tag{1a}$$

$$\mathbf{p}^j = -\mathbf{g}^j + \beta^j \mathbf{p}^{j-1}, \tag{1b}$$

$$\mathbf{x}^{j+1} = \mathbf{x}^j + \alpha^j \mathbf{p}^j, \alpha^j \neq 0, \tag{1c}$$

where $\mathbf{g}^j$ is the gradient of $F(\mathbf{x})$ at $\mathbf{x}^j$, $\beta^j$ [2] is a scalar most commonly given by

$$\beta^j = \frac{(\mathbf{g}^j)^T \mathbf{g}^j}{(\mathbf{g}^{j-1})^T \mathbf{g}^{j-1}} \tag{2a}$$

and $\alpha^j \neq 0$ is a scalar chosen so that $\mathbf{x}^{j+1}$ minimizes $F(\mathbf{x})$ along $\mathbf{p}^j$, i.e.,

$$(\mathbf{p}^j)^T \nabla F(\mathbf{x}^{j+1}) = 0. \tag{2b}$$

Among the properties of such methods when applied to a quadratic function

$$F(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + 0.5\, \mathbf{x}^T \mathbf{A}\, \mathbf{x}, \tag{3}$$

where $\mathbf{A}$ is an n x n positive-definite symmetric matrix, are the following

a) generation of conjugate directions, i.e., directions satisfying the relation

$$(\mathbf{p}^i)^T \mathbf{A}\, \mathbf{p}^j = 0, \ \forall \ i \neq j \tag{4}$$

b) orthogonality of gradients at different iterates,

c) finite termination in at most n steps.

However, in order for these properties to hold, it is necessary that relation (2b) be satisfied at each step, i.e., that $\alpha^j$ be chosen so that the line search is exact.

Many variants of the basic method have been proposed. The Polak-Ribierre [4] suggestion is

$$\beta^j = \frac{(\mathbf{y}^{j-1})^T \mathbf{g}^j}{(\mathbf{g}^{j-1})^T \mathbf{g}^{j-1}} \tag{5a}$$

where

$$\mathbf{y}^{j-1} = \mathbf{g}^j - \mathbf{g}^{j-1} \tag{5b}$$

It follows from the above properties a) and b) that, for quadratic functions and exact line searches, the alternative choices for $\beta^j$ are equivalent, but for nonquadratic functions or inexact line searches, each choice leads to a distinct algorithm. There is little conclusive numerical evidence on which of the laternative fomulas for $\beta$ is preferable but in [4] Powell recommends the Polak-Ribierre form (5a). He shows that if the orthogonality property $(\mathbf{g}^j)^T \mathbf{g}^{j-1} = 0$ has already been lost, then the Polak-Ribierre form bounds the angle between $\mathbf{p}^j$ and $\mathbf{g}^j$ and can produce a much more downhill direction.

The New Conjugate Gradient Method

The FMCGA package implements the new conjugate gradient algorithm in [2]. This algorithm is based on Nazareth's formula for generating conjugate search directions. This formula has the property that it maintains mutual conjugacy of all search directions over a quadratic function even when the line search is not exact. The recurrence relations for developing successive search directions are given by

$$\mathbf{p}^1 = -\mathbf{g}^1 \tag{6a}$$

and

$$\mathbf{p}^{j+1} = -\mathbf{y}^j + \frac{(\mathbf{y}^j)^T \mathbf{y}^j}{(\mathbf{y}^j)^T \mathbf{p}^j} \mathbf{p}^j + \frac{(\mathbf{y}^{j-1})^T \mathbf{y}^j}{(\mathbf{y}^{j-1})^T \mathbf{p}^{j-1}} \mathbf{p}^{j-1} . \tag{6b}$$

However, in order to be able to retain the finite termination property when line searches are not exact, the algorithm incorporates the gradient prediction method [2,5]. In this method two additional vectors are stored. The basis of the method is to generate the same set of conjugate directions that would have resulted from the use of perfect line searches and to store a correction vector.

Suppose that a step $\mathbf{d}^j$ is taken along the direction $\mathbf{p}^j$ to a new point $\mathbf{x}^{j+1}$, where the new gradient is $\mathbf{g}^{j+1}$ but that, if a perfect step had been taken, the step would have been $(\mathbf{d}^j)^*$. Then from the properties of a quadratic function, the following ratios are equal

$$\theta^j = \frac{\|(\mathbf{d}^j)^*\|}{\|\mathbf{d}^j\|} = -\frac{(\mathbf{p}^j)^T \mathbf{g}^j}{(\mathbf{p}^j)^T \mathbf{y}^j} . \tag{7}$$

This enables us to predict both the point that would have been reached with a perfect line search and the gradient at that point. Also, due to the conjugancy of the directions generated these corrections are independent and we can define $(\mathbf{x}^{j+1})^*$ and $(\mathbf{g}^{j+1})^*$, the point and the gradient that would have been reached after j perfect line searches, as

$$(\mathbf{g}^{j+1})^* = \mathbf{g}^{j+1} - \mathbf{w}^{j+1}, \tag{8a}$$

and

$$(\mathbf{x}^{j+1})^* = \mathbf{x}^{j+1} - \mathbf{z}^{j+1}, \tag{8b}$$

where

$$\mathbf{w}^{j+1} = \mathbf{w}^j - (\theta^j - 1)\,\mathbf{y}^j, \mathbf{w}^1 = \mathbf{0} , \tag{9}$$

$$\mathbf{z}^{j+1} = \mathbf{z}^j - (\theta^j - 1)\,\mathbf{d}^j, \mathbf{z}^1 = \mathbf{0} \tag{10}$$

and

$$\theta^j - 1 = -\frac{(\mathbf{p}^j)^T \mathbf{g}^{j+1}}{(\mathbf{p}^j)^T \mathbf{y}^j} . \tag{11}$$

For a quadratic function if $(\mathbf{g}^j)^* = \mathbf{0}$, it indicates that with perfect line searches, $(\mathbf{x}^j)^*$ would be the solution and a step $\mathbf{z}^j$ is necessary to obtain the solution.

For nonquadratic functions the algorithm is iterative rather than n step. The directions $\mathbf{p}^j$ that are generated are those corresponding to the current local quadratic approximation to the function.

Restarting Procedures

In the early algorithms the restarting strategy was usually to restart only after n or n+1 iterations. However, when n is large this can be very inefficient. There is therefore general agreement that occasional restarting is very helpful in practice. Accordingly, it is felt desirable to restart more regularly. First of all the algorithm is modified to reject the search direction if it lies too nearly along a constant contour of the objective function passing through

$\mathbf{x}^j$, since such a direction would not give much reduction in the value of the objective function. If $\phi$ is the limiting angle with $-\nabla F(\mathbf{x}^j)$ (see Fig. 1), then the new search direction would be used if

$$-(\mathbf{p}^j)^T \nabla F(\mathbf{x}^j) > \|\mathbf{p}^j\| \, \|\nabla F(\mathbf{x}^j)\| \cos\phi \ . \tag{12}$$

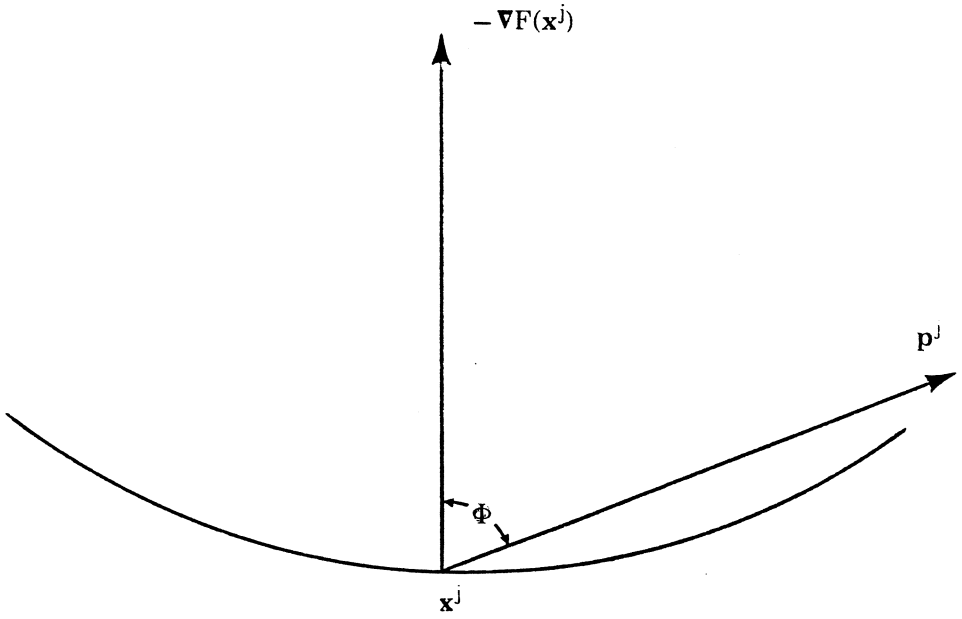


Fig. 1 Graphical representation of the limiting angle.

If this condition is not satisfied then $\mathbf{p}^j$ is generally replaced by $-\nabla F(\mathbf{x}^j)$ and the algorithm is restarted. The algorithm is also restarted [4] whenever

$$(\nabla F(\mathbf{x}^j))^T (\nabla F(\mathbf{x}^{j-1})) \geq 0.2 \, \|\nabla F(\mathbf{x}^j)\|^2 \ . \tag{13}$$

The left hand side in (13) would be zero if the conjugate gradient algorithm were working with perfect line searches on a quadratic function. Its size is, therefore, an indicator of the nonconjugacy of the search directions and so is an indicator that the algorithm should be restarted. Finally, the algorithm is restarted whenever

$$\|(\mathbf{g}^j)^*\| \leq \varepsilon_1 \, \|\mathbf{g}^j\| \tag{14a}$$

or

$$\|\mathbf{p}^j\| \leq \varepsilon_1 \, \|\mathbf{g}^j\|, \tag{14b}$$

as this implies that enough iterations have been undertaken to approximately minimize a quadratic function closely related to the nonlinear objective function.

Inexact Line Search

Quite obviously, accurate minimization along each search direction can be expensive in function evaluations. The results quoted earlier show that high accuracy is not required in most cases, but that the failure to obtain a function decrease can occur if the accuracy is too low. There is also the possibility of finding a local minimum with an increased function value if the function is not unimodal along the search direction. There is therefore great concern to relax the requirement of minimization and to replace it with a stability requirement that the function must decrease at every step.

Now, the special properties of the conjugate gradient algorithms are related to their use on quadratic functions, and for such functions a single step resulting from a quadratic or cubic fit attains the minimum along the line. Thus, the finite termination property will not be impaired if only a single such fit is made at each iteration, and since the special properties do not apply in a non quadratic region nothing is lost by not continuing to the minimum in this case. It seems therefore, important to employ a strategy that uses just enough quadratic or cubic fits to secure a function decrease. In fact, mere stability is not sufficient to ensure convergence. Wolfe [6] gave a sufficient conditions for convergence of well-behaved functions.

The line search in the implementation discussed here consists of first performing a parabolic interpolation using the values of the function and its derivatives at the starting point and the function value at an offset point. If the predicted function value does not satisfy Wolfe's conditions II and III [6], then the Armijio [7] procedure is adopted by halving or doubling the predicted step $\alpha^j$ until

$$F(\mathbf{x} + \alpha^j \mathbf{p}^j) < F(\mathbf{x}^j) + 0.1\, \alpha^j (\nabla F(x^j))^T \mathbf{p}^j$$

and

$$F(\mathbf{x}^j + 2\, \alpha^j \mathbf{p}^j) > F(\mathbf{x}^j) + 0.2\, \alpha^j (\nabla F(x^j))^T \mathbf{p}^j.$$

The selected value of $\alpha^j$ is the one that corresponds to the least function value obtained throughout this procedure. This new line search strategy, although simple, works quite well in almost all cases.

Termination Criteria

The algorithm is terminated when any one of the following conditions is satisfied:

1) the required accuracy is obtained, i.e.,

$$\|\mathbf{g}^j\| \leq \varepsilon_2, \text{ or } \|\mathbf{x}^{j+1} - \mathbf{x}^j\| \leq \varepsilon_2 , \tag{15}$$

2) an uphill search direction is obtained, which can only be due to rounding errors and the required accuracy cannot be obtained in this case,

3) the number of iterations exceeds the limit defined by the user.

Algorithm

The structure of the algorithm [2] consists of the following steps, where we assume the notation $\mathbf{t}_1 = \mathbf{t}^{j+1}$, $\mathbf{t}_0 = \mathbf{t}^j$, $\mathbf{t}_{-1} = \mathbf{t}^{j-1}$ etc. and $\mathbf{t}$ is any given vector.

Step 1 Choose $\mathbf{x}_0$, set $j \leftarrow 0$, iter $\leftarrow 0$.

Step 2 Evaluate $\mathbf{g}_0$, set $\mathbf{p}_0 \leftarrow -\mathbf{g}_0$ .

Comment Start with the steepest descent direction.

Step 3 Perform an approximate line search to find $\alpha_0$.

Comment For details, see the Inexact Line Search discussed in this section.

Step 4 Set $\mathbf{d}_0 \leftarrow \alpha_0\mathbf{p}_0$, $\mathbf{x}_1 \leftarrow \mathbf{x}_0 + \mathbf{d}_0$, iter $\leftarrow$ iter + 1,

$j \leftarrow j+1, \mathbf{g}_{-1} \leftarrow \mathbf{g}_0$ .

Evaluate

$\mathbf{g}_0 = \mathbf{g}(\mathbf{x}_1)$.

Comment Determine the new point and the gradient at this point.

Step 5 If $\|\mathbf{g}_0\| < \varepsilon_2$ or $\|\mathbf{x}_1 - \mathbf{x}_0\| < \varepsilon_2$ stop.

If iter $\geq$ iter max stop.

Set $\mathbf{d}_{-1} \leftarrow \mathbf{d}_0$, $\mathbf{p}_{-1} \leftarrow \mathbf{p}_0$

If $j \geq 2$ set $\mathbf{p}_{-2} \leftarrow \mathbf{p}_{-1}$

Step 6 Calculate

$$\mathbf{y}_0 = \mathbf{g}_0 - \mathbf{g}_{-1}$$

$$\mathbf{p}_0 = -\mathbf{y}_0 + \beta \mathbf{p}_{-2} + \gamma \mathbf{p}_{-1}$$

$$\mathbf{z}_0 = \mathbf{z}_{-1} - (\theta-1)\mathbf{d}_{-1} \quad , \quad \mathbf{z}_{-1} = \mathbf{0} \quad \text{if } j = 1$$

$$\mathbf{w}_0 = \mathbf{w}_{-1} - (\theta-1)\mathbf{y}_0 \quad , \quad \mathbf{w}_{-1} = \mathbf{0} \quad \text{if } j = 1$$

$$\mathbf{g}_0^* = \mathbf{g}_0 - \mathbf{w}_0$$

where

$$\theta - 1 = -\frac{\mathbf{g}_0^T \mathbf{p}_{-1}}{\mathbf{y}_0^T \mathbf{p}_{-1}} \qquad j \geq 1,$$

$$\beta = \begin{cases} \dfrac{\mathbf{y}_0^T \mathbf{y}_{-1}}{\mathbf{y}_{-1}^T \mathbf{p}_{-2}} & j \geq 2 \\ 0 & j = 0,1 \end{cases}$$

$$\gamma = \frac{\mathbf{y}_0^T \mathbf{y}_0}{\mathbf{y}_0^T \mathbf{p}_{-1}}$$

Comment Determine the new search direction using Nazareth's three term formula (6).

Step 7 If $C_1 \|\mathbf{p}_0\| \, \|\mathbf{g}_0\| \geq - \mathbf{p}_0{}^T \mathbf{g}_0$ then go to Step 9.

If $\|\mathbf{g}_0{}^*\| \leq C_1 \|\mathbf{g}_0\|$ or $\|\mathbf{p}_0\| \leq C_1 \|\mathbf{g}_0\|$ then go to Step 9.

Comment The first condition is Wolfe's condition I(12) where $C_1 = \cos\phi$. The significance of these tests is discussed in the Restarting Procedures of this section.

Step 8 If $j \leq n$ set $\mathbf{z}_{-1} \leftarrow \mathbf{z}_0$, $\mathbf{w}_{-1} \leftarrow \mathbf{w}_0$, $\mathbf{x}_0 \leftarrow \mathbf{x}_1$.

Go to Step 3.

Step 9 If $C_1 \|\mathbf{z}_0\| \, \|\mathbf{g}_0\| < - \mathbf{z}_0{}^T \mathbf{g}_0$ then go to Step 11.

Comment Test $\mathbf{z}_0$ for Wolfe's condition I (12).

Step 10 Set $\mathbf{p}_0 \leftarrow - \mathbf{g}_0$, $\mathbf{x}_0 \leftarrow \mathbf{x}_1$, $j \leftarrow 0$.

Go to Step 3.

Comment Restart with the steepest descent direction.

Step 11 Set $\mathbf{p}_0 \leftarrow \mathbf{z}_0$, $\mathbf{x}_0 \leftarrow \mathbf{x}_1$, $j \leftarrow 0$.

Go to Step 3.

Comment Restart with $\mathbf{z}_0$ as the new search direction.

## III. STRUCTURE OF THE PACKAGE

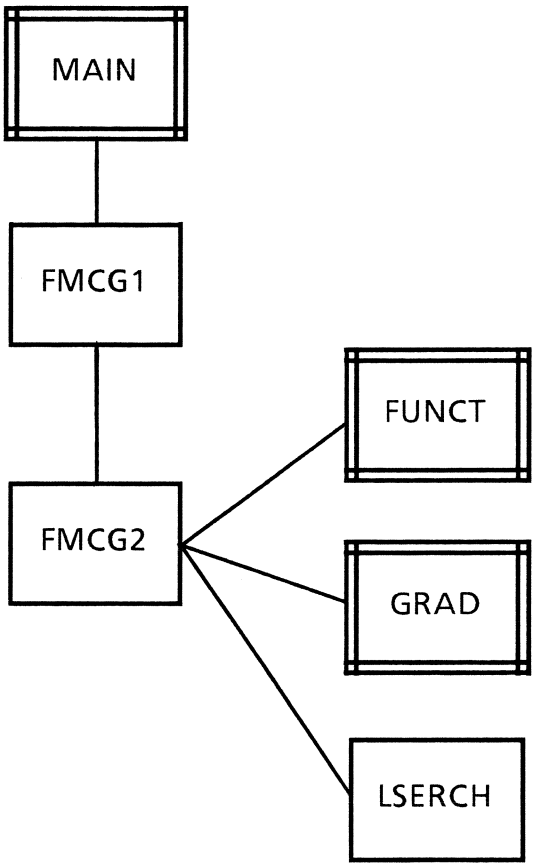


Fig. 2 Structure of the main subroutines of the FMCGA package.

The package is composed of three main subroutines FMCG1, FMCG2 and LSERCH. FMCG1 subdivides the work space (defined by the user) into a set of vectors used by the remaining subroutines and calls FMCG2. FMCG2 controls the minimization procedure, generates a sequence of conjugate search directions, a corresponding sequence of points and checks the convergence of the algorithm. FMCG2 also calls LSERCH at each step to implement the line search. The line search chosen is inexact but incorporates a parabolic interpolation and terminates at a point that satisfies Wolfe's conditions II, III [6]. There are

also three function subprograms, namely, VALUE, ANORM and BNORM. VALUE calculates the quotient of the product of the two vectors A and B divided by the product of C and D. ANORM calculates the norm of vector A. BNORM calculates the product of the two vectors A and B.

The main segment MAIN, subroutine FUNCT for evaluating the objective function and subroutine GRAD for evaluating the gradients of the objective function must be supplied by the user.

All subroutines of the FMCGA package are listed in an alphabetical order in Table I.

TABLE I

LIST OF SUBROUTINES OF THE FMCGA PACKAGE

| | Subroutines | Number of Lines (source text) | Listing (page [3]) |
|---|---|---|---|
| 1. | ANORM | 14 | 10 |
| 2 | BNORM | 15 | 10 |
| 3 | FMCG1 | 100 | 4 |
| 4 | FMCG2 | 257 | 6 |
| 5 | LSERCH | 129 | 11 |
| 6 | VALUE | 17 | 10 |

## IV. HOW TO USE THE PACKAGE

The package is available at McMaster University on the CDC 170/815 system as a permanent indirect group file LIBFMCG under the charge RJWBAND. It exists as a library of binary relocatable subroutines.

The general sequence of NOS commands to use the FMCGA package may be as follows:

/GET (LIBFMCG/GR) - fetch the library,

/LIBRARY (LIBFMCG) - indicate library to the loader

The user's program should declare all arguments appearing in the call statement and initialize required variables.

## V. LIST OF ARGUMENTS

The arguments appear in subroutine FMCG1 which is the entry of the package. The subroutine call is

CALL FMCG1 (FUNCT, GRAD, N, M, X, EPS, C1, ITERM, W, LW, ICH, IFLAG, IPR)

The arguments are as follows

FUNCT — is the name of a subroutine supplied by the user to calculate the values of the objective function at the point **x**. It must have the form

SUBROUTINE FUNCT (N,M, X,F,GD).

GRAD — is the name of a subroutine supplied by the user to calculate the gradient of the objective function at the point **x**. It must have the form

SUBROUTINE GRAD (N,M, X,GD, G)

Note: the two names FUNCT and GRAD must appear in an EXTERNAL statement in the segment calling FMCG1.

N — is an INTEGER argument which must be set to n, the number of optimization parameters. Its value must be positive and it is not changed by the package.

M — is an INTEGER argument. It is the length of the array GD which is defined by the user and is not changed by the package.

X — is a REAL array of the length at least N which on entry must be set to the initial approximation of the solution, $X(I) = x_I^0$. On exit X contains the best solution found by the package.

GD — is a REAL array of length M. It is used to transfer the common mathematical expressions from subroutine FUNCT to subroutine GRAD, hence avoiding repetitions in calculations.

EPS — is a REAL variable which on entry must be set to the required accuracy of the solution. The iteration terminates when the norm of the gradient of the objective function is less than EPS. It is not changed by the package.

C1 — is a REAL variable that determines the limiting angle between the search direction and the steepest descent direction at each iteration. Its value is between 0 and 1. It is supplied by the user and is not changed by the package.

ITERM — is an INTEGER variable which must be set to an upper bound on the number of iterations performed to reach the minimum. It is supplied by the user and is not changed by the package.

W — is a REAL array which is used for working space. Its length is given by LW.

LW — is an INTEGER argument which must be set to the length of W. Its value must be at least

$$LW = 19 * N + M.$$

ICH — is an INTEGER argument which must be set to the unit number (or channel number) that is to be used for the printed output generated by the package. Usually it is the unit number of the file OUTPUT.

IFLAG — is an INTEGER variable which on exit contains information about the solution:

IFLAG = $-1$ incorrect input arguments,

IFLAG = 0 required accuracy obtained,

IFLAG = 1 limit of the number of iterations is reached.

IPR is an INTEGER argument which controls the printed output generated by the package.

IPR = 0 the printed output of the package is supressed,

IPR = 1 the point **x**, the objective function and its gradients are printed at each iteration.

IPR = 2 the point **x**, the objective function and its gradients are printed only at the starting point and the last iteration.

## VI. EXAMPLES

Example 1 (TRIDIA Test Problem [2])

Minimize

$$F(\mathbf{x}) = \sum_{i=2}^{n} [i(2x_i - x_{i-1})^2] .$$

We consider the starting point

$$\mathbf{x}^0 = [1 \ 1 \ \ldots \ 1]^T .$$

Two solutions are shown for

a) $n = 10$

b) $n = 20$.

This example demonstrates the quadratic termination property of the package when applied to quadratic functions. For the required accuracy $10^{-4}$ and $n = 10$, the solution is obtained after 9 iterations. Also, for the same accuracy and $n = 20$, the solution is obtained after 19 iterations. We can also see from the listing of the program how the array GD is used to reduce the computational effort by transferring the common mathematical expressions from subroutine FUNCT to subroutine GRAD, hence avoiding repetitions in calculations. Its size M is variable and is defined by the user. In this example, we use $M = 9$.

```
      PROGRAM FMEX1(OUTPUT,TAPE6=OUTPUT)                                A    1
C                                                                       A    2
C     TRIDIA TEST PROBLEM                                               A    3
C                                                                       A    4
      DIMENSION X(10), W(200)                                           A    5
      EXTERNAL FUNCT,GRAD                                               A    6
      LW=200                                                            A    7
      N=10                                                              A    8
      M=9                                                               A    9
      ICH=6                                                             A   10
      EPS=1E-4                                                          A   11
      C1=1E-3                                                           A   12
      IPR=2                                                             A   13
      ITERM=45                                                          A   14
      DO 10 I=1,N                                                       A   15
      X(I)=1.                                                           A   16
10    CONTINUE                                                          A   17
      CALL FMCG1 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,W,LW,ICH,IFLAG,IPR)     A   18
      STOP                                                              A   19
      END                                                               A   20
C                                                                       B    1
C                                                                       B    2
      SUBROUTINE FUNCT (N,M,X,GD,F)                                     B    3
      REAL X(N),GD(M)                                                   B    4
      P=0.                                                              B    5
      DO 10 I=2,N                                                       B    6
      J=I-1                                                             B    7
      GD(J)=2.*X(I)-X(I-1)                                              B    8
      P=P+1*(GD(J)**2)                                                  B    9
10    CONTINUE                                                          B   10
      F=P                                                               B   11
      RETURN                                                            B   12
      END                                                               B   13
C                                                                       C    1
C                                                                       C    2
      SUBROUTINE GRAD (N,M,X,GD,G)                                      C    3
      REAL X(N),G(N),GD(M)                                              C    4
      MM=N-1                                                            C    5
      DO 10 I=2,MM                                                      C    6
      J=I-1                                                             C    7
      G(I)=4.*1*GD(J)-2.*(I+1.)*GD(J+1)                                 C    8
10    CONTINUE                                                          C    9
      G(1)=-4.*GD(1)                                                    C   10
      G(N)=4.*N*GD(MM)                                                  C   11
      RETURN                                                            C   12
      END                                                               C   13
```

UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCGA PACKAGE)

INPUT DATA

NUMBER OF VARIABLES (N)..... 10

ACCURACY (EPS)..... 1.000E-04

WORKING SPACE (LW)..... 200

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 5.40000E+01

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .10000E+01 | -.40000E+01 |
| 2 | .10000E+01 | .20000E+01 |
| 3 | .10000E+01 | .40000E+01 |
| 4 | .10000E+01 | .60000E+01 |
| 5 | .10000E+01 | .80000E+01 |
| 6 | .10000E+01 | .10000E+02 |
| 7 | .10000E+01 | .12000E+02 |
| 8 | .10000E+01 | .14000E+02 |
| 9 | .10000E+01 | .16000E+02 |
| 10 | .10000E+01 | .18000E+02 |

SOLUTION

OBJECTIVE FUNCTION : 2.44115E-26

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .14985E+01 | -.24869E-13 |
| 2 | .74927E+00 | .15586E-13 |
| 3 | .37463E+00 | .68012E-13 |
| 4 | .18732E+00 | -.12985E-12 |
| 5 | .93659E-01 | .40494E-12 |
| 6 | .46829E-01 | -.81448E-12 |
| 7 | .23415E-01 | .12559E-11 |
| 8 | .11707E-01 | -.10321E-11 |
| 9 | .58537E-02 | .11394E-11 |
| 10 | .29268E-02 | .23055E-12 |

TYPE OF SOLUTION (IFLAG)..... 0

NUMBER OF ITERATIONS ..... 9

EFFECTIVE FUNCTION EVALUATIONS (EFE) ..... 120

EXECUTION TIME (IN SECONDS)..... .076

UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCGA PACKAGE)

INPUT DATA

NUMBER OF VARIABLES (N)..... 20

ACCURACY (EPS)..... 1.000E-04

WORKING SPACE (LW)..... 400

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 2.09000E+02

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .10000E+01 | -.40000E+01 |
| 2 | .10000E+01 | .20000E+01 |
| 3 | .10000E+01 | .40000E+01 |
| 4 | .10000E+01 | .60000E+01 |
| 5 | .10000E+01 | .80000E+01 |
| 6 | .10000E+01 | .10000E+02 |
| 7 | .10000E+01 | .12000E+02 |
| 8 | .10000E+01 | .14000E+02 |
| 9 | .10000E+01 | .16000E+02 |
| 10 | .10000E+01 | .18000E+02 |
| 11 | .10000E+01 | .20000E+02 |
| 12 | .10000E+01 | .22000E+02 |
| 13 | .10000E+01 | .24000E+02 |
| 14 | .10000E+01 | .26000E+02 |
| 15 | .10000E+01 | .28000E+02 |
| 16 | .10000E+01 | .30000E+02 |
| 17 | .10000E+01 | .32000E+02 |
| 18 | .10000E+01 | .34000E+02 |
| 19 | .10000E+01 | .36000E+02 |
| 20 | .10000E+01 | .80000E+02 |

SOLUTION

OBJECTIVE FUNCTION : 3.11268E-15

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .15000E+01 | -.95932E-11 |
| 2 | .75000E+00 | .40555E-10 |
| 3 | .37500E+00 | -.58839E-10 |
| 4 | .18750E+00 | .31390E-10 |
| 5 | .93750E-01 | .47737E-11 |
| 6 | .46875E-01 | -.45511E-11 |
| 7 | .23437E-01 | -.17471E-11 |
| 8 | .11719E-01 | -.14340E-10 |
| 9 | .58594E-02 | .85192E-10 |
| 10 | .29297E-02 | -.44494E-09 |
| 11 | .14648E-02 | .19743E-08 |
| 12 | .73242E-03 | -.74905E-08 |
| 13 | .36621E-03 | .24495E-07 |
| 14 | .18311E-03 | -.69067E-07 |
| 15 | .91553E-04 | .16711E-06 |
| 16 | .45775E-04 | -.34306E-06 |
| 17 | .22890E-04 | .58511E-06 |
| 18 | .11442E-04 | -.79676E-06 |
| 19 | .57246E-05 | .79238E-06 |
| 20 | .28597E-05 | -.41983E-06 |

TYPE OF SOLUTION (IFLAG)... ..... ..... 0

NUMBER OF ITERATIONS ..... 19

EFFECTIVE FUNCTION EVALUATIONS (EFE) ..... 440

EXECUTION TIME (IN SECONDS)..... .264

Example 2 (NONDIA Test Problem [2])

Minimize

$$F(\mathbf{x}) = \sum_{i=2}^{n} [100(x_1 - x_i^2)^2 + (1 - x_2)^2] .$$

We consider the starting point

$$\mathbf{x}^0 = [-1.2 \ 1 \ \ldots \ 1]^T .$$

For n = 10 and required accuracy $10^{-4}$ the solution is obtained after 288 effective function evaluation (number of function calls + n times the number of gradient calls) which is much less than that in [2]. This example is one of the commonly used test problems to compare different conjugate gradient methods. It indicates the capability of the package to deal efficiently with non quadratic functions. See Table II in Section VII.

```
      PROGRAM FMEX2(OUTPUT,TAPE6=OUTPUT)                                A    1
C                                                                       A    2
C     NONDIA TEST PROBLEM                                               A    3
C                                                                       A    4
      DIMENSION X(10), W(200)                                           A    5
      EXTERNAL FUNCT,GRAD                                               A    6
      LW=200                                                            A    7
      N=10                                                              A    8
      M=10                                                              A    9
      EPS=1.E-4                                                         A   10
      ICH=6                                                             A   11
      C1=1E-3                                                           A   12
      IPR=2                                                             A   13
      ITERM=80                                                          A   14
      X(1)=-1.2                                                         A   15
      DO 10 I=2,N                                                       A   16
      X(I)=1.                                                           A   17
10    CONTINUE                                                          A   18
      CALL FMCG1 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,W,LW,ICH,IFLAG,IPR)     A   19
      STOP                                                              A   20
      END                                                               A   21
C                                                                       B    1
C                                                                       B    2
      SUBROUTINE FUNCT (N,M,X,GD,F)                                     B    3
      REAL X(N),GD(M)                                                   B    4
      P=0.                                                              B    5
      GD(1)=(1.-X(2))**2                                                B    6
      DO 10 I=2,N                                                       B    7
      GD(I)=X(1)-X(I)**2                                                B    8
      P=P+100.*(GD(I)**2)+GD(1)                                         B    9
10    CONTINUE                                                          B   10
      F=P                                                               B   11
      RETURN                                                            B   12
      END                                                               B   13
C                                                                       C    1
C                                                                       C    2
      SUBROUTINE GRAD (N,M,X,GD,G)                                      C    3
      REAL X(N),G(N),GD(M)                                              C    4
      S=0.                                                              C    5
      DO 10 I=2,N                                                       C    6
      S=S+200.*GD(I)                                                    C    7
10    CONTINUE                                                          C    8
      G(1)=S                                                            C    9
      DO 20 I=3,N                                                       C   10
      G(I)=-400.*X(I)*GD(I)                                             C   11
20    CONTINUE                                                          C   12
      G(2)=-400.*X(2)*GD(2)-2.*(N-1.)*(1.-X(2))                         C   13
      RETURN                                                            C   14
      END                                                               C   15
```

UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCGA PACKAGE)

INPUT DATA

NUMBER OF VARIABLES (N)..... 10

ACCURACY (EPS)..... 1.000E-04

WORKING SPACE (LW)..... 200

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 4.35600E+03

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | -.12000E+01 | -.39600E+04 |
| 2 | .10000E+01 | .88000E+03 |
| 3 | .10000E+01 | .88000E+03 |
| 4 | .10000E+01 | .88000E+03 |
| 5 | .10000E+01 | .88000E+03 |
| 6 | .10000E+01 | .88000E+03 |
| 7 | .10000E+01 | .88000E+03 |
| 8 | .10000E+01 | .88000E+03 |
| 9 | .10000E+01 | .88000E+03 |
| 10 | .10000E+01 | .88000E+03 |

SOLUTION

OBJECTIVE FUNCTION : 2.50430E-12

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .10000E+01 | -.52664E-04 |
| 2 | .10000E+01 | -.32212E-04 |
| 3 | .10000E+01 | .17532E-04 |
| 4 | .10000E+01 | .17532E-04 |
| 5 | .10000E+01 | .17532E-04 |
| 6 | .10000E+01 | .17532E-04 |
| 7 | .10000E+01 | .17532E-04 |
| 8 | .10000E+01 | .17532E-04 |
| 9 | .10000E+01 | .17532E-04 |
| 10 | .10000E+01 | .17532E-04 |

TYPE OF SOLUTION (IFLAG)..... 0

NUMBER OF ITERATIONS ..... 22

EFFECTIVE FUNCTION EVALUATIONS (EFE) ..... 288

EXECUTION TIME (IN SECONDS)..... .174

Example 3 (EXP2 Test Problem [2])

Minimize

$$F(\mathbf{x}) = \sum_{i=1}^{10} (e^{-x_1 z_i} - 5e^{-x_2 z_i} - e^{-z_i} + 5e^{-10 z_i})^2 ,$$

where

$$z_i = i/10.$$

We consider the starting point

$$\mathbf{x}^0 = [1 \;\; 2]^T.$$

This problem was introduced by Biggs [2]. More details and different methods of solution are given in [5]. The solution is obtained after 60 effective function evaluation which is quite comparable to other conjugate gradient methods as we can see from Table II. However, it indicates that the results for small dimensional problems are less interesting than the results obtained for larger dimensional problems.

```
      PROGRAM FMEX3(OUTPUT,TAPE6=OUTPUT)                                    A    1
C                                                                           A    2
C     BIGGS TEST PROBLEM                                                    A    3
C                                                                           A    4
      DIMENSION X(2), W(50)                                                 A    5
      EXTERNAL FUNCT,GRAD                                                   A    6
      LW=50                                                                 A    7
      N=2                                                                   A    8
      M=10                                                                  A    9
      ICH=6                                                                 A   10
      EPS=1.E-4                                                             A   11
      C1=1E-3                                                               A   12
      IPR=2                                                                 A   13
      ITERM=60                                                              A   14
      X(1)=1.                                                               A   15
      X(2)=2.                                                               A   16
      CALL FMCG1 (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,W,LW,ICH,IFLAG,IPR)         A   17
      STOP                                                                  A   18
      END                                                                   A   19
C                                                                           B    1
C                                                                           B    2
      SUBROUTINE FUNCT (N,M,X,GD,F)                                         B    3
      REAL X(N),GD(M)                                                       B    4
      P=0.                                                                  B    5
      DO 10 I=1,10                                                          B    6
      ZI=I/10.                                                              B    7
      GD(I)=(EXP(-X(1)*ZI)-5.*EXP(-X(2)*ZI)-EXP(-ZI)+5.*EXP(-10.*ZI))       B    8
      P=P+GD(I)**2                                                          B    9
10    CONTINUE                                                              B   10
      F=P                                                                   B   11
      RETURN                                                                B   12
      END                                                                   B   13
C                                                                           C    1
C                                                                           C    2
      SUBROUTINE GRAD (N,M,X,GD,G)                                          C    3
      REAL X(N),G(N),GD(M)                                                  C    4
      R=0.                                                                  C    5
      S=0.                                                                  C    6
      DO 10 I=1,10                                                          C    7
      ZI=I/10.                                                              C    8
      H=2.*ZI*GD(I)                                                         C    9
      R=R-EXP(-X(1)*ZI)*H                                                   C   10
      S=S+5.*EXP(-X(2)*ZI)*H                                                C   11
10    CONTINUE                                                              C   12
      G(1)=R                                                                C   13
      G(2)=S                                                                C   14
      RETURN                                                                C   15
      END                                                                   C   16
```

UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCG PACKAGE)

INPUT DATA

NUMBER OF VARIABLES (N)..... 2

ACCURACY (EPS)..... 1.000E-04

WORKING SPACE(LW)..... 50

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 3.22626E+01

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .10000E+01 | .83082E+01 |
| 2 | .20000E+01 | -.25326E+02 |

SOLUTION

OBJECTIVE FUNCTION : 7.32688E-11

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .10000E+01 | -.34066E-06 |
| 2 | .10000E+02 | -.39134E-05 |

TYPE OF SOLUTION (IFLAG)..... 0

NUMBER OF ITERATIONS ..... 14

EFECTIVE FUNCTION EVALUATION (EFE) ..... 60

EXECUTION TIME (IN SECONDS)..... .1860

Example 4 [10, Example 2]

The problem is to solve a system of two nonlinear equations

$$4(x_1 + x_2) = 0,$$

$$(x_1 - x_2)((x_1 - 2)^2 + x_2^2) + 3x_1 + 5x_2 = 0.$$

The solution can be obtained by minimizing the least squares objective function

$$F(\mathbf{x}) = \mathbf{f}^T\mathbf{f},$$

where

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x})]^T,$$

$$f_1(\mathbf{x}) = 4(x_1 + x_2)$$

and

$$f_2(\mathbf{x}) = (x_1 - x_2)((x_1 - 2)^2 + x_2^2) + 3x_1 + 5x_2.$$

The solutions are shown for two different starting points $\mathbf{x}^0$

$$[-2 \quad -2]^T, \quad [2 \quad 0]^T.$$

```
      PROGRAM FMEX4(OUTPUT,TAPE6=OUTPUT)                                A    1
C                                                                       A    2
C     BRENT TEST PROBLEM                                                A    3
C                                                                       A    4
      DIMENSION X(2), W(50)                                             A    5
      EXTERNAL FUNCT,GRAD                                               A    6
      LW=50                                                             A    7
      ICH=6                                                             A    8
      N=2                                                               A    9
      M=4                                                               A   10
      IPR=2                                                             A   11
      EPS=1.E-6                                                         A   12
      C1=1.E-3                                                          A   13
      ITERM=30                                                          A   14
      X(1)=2.                                                           A   15
      X(2)=0.                                                           A   16
      CALL FMCG1  (FUNCT,GRAD,N,M,X,EPS,C1,ITERM,W,LW,ICH,IFLAG,IPR)    A   17
      STOP                                                              A   18
      END                                                               A   19
C                                                                       B    1
C                                                                       B    2
      SUBROUTINE FUNCT (N,M,X,GD,F)                                     B    3
      REAL X(N),GD(M)                                                   B    4
      GD(1)=X(1)+X(2)                                                   B    5
      GD(2)=X(1)-X(2)                                                   B    6
      GD(3)=(X(1)-2.)**2+X(2)*X(2)                                      B    7
      GD(4)=GD(2)*GD(3)+3.*X(1)+5.*X(2)                                 B    8
      F=16.*(GD(1)**2)+GD(4)**2                                         B    9
      RETURN                                                            B   10
      END                                                               B   11
C                                                                       C    1
C                                                                       C    2
      SUBROUTINE GRAD (N,M,X,GD,G)                                      C    3
      REAL X(N),GD(M),G(N)                                              C    4
      R1=32.*GD(1)                                                      C    5
      H1=2.*GD(2)                                                       C    6
      H2=2.*GD(4)                                                       C    7
      R2=3.+H1*(X(1)-2.)+GD(3)                                          C    8
      R3=5.+H1*X(2)-GD(3)                                               C    9
      G(1)=R1+H2*R2                                                     C   10
      G(2)=R1+H2*R3                                                     C   11
      RETURN                                                            C   12
      END                                                               C   13
```

**UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCGA PACKAGE)**

INPUT DATA

NUMBER OF VARIABLES (N)..... 2

ACCURACY (EPS)..... 1.000E-06

WORKING SPACE (LW)..... 40

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 5.12000E+02

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | -.20000E+01 | -.86400E+03 |
| 2 | -.20000E+01 | .35200E+03 |

SOLUTION

OBJECTIVE FUNCTION : 4.93869E-15

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | -.88305E-08 | -.61420E-07 |
| 2 | .23621E-07 | .39690E-06 |

TYPE OF SOLUTION (IFLAG)..... 0

NUMBER OF ITERATIONS ..... 11

EFFECTIVE FUNCTION EVALUATIONS (EFE) ..... 51

EXECUTION TIME (IN SECONDS)..... .031

UNCONSTRAINED MINIMIZATION BY CONJUGATE GRADIENTS (FMCGA PACKAGE)

INPUT DATA

NUMBER OF VARIABLES (N)..... 2

ACCURACY (EPS)..... 1.000E-06

WORKING SPACE (LW)..... 50

PRINTOUT CONTROL (IPR)..... 2

STARTING POINT. OBJECTIVE FUNCTION : 1.00000E+02

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | .20000E+01 | .10000E+03 |
| 2 | 0. | .12400E+03 |

SOLUTION

OBJECTIVE FUNCTION : 1.01826E-22

| | VARIABLES | GRADIENT |
|---|---|---|
| 1 | -.10904E-11 | -.16141E-09 |
| 2 | -.42751E-12 | -.64693E-10 |

TYPE OF SOLUTION (IFLAG)..... 0

NUMBER OF ITERATIONS ..... 8

EFFECTIVE FUNCTION EVALUATIONS (EFE) ..... 37

EXECUTION TIME (IN SECONDS)..... .025

## VII. COMPARISON OF THE ALGORITHMS

It is generally accepted that the primary criterion in evaluating different algorithms must be whether or not the algorithm can solve most of the problems posed, that is, is the algorithm robust?. It is obvious that any algorithm can be defeated by a suitably designed problem; moreover, we cannot realistically ask that an algorithm can solve all possible problems. However, the robustness of the algorithm is always of primary concern. A second criterion is the value of the objective function $F(\mathbf{x})$ and the elements of the vector of variables $\mathbf{x}$ at the mininum. This usually depends upon the termination criteria used to end the computation. Assuming that uniform termination criteria are adopted, a third criterion of the effectiveness of an algorithm is the number of function and gradient evaluations to reach the desired precision in $F(\mathbf{x})$ and $\mathbf{x}$. Certainly this criterion is better than the number of iterations which varies quite widely from algorithm to algorithm. As a matter of fact, the last criterion has been widely used by different authors in comparing different conjugate gradient codes. For the sake of comparison, we have quoted the results presented in [2] for a different number of algorithms.

For each algorithm, the amount of computational effort is measured in terms of the effective function evaluation (EFE) given by $EFE = N_F + nN_G$, where

$n$ = number of unknowns

$N_F$ = number of function calls

$N_G$ = number of gradient calls.

In all cases, the termination criterion was set at

$$\mathbf{g}^T \mathbf{g} < \varepsilon_2 = 10^{-4}$$

and the constant $C_1$ in Step 7 and Step 9 was set at $10^{-3}$. It is quite clear from the results presented in Table II that, in most of the cases, the FMCGA package is indeed an improvement over existing available codes, especially for large dimensional probelms.

It is our numerical experience that the values of $C_1$ ranging from 0.03 to 0.001 have been found to be the best. This actually means that the maximum allowable angle between

the conjugate search direction and the steepest descent direction at each iteration is from 88° to 89.9°. On the other hand, for larger values of $C_1$ (e.g. above 0.5), the convergence may be affected since the conjugate search directions generated in this case will be approximately similar to the steepest descent directions. Also, it is not recommended to increase the accuracy beyond $10^{-10}$. Besides the fact that this is a reasonable accuracy in almost all practical problems, the computational effort expressed in terms of the number of iterations and function evaluations is increased significantly.

TABLE II

COMPARISON OF EFFECTIVE FUNCTION EVALUATIONS (EFE†)

| Program†† | TRIDIA n = 10 | TRIDIA n = 20 | NONDIA n=10 | EXP2 n = 2 |
|---|---|---|---|---|
| FMCGA | 120 | 440 | 288 | 60 |
| BLALN | 209 | 819 | 858 | - |
| VA14A | 297 | 819 | 935 | 45 |
| VA08A | 440 | 1029 | 1100 | 156 |
| OPCG2 | 119 | 439 | 702 | 59 |
| CONGRA | - | - | - | 84 |

† EFE = Number of function calls + n times the number of gradient calls.

†† BLALN = Buckley and Le Nir [5] method
VA14A = Harwell subroutines [4]
VA08A = Harwell subroutines [4]
OPCG2 = Dixon et al. [2] method
CONGRA = Fletcher and Reeves [1]

## VIII. CONCLUSIONS

A new computer package implementing the new three term conjugate gradient method has been fully presented and discussed. Quite apart from its simplicity and low storage requirement, the package performance results compare favourably with existing codes. A new, efficient line search strategy is employed. Different numerical examples strongly support the use of FMCGA in unconstrained minimization.

## IX. REFERENCES


[1] L. Nazareth, "A conjugate gradient algorithm without line searches", J. Optimization Theory and Applications, vol. 23, 1977, pp. 373-387.

[2] L.C.W. Dixon, P.G. Ducksbury and P. Singh, "A new three term conjugate gradient method", Numerical Optimization Centre, Hatfield Polytechnic, England, Technical Report No. 130, 1983.

[3] J.W. Bandler and M.A. El-Gamal, "FMCGA - A Fortran package for unconstrained minimization by conjugate gradients", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-84-7-L, 1984.

[4] M.J.D. Powell, "Restart procedures for the conjugate gradient method", Mathematical Programming, vol. 12, 1977, pp. 241-254.

[5] L.C.W. Dixon, "Conjugate directions without line searches", Numerical Optimization Center, Hatfield Polytechnic, England, Technical Report No. 38, 1973.

[6] P. Wolfe, "Convergence conditions for ascent methods", SIAM Review, vol. 11, 1969, pp. 226-235.

[7] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivative", Pacific Journal of Mathematics, vol. 16, 1966, pp. 1-3.

[8] R. Fletcher and C. Reeves, "Function minimization by conjugate gradients", Computer J., vol. 17, 1964, pp. 149-154.

[9] L.C.W. Dixon, "On Nazareth's three term conjugate gradient method", Numerical Optimization Centre, Hatfield Polytechnic, England, Technical Report No. 133, 1983.

[10] J.W. Bandler and W.M. Zuberek, "MFNC - A Fortran package for minimization with general constraints", Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, Report SOS-82-6-U/L, 1983.