

A STUDY OF THE GENERALIZED
REDUCED GRADIENT METHOD

by

YOUSEF SHAFII

B.S., TEHRAN POLYTECHNIC, TEHRAN, IRAN, 1961

A MASTER'S REPORT

Submitted in partial fulfillment of the
requirement for the degree

MASTER OF SCIENCE

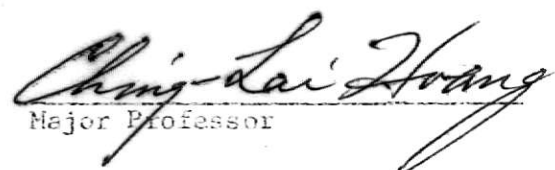
Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1973

Approved by


Major Professor

LD
2668
R4
1973
S5
C.2

ACKNOWLEDGMENTS

Doc. The author wishes to express his sincere appreciation to Dr. C. L. Hwang, major professor, for his encouragement and many helpful suggestions and criticisms during the preparation of this report.

The author extends his gratitude to the members of his graduate committee, Dr. F. A. Tillman, Dr. L. T. Fan, and Dr. D. L. Grosh for their guidance and suggestions.

Thanks are due to Mrs. Marie Jirak for her excellent typing of this report.

TABLE OF CONTENTS

	page
Acknowledgement	ii
General Introduction	1
CHAPTER 1 Introduction to Generalized Reduced Gradient Method	
1.1 Introduction	4
1.2 Statement of problem	4
1.3 Basic principle of the reduced gradient	5
1.4 Basic outline of the GRG algorithm	7
CHAPTER 2 Introduction to the GREG Double Precision Program	
2.1 Introduction	11
2.2 Double precision version of GREG	11
2.3 Basic principle of GREG program	12
2.4 Main program and nine related internal subroutines	15
2.5 GREG, four external subroutines	20
2.6 Input data for the GREG double precision	32
2.7 Job control language for utilizing the GREG double precision program	37
2.8 The GREG double precision output	37
CHAPTER 3 Numerical Testing of the GREG Double Precision Program	
3.1 Introduction	52
3.2 Problem number one	53
3.3 Problem number two	54
3.4 Problem number three	55
3.5 Problem number four	56
3.6 Problem number five	57

3.7 Concluding remarks	61
References	63
Appendix 1 A brief summary of GREG algorithm	66
Appendix 2 GREG programming material for chapter 3	88

GENERAL INTRODUCTION

Linear programming [8] has long been accepted technique. For the last twenty-five years, the areas of applications have been continuously expanding, and increasingly sophisticated and powerful computer codes have been written.

Nevertheless, the real word is seldom linear, this fact plainly justifies the need for nonlinear programming. Without going back as far as Lagrange and Cauchy, the pioneering works of R. Courant (who created the penalty method approach for solving nonlinear optimization problems), Fritz John, H. W. Kuhn and A. W. Tucker (who set theoretical background) [15, 16], are classical vintages aged twenty years or more. For the last ten years, a number of numerical methods have been proposed for nonlinear programming under various hypotheses (convexity or continuity, or differentiability of some or all the functions involved). During the last five years, there has been an explosion of knowledge, and recently some thirty methods, among others, have been compared on computers via test problems [2].

The development of nonlinear programming, however, has been hampered by theoretical difficulties. The problems of nonconvex sets and local optima are encountered and prevalent. Consequently, the coding of existing algorithms has been limited to special cases for which the behavior of the functions are predictable. Indeed, an algorithm superior to all algorithms is probably nonexistent [13].

The purpose of this report is to study one of the existing and apparently efficient computer coded nonlinear programming algorithms, the generalized reduced gradient method. The generalized reduced gradient method (GRG) which has been proposed by Abadie and Carpentier [3,4,5] is an elaborate extension of the Wolfe reduced gradient method [18,19].

The Wolfe reduced gradient method solves a problem with a nonlinear objective function and linear (equality) constraints. It classes the variables as independent and dependent variables. This essentially reduces the original problem to an unconstrained problem of reduced dimension. Applying the same concept to a problem with a nonlinear set of constraints complicates matters, but it is possible by using numerical methods.

The GRG method has been studied extensively and coded in FORTRAN by Abadie [1], Abadie and Guigou [6] and Guigou [11,12]. The first was an experimental code called GRG 66 which was followed by the second code, GRG 69. An improved code, GREG, is the outgrowth of the first two codes and promises to remain among the highly regarded nonlinear programming procedures.

The original GREG code has been written in FORTRAN and developed on the CDC 6600 computer. Later, the program has modified to FORTRAN single precision code for IBM system 360/50 by Willims [17]. Because of an important need of solving some nonlinear programming problems with more accurate results the GREG code has been changed to FORTRAN double precision in this report.

Because of the self contant reason, the GREG theory is reviewed in chapter one. This is followed by chapter two describing the GREG code and introducing the usage of GREG double precision code. In chapter three a series of numerical example is solved by GREG double precision code and the two version of GREG single and double presion are compared. And, finally a brief outline of GREG algorithm is given in Appendix one along with a numerical example.

CHAPTER ONE

INTRODUCTION TO GENERALIZED REDUCED GRADIENT METHOD

1.1 INTRODUCTION

Before the development of the Generalized Reduced Gradient algorithm, several nonlinear programming methods were able to solve only special cases. In 1963 and 1967 Wolfe [18,19] developed an algorithm which was able to solve problems with linear constraints and nonlinear objective function. The extension of Wolfe's method in order to solve a general form of nonlinear programming problem, with the name of Generalized Reduced Gradient (GRG) has been proposed by Abadie and Carpentier [3,4,5] in 1967.

The GRG method has been coded in FORTRAN by Abadie [1], Abadie and Guigou [6,7] and Guigou [11,12]. The first two were an experimental codes called GRG66 and GRG69. The third code (GREG) has been written with much better results in computing time as well as in accuracy. This code (GREG) permits the solution of mathematical programs with linear or nonlinear objective function, under constraints which can be linear or nonlinear. The domain of application of this code is quite vast, but its main purpose is the solution of problems having nonlinear constraints.

1.2 STATEMENT OF PROBLEM

Any mathematical programming problem may be stated as follows:

Maximize (1.1)

$$f_0(\bar{X})$$

subject to

$$f_i(\bar{X}) = 0, \quad i = 1, \dots, m$$

$$\bar{a} \leq \bar{X} \leq \bar{b}$$

Where $\bar{X}$, $\bar{a}$, $\bar{b}$ are n dimensional column vectors.

Both $f_0(\bar{X})$ and $f_i(\bar{X})$, [$i = 1, \dots, m$] are assumed continuously differentiable functions.

Note that any inequality constraint, by adding a slack variable, can be written in the preceeding standard form.

1.3 BASIC PRINCIPLE OF THE REDUCED GRADIENT

The GRG method is based on transforming a constrained optimization problem into an unconstrained problem. This is accomplished by dividing the vector $\bar{X}$ into $\bar{x}$ and $\bar{y}$, where $\bar{y}$ represents the dependent variables (basis) with m components, while $\bar{x}$, represents the independent variables with $(n-m)$ components. Thus the objective function can be written as:

$$f_0(\bar{X}) = f_0(\bar{x}, \bar{y}) = f_0[\bar{x}, \bar{\phi}(\bar{x})] = F(\bar{x}) \quad (1.2)$$

Now, the problem has been changed to maximizing $F(\bar{x})$ subject to the boundaries of the variables.

The reduced gradient can be determined as:

$$\frac{\partial F}{\partial \bar{x}} = \frac{\partial f_0}{\partial \bar{x}} + \frac{\partial f_0}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \bar{x}} \quad (1.3)$$

Where $\frac{\partial f_0}{\partial \bar{x}}$ and $\frac{\partial f_0}{\partial \bar{y}}$ designate the row-vectors of the partial derivatives of f_0 with respect of the $\bar{x}$ and $\bar{y}$. Thus $\frac{\partial \bar{y}}{\partial \bar{x}}$ represents a matrix of $(n-m)$ by m components, where each row of the matrix designates the partial derivatives of $\bar{y} = \bar{\phi}(\bar{x})$ with respect to the independent variables. For determining the $\frac{\partial \bar{y}}{\partial \bar{x}}$ we do as follows. Setting

$$\bar{f}(\bar{X}) = \bar{f}(\bar{x}, \bar{y}) = 0$$

gives

$$\frac{\partial \bar{f}}{\partial \bar{x}} + \frac{\partial \bar{f}}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \bar{x}} = 0,$$

so that

$$\frac{\partial \bar{y}}{\partial \bar{x}} = - \left[\frac{\partial \bar{f}}{\partial \bar{y}} \right]^{-1} \left[\frac{\partial \bar{f}}{\partial \bar{x}} \right]. \quad (1.4)$$

By substituting (1.4) into (1.3) the reduced gradient becomes

$$\frac{\partial F}{\partial \bar{x}} = \frac{\partial f_0}{\partial \bar{x}} - \frac{\partial f_0}{\partial \bar{y}} \left[\frac{\partial \bar{f}}{\partial \bar{y}} \right]^{-1} \left[\frac{\partial \bar{f}}{\partial \bar{x}} \right] \equiv g^{-T} \quad (1.5)$$

Where $\left[\frac{\partial \bar{f}}{\partial \bar{y}} \right]^{-1}$ is the inverse of the matrix which the partial derivatives of the constraints with respect to the dependent variables. Similarly

$\frac{\partial \bar{f}}{\partial \bar{x}}$ designates the matrix of partial derivatives of the constraints with respect to the independent variables.

1.4 BASIC OUTLINE OF THE GRG ALGORITHM

Assume that point $\bar{x}^0$ as a starting point can be dividable to $\bar{x}$ and $\bar{y}$ such that:

$$a_i < y_i < b_i, \quad i = 1, \dots, m,$$

$$\frac{\partial \bar{f}}{\partial \bar{y}} \quad \text{is non singular.}$$

Then we have the following algorithm:

Step 1.

Compute the direction of movement ($\bar{h}^0$) for independent variables ($\bar{x}$), by the following three substeps.

Step 1.1

Compute the reduced gradient according to (1.5)

$$\bar{g}^0 = \frac{\partial f_0}{\partial \bar{x}} - \frac{\partial f_0}{\partial \bar{y}} \left[\frac{\partial \bar{f}}{\partial \bar{y}} \right]^{-1} \left[\frac{\partial \bar{f}}{\partial \bar{x}} \right].$$

Step 1.2

Compute the projected reduced gradient ($\bar{p}^0$) as follows:

$$\bar{p}_i^0 = \begin{cases} 0 & \text{if } x_i^0 = a_i \text{ and } g_i^0 \leq 0 \\ 0 & \text{if } x_i^0 = b_i \text{ and } g_i^0 \geq 0 \\ g_i^0 & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (n-m)$$

Step 1.3

Compute the modified projected reduced gradient ($\bar{h}^0$).

At this step we are computing the direction of movement for independent variables. In the simplest case $\bar{h}^0 = \bar{p}^0$, but it may be modified by one or both of the following devices where $\bar{h}^0 \cdot \bar{p}^0 > 0$:

a) By constructing all components such that

$p_i^0(x_i - a_i)$ or $p_i^0(x_i - b_i)$ is positive and very small, or even equal to zero.

b) Using conjugate direction (Fletcher and Powell [10] or Fletcher and Reeves [9]).

Step 2.

At this step we know the direction of independent variables, and therefore it is possible to compute a set of new values for independent variables with respect of an unknown parameter θ .

$$\bar{x}^0 + \theta \bar{h}^0 \quad (1.6)$$

Similarly by knowing the direction of movement for dependent variables, we are able to compute a set of new values for the dependent variables. The following formula give us the direction of movement for the dependent variables.

$$\bar{k}^0 = - \left[\frac{\partial \bar{f}}{\partial \bar{y}} \right]^{-1} \left[\frac{\partial \bar{f}}{\partial \bar{x}} \right] \bar{h}^0 \quad (1.7)$$

and therefore,

$$\bar{y}^0 + \theta \bar{k}^0 \quad (1.8)$$

are the new calculated values for dependent variables.

Theta, θ , could be calculated by maximizing the objective function with respect to the new value of the variables. This could be accomplished by one of the one-dimensional search techniques such as Golden Section, Fibonacci search or quadratic interpolation.

Step 3.

Obtain the new values for the independent variables with respect to the following modification.

$$x_i^1 = \begin{cases} a_i & \text{if } x_i^0 + \theta h_i^0 < a_i \\ b_i & \text{if } x_i^0 + \theta h_i^0 > b_i \\ x_i^0 + \theta h_i^0 & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (n-m)$$

Step 4.

Compute a feasible $\bar{x}^1$ corresponding to θ , solve the system of m equation with respect to the m unknown $\bar{y}$.

$$f(\bar{x}^1, \bar{y}^1) = 0$$

This may be done by an iterative method. If no speedy convergence is observed then, decrease θ (e.g. set $\theta = \frac{1}{2}\theta$), and go to step 3 with the same $\bar{h}^0$. Otherwise let $\bar{y}^1$ be the solution obtained. If $f_0(\bar{x}^1, \bar{y}^1) > f_0(\bar{x}^0, \bar{y}^0)$ go to step 5, otherwise, decrease θ to $\theta = \frac{1}{2}\theta$ and go to step 3 with the same $\bar{h}^0$.

Step 5.

Set $\bar{X}^0 = \bar{X}^1$ and repeat the algorithm. But if for some reason a better value for θ may be determined by previous information, a return to step 3 is possible.

The details of the above procedures are presented in [14].

CHAPTER TWO

INTRODUCTION TO THE GREG DOUBLE PRECISION

2.1 INTRODUCTION

GREG is a computer program which has been developed by GUIGOU of Electricite de France [12]. The main objective of this program is to solve mathematical programming problems with linear or nonlinear objective function under nonlinear constraints. The GREG is based on Generalized Reduced Gradient method; and consists of a main program, nine internal subroutines, and four user supplied, external subroutines. The single precision version of this program has been compiled and stored in partitioned data set by Williams [17]. The single precision version required approximately 120K (K = 1024 bytes) of storage.

2.2 DOUBLE PRECISION VERSION OF GREG

Because of an important need for solving some of the nonlinear programming problems with more accurate results, the original GREG program has been completely changed to the FORTRAN double precision code. This has been accomplished as follows:

1. All of the real variables have been declared to the double precision by listing them in the DOUBLE PRECISION statements.
2. All of the arithmetic statements have been changed to the double precision arithmetic statements.

3) All of the mathematical function which have been used in the program changed to the double precision standard.

4) The input and output formats have been changed to the double precision code.

The exact number of digits in the double precision form varies from one computer to another, but it is ordinarily more than twice the number of digits in single precision. Double precision may contain a maximum of 16 significant digits in most versions of FORTRAN IV.

The GREG main program and nine related internal subroutines in the double precision form have been compiled and stored in partitioned data set. This step reduces the CPU time, and consequently the cost. It also allows the program to be run with less than 128 K of storage.

2.3 BASIC PRINCIPLE OF GREG PROGRAM

Any mathematical programming problem can be written in the following form:

$$\text{Maximize } f_0(\bar{X}) \quad (2.1)$$

$$\text{Subject to } \bar{f}(\bar{X}) \begin{matrix} \geq \\ \leq \end{matrix} 0$$

$$\bar{a} \leq \bar{X} \leq \bar{b}$$

Where the number of components of $\bar{X}$ is equal to the number of primal variables.

In order to solve this mathematical programming problem by the Generalized Reduced Gradient method, the objective function and all of the constraints must be continuously differentiable functions of vector $\bar{X}$.

By the introducing of slack and artificial variables, the formula (2.1) may be written as follows.

$$\text{Maximize } \phi_0(\bar{X}) \quad (2.2)$$

$$\text{Subject to } \bar{\phi}(\bar{X}) = 0$$

$$\bar{a} \leq \bar{X} \leq \bar{b}$$

where in this case the number of components of $\bar{X}$ is equal to the number of primal variables plus the number of slack and artificial variables which have been generated.

The first treatment in GREG FORTRAN program is changing the form of the problem from formula (2.1) to formula (2.2).

Let

NV = The number of primal variables.

NIN = The number of inequality constraints.

NEG = The number of equality constraints.

NC = NIN + NEG = Total number of constraints.

The GREG FORTRAN program automatically provides slack and artificial variables. It provides NIN slack variables in order to change the inequality constraints to equality type constraints. It also computes the value of each constraint with respect to the given starting point $\bar{X}^0$, and if any constraint does not satisfied by the starting point an artificial

variable automatically will be added to the constraint, the same artificial variable being supplied with an appropriate penalty attached to the objective function.

Let

NIN1 = The number of inequality constraints not satisfied.

NIN2 = The number of inequality constraints satisfied.

NIN4 = The number of equality constraints not satisfied.

Therefore,

Total number of slack variables = NIN = NIN1 + NIN2

Total number of artificial variables = NIN1 + NIN4

Then the total number of variables including the primal and added variables is:

$$NTV = NV + 2(NIN1) + NIN2 + NIN4$$

These additional variables are generated by the internal subroutine CALC. CALC also computes the penalty for the artificial variables which should be attached to the objective function as follows:

$$\frac{PENAL}{2} = \text{Max}[f_0(\bar{X}), f_0(\bar{a}), f_0(\bar{b})]$$

if $PENAL < 10^3$ then PENAL will be taken 10^3 .

CALC determines the subsets of dependent (basis) and independent variables at the starting point. If this is possible, it takes the basis made up of the supplementary variables. If not, a search for finding the basis will be done by subroutine INV(L).

The GREG algorithm at this stage is similar to the GRG as introduced in chapter one. A brief summary of the GREG algorithm is given in Appendix 1.

In the case of degenerate solution, GREG treats the problem as in the simplex method. This procedure makes the possibility of admitting into the basis some of the variables which are at their upper or lower bounds. This solution is not perfect, but in most of the cases converges the problem into the optimal solution.

2.4 MAIN PROGRAM AND NINE RELATED INTERNAL SUBROUTINES

MAIN PROGRAM

This program lists the parameters, defines the starting point $\bar{x}^0$, and their related bounds $\bar{a}$, $\bar{b}$, as supplied by the user, then calls subroutine CALC and GRG respectively. The computer flow diagram for the main program is given in Figure 2.1.

INTERNAL SUBROUTINE CALC

This subroutine initializes the counters, the Jacobian of the constraints, and the gradient of the objective function. At the starting point it generates the sufficient number of slack and artificial variables. It also computes the penalty for the artificial variables. Then it arranges the constraints by the following order:

First, inequality constraints not satisfied by the starting point.

Second, inequality constraints satisfied by the starting point.

Third, equality constraint not satisfied by the starting point.

Fourth, equality constraint satisfied by the starting point.

CALC also defines the subsets of dependent and independent variables, then calls subroutines CPH11 and PHIX1 for adding the additional variables into the constraints and adding the product of each artificial variable

**THIS BOOK
CONTAINS
NUMEROUS PAGES
WITH DIAGRAMS
THAT ARE CROOKED
COMPAIRED TO THE
REST OF THE
INFORMATION ON
THE PAGE.**

**THIS IS AS
RECEIVED FROM
CUSTOMER.**

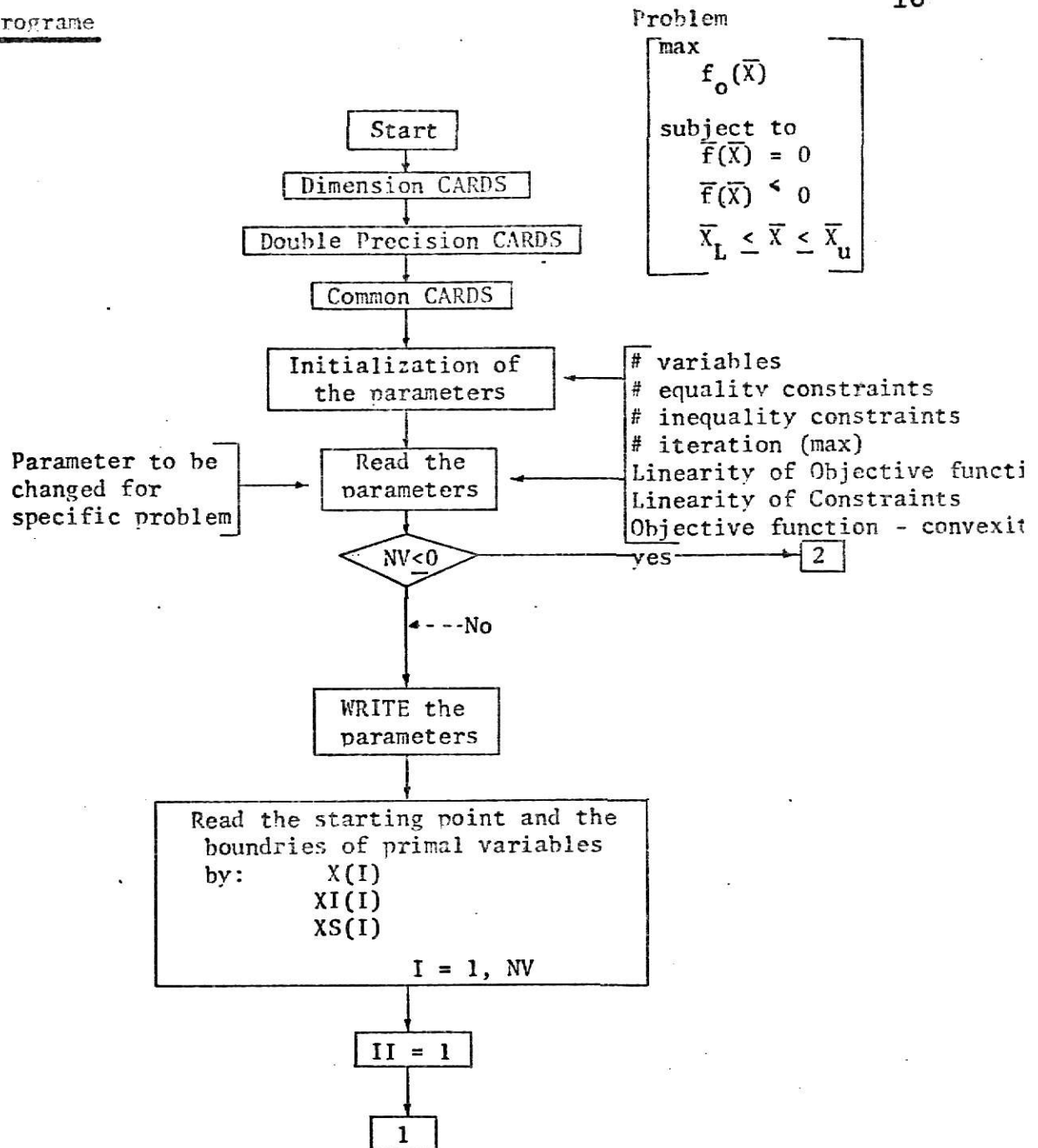
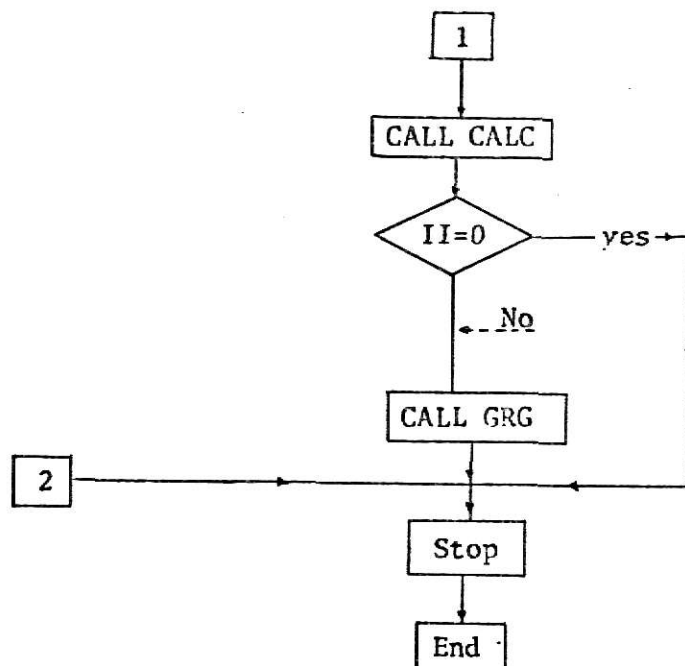


Fig 2.1 The computer flow diagram for the main program.



(II=0 the problem is not soluble)

X(I) = Starting Point
XI(I) = Lower bound
XS(I) = Upper bound

I = 1, NV
I = 1, NV
I = 1, NV

Fig 2.1 (continued).

and penalty into the objective function. At this stage CALC prints the solution of the starting point, related boundaries, the value of the objective functions and the value of each constraint with respect to the primal variables.

INTERNAL SUBROUTINE PHIX1

This subroutine adds the summation of the product of each artificial variable and penalty to the objective function.

INTERNAL SUBROUTINE CPHI1

This subroutine adds the slack and artificial variables into the related constraint.

INTERNAL SUBROUTINE GRG

The following computational work has been assigned to this subroutine:

- * The computation of the reduced gradient.
- * The computation of the projected reduced gradient.
- * Checking the stopping conditions.
- * The computations of the direction of movement for both the dependent and independent variables.
- * The computation of the direction of the movement by the diagonal direction method, if the user wishes to use that.
- * The computation of the direction of the movement by the conjugate direction method, if the user wishes to use that.
- * The computation of θ' .
- * Calling subroutine MAXFI for maximizing θ over the segment $(0, \theta')$, by the parabolic interpolation method.

- * Setting up $\bar{X}^1 = \bar{X}^0 + \theta \bar{h}^0$
- * Calling subroutine RENDOM for obtaining a feasible solution point which satisfying the constraints by the Newton's method.
- * Calculation the value of objective function with respect of the obtained feasible solution by calling subroutine PHIX and PHIX1, and checking whether it has been improved.
- * Doing the rest of the calculation in terms of changing the value of θ in the case that the objective function has not been improved or changing the basis when the treatment results in a degenerate solution.
- * Calling subroutine EXTRAC for printing the intermediate and/or the final solutions.

INTERNAL SUBROUTINE RENDOM

By applying the Newton's method, this subroutine obtains a feasible point which satisfies the constraints and improves the objective function.

INTERNAL SUBROUTINE CDB

Usually this subroutine is called by subroutine RENDOM, and it carries out the change of basis after finding a new point, and then defines the dependent and independent variables.

INTERNAL SUBROUTINE INV(L)

This subroutine with respect to the argument L does two different calculation: if $L = 1$, the subroutine calculates only the inverse of the given basis. If $L = 2$, the subroutine searches for a new basis and then finds the inverse.

INTERNAL SUBROUTINE MAXFI

This subroutine maximizes $\phi_0(\theta)$ with the given θ' over the segment of $(0, \theta')$. The parabolic interpolation method is used to calculate the θ .

INTERNAL SUBROUTINE EXTRC

This is the output subroutine. It prints the intermediate and/or the final solution. It also prints the value of the objective function at each iteration, a set of counters and the code of the stopping condition which caused during the computation.

2.5 GREG, FOUR EXTERNAL SUBROUTINES

Four external supplied subroutines are available for the user to define the objective function, the constraints, the Jacobian of the constraints and finally the gradient of the objective function.

The following consideration should be applied to each of the external subroutines.

1. Each of the externally supplied subroutines must have a set of "DOUBLE PRECISION" and "COMMON" statements that are commensurate to the internal GREG subroutines. Figure 2.2 shows the general list of the GREG Double precision and Common block definition statements.
2. In the case of using some extra real variables for defining the above functions, they must be declared by a "DOUBLE PRECISION" statement.
3. In all of the external user supplied subroutines the possibility of reading a data set and/or doing some additional arithmetical operations at the first call are provided. CALC is the first internal subroutine which calls to each of the external subroutines. The variable IT, which is the iteration counter, is equal to -1 when the first call occurs. By checking

ILLEGIBLE DOCUMENT

**THE FOLLOWING
DOCUMENT(S) IS OF
POOR LEGIBILITY IN
THE ORIGINAL**

**THIS IS THE BEST
COPY AVAILABLE**

```

DIMENSION B(100)
PRECISION A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
1,Y(150),C(150),VC(50),IPAS(50),IPAS(50),IVC(50),IVA(100)
DOUBLE PRECISION A,ALFA,ALC,ALA,ALR,TRA,C,DELTFI,DELTA,PTP,DELTA
1X,DL,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
2,EPSIL6,EPSIL9,FPS,EPSIL,FIL,FLO,F20,F1,F2,G,GAMA,GNDPM,GKS,
3GX,PHI,PSI,PSI2,PC,PHIO,PHIL,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
4,PE4,POB,RQ,RR,RA,SCAL,TR,TD,TC,TRFN,TRE,TR,TR1,TETA,TETAT,TOTG,TQ
5TGI,TETAL,TET2,TET3,TET4,TET5,TET,TETI2,TQ,TQ1,TQ2,TET1,TI
DOUBLE PRECISION TRB,T2,T22,TEX,TE,U,VC,VQ,V2,VI,V3,V,VCI,VCL,XI,
1XNV,XNO,XSR,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
2,XSE1,XSE2,XTP1,XTR,XTP2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORTQ,YRC,YR
3,Z,Z1,X,XC,XS,X1,X2,XP,B
DOUBLE PRECISION PIT,DISOLS
PRECISION DMDQ,DSORT,DABS,DMINI,DMAX1
COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,FIA
COMMON FIL,PHI,PSI,PSI3,TB,TD,TC,EPSI1,EPSILO,EPSIL2,YSORT
COMMON NV,NC,NK,NEG,NIN,NIV,NV1,NEV,NEVL,NTC,NIN1,NIN2,NIN3,NI
1N4,NVIN1,NVIN2,NVIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDR,JCDR,KC
2DB,KFIL,KLIN,KREN,KD,IRAS,IH3,IVC,IVA,IVR
COMMON KFUNC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,K
1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,ICIREC,JKO,LC

```

Fig 2.2. GREG double precision and common block definition statements.

if $IT < 0$, in the initialization a data set is read and/or some arithmetic operations are performed on the first call to each external subroutine.

4. All of the FORTRAN statements at each of the external subroutines must be written in the double precision form.

In order to describe the external subroutines, the following example problems illustrates each subroutine.

Problem 2.1

Maximize

$$f_0(\bar{X}) = (2X_1 - \frac{1}{2} X_1^2) + (3X_2 - \frac{1}{2} X_2^2)$$

Subject to:

$$f_1(\bar{X}) = X_1^2 + X_2^2 - 1 \leq 0$$

$$X_i \geq 0 \quad i = 1, 2$$

EXTERNAL SUBROUTINE PHIX

This subroutine defines the objective function by the FORTRAN variable PHI. PHI is described in terms of FORTRAN vector array XC(I), where I ranges from one to NV. Subroutine PHIX for problem 2.1 is given in Figure 2.4. The GREG double precision version has been written for the maximizing problem, but for the minimization problem, set $PHI = -PHI$. The flow diagram of this subroutine is given in Figure 2.3.

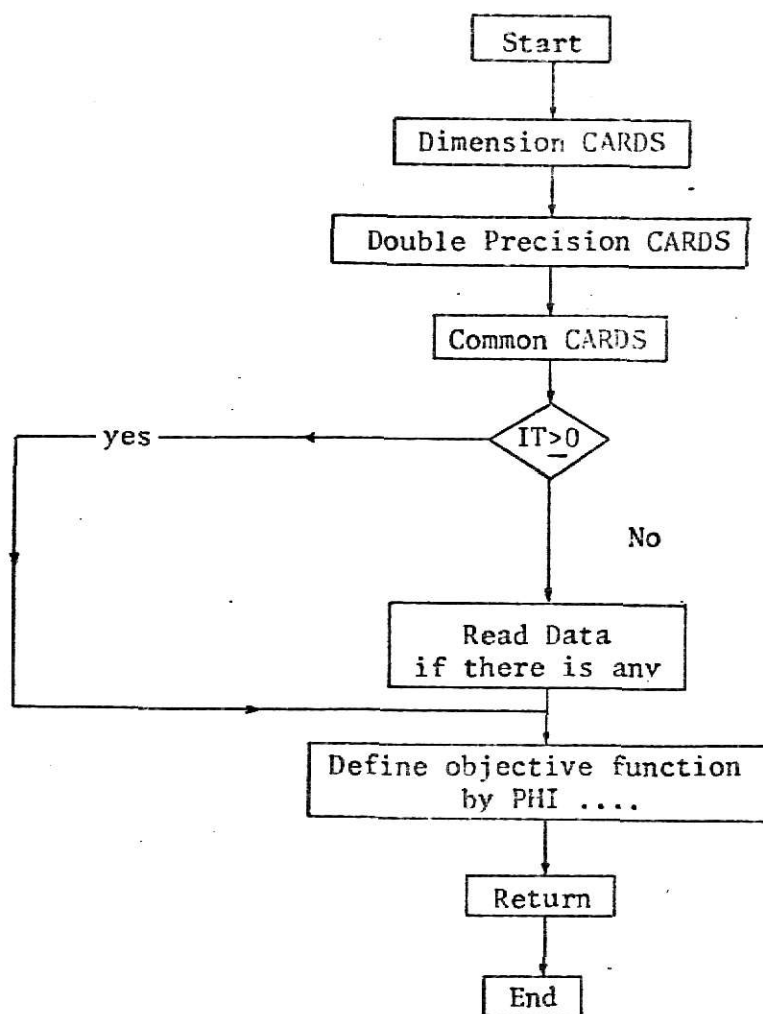


Fig 2.3 The flow diagram for subroutine PHIX.

```

FORTRAN IV G LEVEL 21          PHIX          DATE = 73111          05/53/50          PAGE 0001

0001  SUBROUTINE PHIX          GEGK 10
0002  DIMENSION R(100)          GEGK 20
0003  1,Y(150),C(150),VC(50),ALFA(50,50),X(150),XC(150),XI(150),XS(150)  GEGK 30
0004  DOUBLE PRECISION A,ALFA,ALC,ALA,ALP,IRA,C,DELTF1,DELTF2,DTP,DELTA
0005  1X,D1,D,EPSTL,EPSTL2,ETA,EPSTL3,EPSTL4,EPSTL5,EPSTL7,
0006  2EPSILU,EPSTL6,EPSTL9,EPSTL10,EPSTL11,EPSTL12,EPSTL13,EPSTL14,EPSTL15,EPSTL16,
0007  3GK,PHI,PSI,PSI2,PSI3,PC,PHI0,PHI1,PHI2,PSI1,TQP,PSI4,PENAL,PE1,PE2,PE3,
0008  4RPG,RJR,RC,RB,RA,SCAL,TH,TD,TC,THEN,THE,TR,TRI,TETA,TETAT,TQIG,TQ
0009  5IG1,TETAL,TET2,TET3,TET4,TET5,TET,TET2,TQ,TQ1,TQ0,TET1,TI
0010  DOUBLE PRECISION JN1RR,T2,T22,TEX,TE,U,VC,VU,V2,VI,V3,V,VCL,XI,
0011  1XNV,XNU,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,ASC,XSBL,XSH2,XMK,XIE1,XIE2
0012  2,XSE1,XSE2,XTR1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORTU,YRU,YR
0013  3,Z,ZI,X,XC,XS,X1,X2,XR,B
0014  DOUBLE PRECISION DIT,DISOLS
0015  DOUBLE PRECISION PMOD,DSORT,DAUS,DMINI,DMAX1
0016  COMMON R,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTF1,ETA
0017  COMMON FIL,PSI,PSI2,PSI3,TR,TD,TC,EPSTL,LPSILU,EPSTL2,YSORT
0018  COMMON NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEGA
0019  1N4,NVIN1,NV1,N2,NVIN3,INDEX,I1,IR,IR4,IS,IS1,I1,IBP,ICDB,JCDB,KCGEGA
0020  2EB,KFIL,KLIN,KREN,KD,IBAS,IB8,IVC,IVA,IVB
0021  COMMON KFCNC,KGRAD,KCONT,KINV1,KINV2,KCDRA,KJACO,KMAX1,KMAX2,KGEGA
0022  1KEN1,KKEN2,KINV,KCDBAL,KREN1,KREN21,IDIREC,JKO,LC
0023  PHI=2.00*XC(1)-0.500*XC(11)*2+3.00*XC(2)-0.500*XC(2)**2
0024  RETURN
0025  END

```

Fig 2.4. Subroutine PHIX for problem 2.1.

EXTERNAL SUBROUTINE CPHI

This subroutine defines the constraints by the FORTRAN vector array $VC(I)$, where $I = 1, \dots, NC$. $VC(I)$ is described in terms of FORTRAN vector array $XC(J)$, where $J = 1, \dots, NV$. Subroutine CPHI for the problem 2.2 is given in Figure 2.6. The constraints must be written in the following order.

First, inequality constraints

Second, equality constraints.

It is very important to note here that the GREG program is written only to solve the problem where the inequality constraints are in the $\bar{f}(\bar{X}) \leq 0$ form. The flow diagram of subroutine CPHI is given in Figure 2.5.

EXTERNAL SUBROUTINE JACOB

This subroutine defines the partial derivative of the constraints with respect to the primal variables $\frac{\partial f_i}{\partial X_i}$, where $i = 1, \dots, NC$ and $J = 1, \dots, NV$.

The Jacobian is stored in the FORTRAN matrix array $A(I,J)$. The rows of this matrix represent the constraint subscripts while the column of the matrix represents the order subscripts of the primal variables.

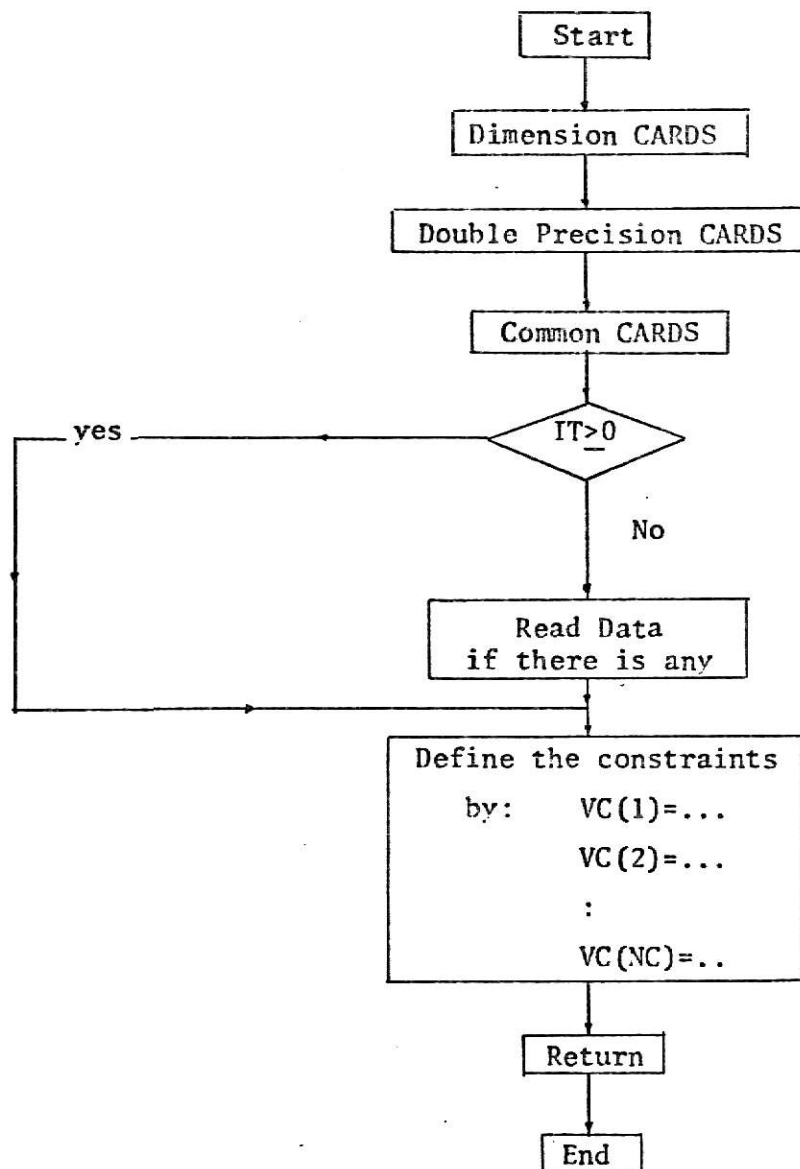
Each $A(I,J)$ is described by the FORTRAN array $SC(J)$, where $J = 1, \dots, NV$.

The subroutine JACOB for the problem 2.1 is given in Figure 2.8. The flow diagram for this subroutine is given in Figure 2.7.

EXTERNAL SUBROUTINE GRADFI

This subroutine defines the partial derivative of the objective function with respect to the primal variables, $\frac{\partial f_0}{\partial X_i}$, where $i = 1, \dots, NV$.

The gradient are stored in FORTRAN array $C(I)$, where I is the order



NC = # of constraints = NIN+NEG

NIN = # of inequality constraints

NEG = # of equality constraints

Note: The user should arrange the constraints of his problem in such a manner that all inequalities come first and the equalities follow.

Fig 2.5 The flow diagram for subroutine CPHI.

```

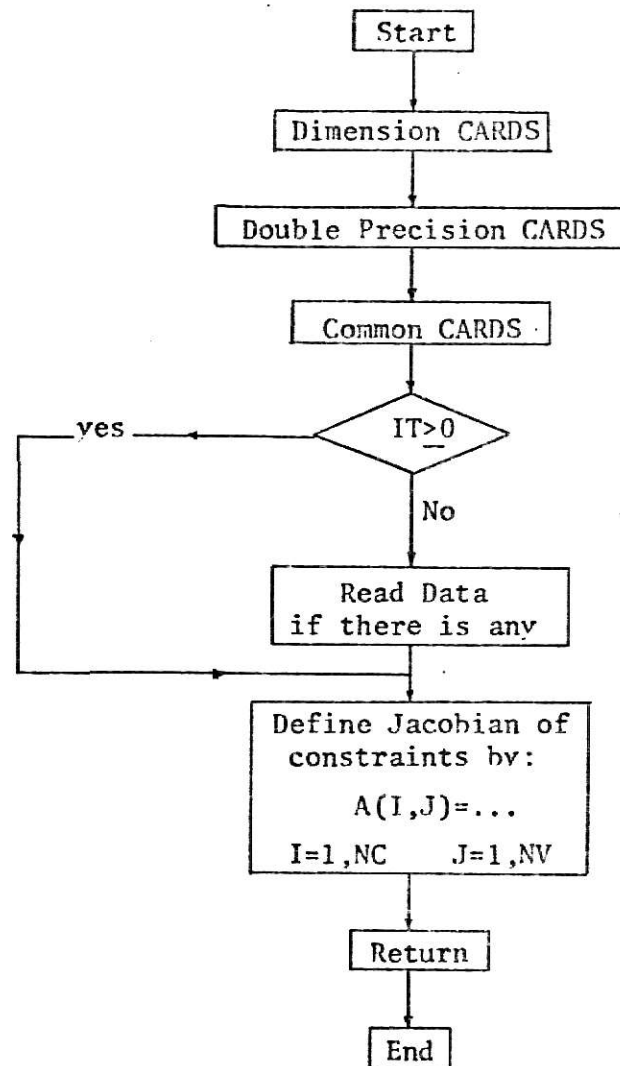
FORTRAN IV G LEVEL 21          JACOB          DATE = 73111          05/53/50          PAGE 0001

0001      SUBROUTINE JACOB
0002      DIMENSION B(100)
0003      1,Y(150),C(150),VC(50),IRAS(50),IHB(100),IVC(50),IVA(100)
0004      DOUBLE PRECISION A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFI,DELTA
1X,CL,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
2EPSILU,EPSIL6,EPSIL9,EPS,EPH1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
3CK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
4APG,ROR,RO,RE,RA,SCAL,TR,TD,TC,TREN,TRE,TR,TRI,TETA,TETAT,TQIG,TQ
5TCA,TETAI,TET2,TET3,TET4,TET5,TET,TET2,TQ,TQ1,TQ,TET1,T1
0005      DOUBLE PRECISION TRB,T2,T22,TEX,TE,U,VC,VU,V2,VI,V3,V,VC1,VCL,XI,
1XNV,XND,XSB,XSA,XNORM,XNDR,XINDRM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
2,XSEL,XSE2,XTF1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
3,Z,Z1,X,XC,XS,X1,X2,XR,B
0006      DOUBLE PRECISION DIT,DISOLS
0007      DOUBLE PRECISION DMUD,DSORT,DABS,DMIN1,DMAX1
0008      COMMON H,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
0009      COMMON F11,PHI,PSI,PSI3,TR,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
0010      COMMON NV,NC,NK,NFG,NIN,NTV,NV1,NEV,NEVL,NTG,NIN1,NIN2,NIN3,NIGEGA
1N4,NVHIN1,NVHIN2,NVHIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA
2DR,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
0011      COMMON KFGC,KGRAD,KCNT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
0012      A(1,1)=2.D0*XC(2)
0013      A(1,2)=2.D0*XC(2)
0014      RETURN
0015      END

          GEGM 10
          GEGM 20
          GEGM 30
          GEGM 180
          GEGM 190

```

Fig 2.6. Subroutine JACOB for problem 2.1.



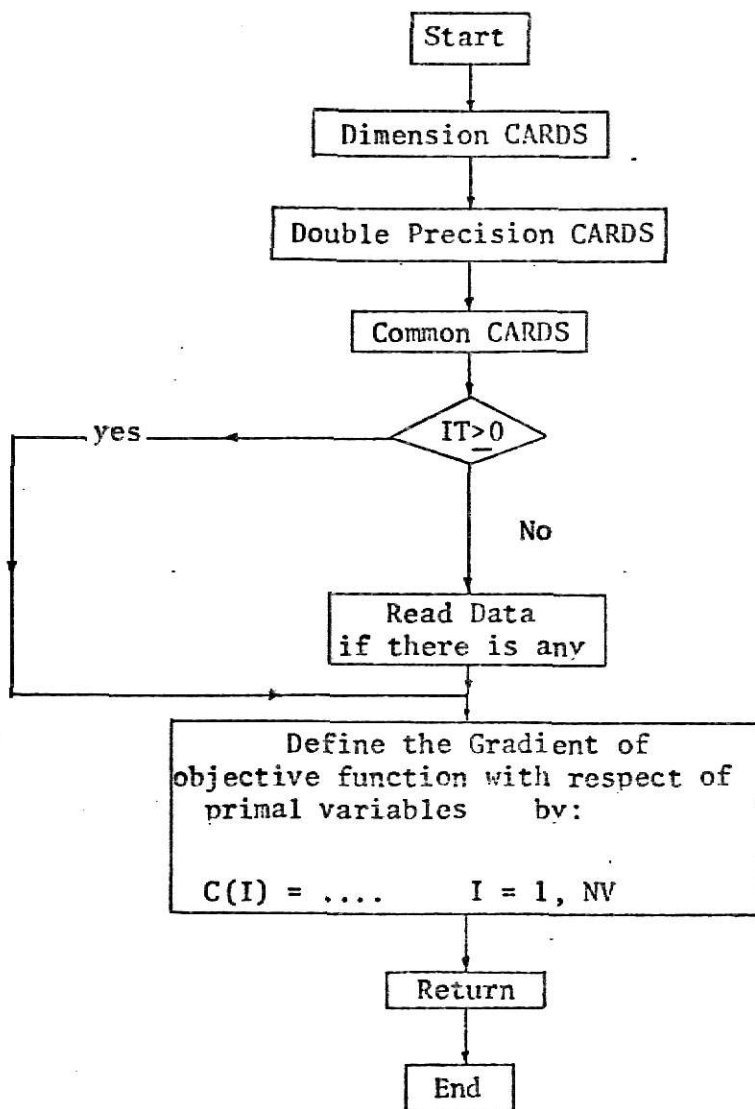
NC=# of constraints

NV=# of variables

Fig 2.7 The flow diagram for subroutine JACOB.

[illegible]

Fig 2.8. Subroutine CPHI for problem 2.1.

External Subroutine GRADFI (the gradient of the objective function)

NV = # of primal variables

Fig 2.9 The flow diagram for subroutine GRADFI.

```

FORTRAN IV G LEVEL 21          GRADFI          DATE = 73111          05/53/50          PAGE 0000

0001      SUBROUTINE GRADFI          GEGN 10
0002      DIMENSION B(100)
0003      DIMENSION A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
0004      1,Y(150),C(150),VC(150),IBAS(150),IBR(100),IVC(50),IVAI(100)
          DOUBLE PRECISION A,ALFA,ALC,ALA,ALR,TRA,C,DELTFI,DELTFB,DTP,DELTA
          1X,DI,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
          2EPSIL0,EPSIL6,EPSIL9,EPS,EPSI1,F11,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
          3GK,PHI,PSI,PSI3,PC,PHI0,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
          4RAPS,RUR,RU,RP,RA,SCAL,TB,TD,TC,TREN,TRE,TR,TR1,TETA,TETAT,TQTG,TQ
          5TG1,TETAL,TET2,TET3,TET4,TET5,TET,TETAT,TQ,TQ1,TQ0,TET1,T1
          DOUBLE PRECISION TRB,T2,T22,TEX,TE,U,VC,V0,V2,V1,V3,V,VCI,VCL,XI,
          1XNV,ANO,XSR,XSA,XNOPM,XNOP,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
          2,XSEL,XSE2,XTE1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORT0,YR0,YR
          3,Z,Z1,X,XC,XS,X1,X2,XR,B
          DOUBLE PRECISION DIT,DISOLS
          DOUBLE PRECISION DMOD,DSORT,DARS,DMIN1,DVAX1
          COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTF1,ETA
          COMMON F11,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
          COMMON NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTQ,NIN1,NIN2,NIN3,NIGEGA
          1N4,NVNIN1,NVAIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA
          2D9,KFIL,KLIN,KREN,KD,IBAS,IBR,IVC,IVA,IVB
          COMMON KFUNC,KGRAD,KCONT,KINVL,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
          1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JK0,LC
          C(1)=2.D0-XC(1)
          C(2)=3.D0-XC(2)
          RETURN
          END
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015

```

Fig 2.10. Subroutine GRADFI for problem 2.1.

subscripts of the primal variable $XC(I)$. Subroutine GRADFI for the problem 2.1 is given in Figure 2.10. In the case of a minimization problem set

$$C(I) = -C(I) \quad \text{where } I = 1, \dots, NV.$$

The flow diagram of this subroutine is given in Figure 2.9.

2.6 INPUT DATA FOR THE GREG DOUBLE PRECISION

Input data consists of parameters, the starting point with the related upper and lower bounds and finally the user given data for initialization purposes in subroutines PHIX, CPHI, JACOB and GRADIFI. This section has been allocated for describing these three types of input data.

1 - PARAMETERS

In the GREG double precision there are nineteen parameters which describe the specification of any kind of mathematical programming problem such as number of primal variables, number of inequality constraints, and so forth. A complete list of parameters and their definitions are given in Table 2.1. In the GREG main program, each of the parameters is given a default value. If the user wants to change the value of some or all of the parameters the IBM FORTRAN "NAMELIST" has been provided. PARM is the selected namelist title, and an example of this type of input is given in Figure 2.11. The parameters of this example have been listed for Problem 2.1.

An important fact should be mentioned here, namely that the noninteger parameters must be written by D FORMAT.

TABLE 2.1 - THE PARAMETERS

FORTRAN SYMBOLS	DESCRIPTION OF THE PARAMETERS	DEFAULT VALUE
NV	The number of primal variables	0
NIN	The number of inequality constraints	0
NEG	The number of equality constraints	0
NEVL	The maximum number of iteration for Newton's Method	20*
NT0(NT zero)	The maximum number of bisections in the parabolic interpolation method.	6*
ITET	The number of previous iterations used to help determine a maximum value for θ .	20*
ICONJ	Equal one if conjugate directions are desired, otherwise, zero	1
IDIAG	Equal one if diagonal directions are desired, otherwise, zero	0
ITMAX	The maximum number of iterations	50
KFIL	Equal one if the objective function is linear, otherwise, zero	0
KLIN	Equal one if all of the constraints are linear, otherwise, zero	0
NCO(NC zero)	Equal zero for the linear programming problem and is greater or equal to the number con- straints for nonlinear programming problem	10
ITSOR	The iteration which recording of the intermediate output starts ($1 \leq \text{ITSOR} \leq \text{ITMAX}$)	1*
ISOLSR	The solution is printed out every ISOLSR iter- ation. For the small problems, ISOLSR=ITMAX. For large problems, ISOLSR<ITMAX	50
EPSIL	Used as a criterion for the choice of a pivot in the changing an inversion of a basis. ($10^{-2} \leq \text{EPSIL} \leq 10^{-1}$)	0.1D0*

FORTTRAN SYMBOLS	DESCRIPTION OF THE PARAMETERS	DEFAULT VALUE
EPSILO (EPSIL zero)	Is used as a stopping criteria for Newton's Method	0.1D-02
EPSIL1	Is used as a stopping criteria if the problem is declared convex.	0.1D-02
EPSIL2	Is used as a stopping criterion by the norm of the reduced gradient	0.1D-02
PC	Equal zero if problem is non-convex, otherwise, one	0.D0

* These values are recommended not to be changed.

	<u>COLUMN</u>				
	1	2	3	4	5
<u>FIRST CARD</u>	1234567890	1234567890	1234567890	1234567890	1234567890
	8PARM NV=2,NIN=1,I CONJ=0, PC=.1D01,				
<u>SECOND CARD</u>	EPSILO=0.1D-06,EPSIL1=0.1D-04,EPSIL2=0.1D-04,8END				

FIGUR 2.11 — The fortran double precision namelist input for the problem 2.1 .

2 - STARTING POINT AND RELATED BOUNDS

A starting point, the upper bounds, and the lower bounds for each of the primal variables must be defined by the user. Main program reads starting point, lower bounds and upper bound according to the 8F10.2 format. This format allows eight values per card. If the number of variables are more than eight, use the next cards and repeat the process until all of the primal variables are listed.

3 - USER DATA

This portion of input is optional and it is dependent to the size of the problem. If the objective function, constraints, Jacobian or gradient of the problem need a large number of data to be written, this part of the input data will be used. In this case the formats for reading the noninteger data must be written in either nFw.d or nDw.d formats. CALC calls the external subroutines by the following order

PHIX

CPHI

JACOB

GRADFI

so the data has to be ordered in the same calling sequence of external subroutines.

THE GENERAL SEQUENCES OF THE DATA

The input data cards must be ordered into the following sequence.

Parameters data cards
Starting point cards
Lower Bounds cards
Upper Bounds cards
User data cards for subroutine PHIX
User data cards for subroutine CPHI
User data cards for subroutine JACOB
User data cards for subroutine GRADFI

2.7 JOB CONTROL LANGUAGE FOR UTILIZING THE GREG DOUBLE PRECISION PROGRAM

In this section the job control language for utilizing a GREG double precision program at Kansas State University is described.

The job control language for both load and access procedures are given in Figures 2.12, 2.13, 2.14, 2.15, 2.16 and 2.17.

Details for each procedure can be found at the Dr. C. L. Hwang's office, Industrial Engineering Department.

2.8 THE GREG DOUBLE PRECISION OUTPUT

The headings for the GREG single precision are in French, but for the GREG double precision version they have been translated into the English.

The general output in the GREG double precision may be classified into the following groups:

1. Definition of the variables
2. Definition of the stopping conditions codes
3. Parameter echo check
4. Starting conditions
5. Intermediate output

```
//          JOB (standard K.S.U. Job Card)
//          EXEC FORTGCLG
//FORT.SYSIN DD *

        complete GREG double precision source deck
        including the external subroutines.

/*
//LKED.SYSLMOD DD DSN=xxxxxx.yyyy(zzzzzz),
//  DISP=(,CATLG),UNT=2314,VOL=SER=nnnnnn,
//  SPACE=(TRK,(35,10,1),RLSE)
//GO.SYSIN DD*

        data

/*
```

Figure 2.12 JCL for loading the GREG double precision source deck into the partitioned data-set.

```

//          JOB(standard K.S.U. Job card)
//          EXEC FORTGLG
//LKED.SYSLMOD DD DSN=xxxxxx.yyyy(zzzzzz),
//  DISP=(,CATLG),UNIT=2314,VOL=SER=nnnnnn,
//  SPACE=(TRK,(35,10,1),RLSE)
//LKED.SYSINDD*
//          complete GREG double precision object deck including the
//          external subroutines
/*
//GO.SYSIN DD*
//          data
/*

```

Figure 2.13 JCL for loading the GREG double precision object deck into a partitioned data-set.

```
//          JOB(standard K.S.U. Job card)
//          EXEC FORTGCLG
//FORT.SYSIN DD *

          source deck

/*
//LKED.SYSLMOD DD DSN=xxxxxx.yyyy(zzzzzz),
//  DISP=(,CATLG),UNIT=2314,VOL-SER=nnnnnn,
//  SPACE=(TRK,(35,10,1),RLSE)
//LKED.SYSIN DD*

          object deck

/*
//GO.SYSIN DD *

          data

/*
```

Figure 2.14 JCL for loading the GREG double precision a combination of source and object decks into a partitioned data-set.

```

//          JOB(standard K.S.U. Job card)
//          EXEC FORTGCLG
//FORT.SYSIN DD *

          source deck for external subroutines
          (PHIX,CPHJ,JACOB and GRADFI)

/*
//LKED.LIB DD DSN=yyyy.xxxxxx,DISP=SHR,UNIT=2314,
//  VOL=SER=nnnnnn
//LKED.SYSIN DD *
          INCLUDE LIB(zzzzzz)
          ENTRY MAIN

/*
//GO.SYSIN DD *
          data
/*

```

Figure 2.15 Access to the stored GREG double precision program using a source deck only for subroutines PHIX, CPHI, JACOB and GRADFI.

```

//          JOB9standard K.S.U. Job card)
//          EXEC FORTGLG
//LKED LIB DD DSN=yyyy.xxxxxx,DISP=SHR,UNIT=2314,
//          VOL=SER=nnnnnn
//LKED.SYSIN DD *

          object deck for external subroutines
          (PHIX, CPHI, JACOB and GRADFI)

          INCLUDE LIB (zzzzzz)
          ENTRY MAIN
/*
//CO.SYSIN DD *

          data
/*

```

Figure 2.16 Access to the stored GREG double precision program using an object deck for subroutines (PHIX, CPHI, JACOB and GRADFI).

```

//          JOB(standard K.S.U. Job card)
//          EXEC FORTGCLG
// FORT.SYSIN DD *

          source deck

/*
//LKED.LIB DD DSN = yyyy.xxxxxx,DISP=SHR,UNIT=2314,
//  VOL=SER=nnnnnn
//LKED.SYSIN DD*

          object deck

          INCLUDE LIB (zzzzzz)

          ENTRY MAIN

/*
//GO.SYSIN DD *

          data

/*

```

Figure 2.17 Access to the stored GREG double precision program using a combination of object and source deck for subroutines (PHIX, CPHI, JACOB and GRADFI)

6- Final solution

7- Summary of the results.

Table 2.2 shows the output for the problem 2.1.

DEFINITION OF TERMS

NV = THE # OF ORIGINAL PROBLEM VARIABLES.
 NIN = THE # OF INEQUALITY CONSTRAINTS.
 NEG = THE # OF EQUALITY CONSTRAINTS.
 NEVL = THE MAXIMUM # OF ITERATION FOR NEWTON'S METHOD.
 NTO = THE MAXIMUM # OF BISECTIONS IN THE PARABOLIC INTERPOLATION PROCESS WHEN MAXIMIZING A FUNCTION OF A SINGLE VARIABLE.
 ITET = THE # OF PREVIOUS ITERATIONS USED TO HELP DETERMINE A MAXIMUM VALUE FOR TETA.
 ICONJ = 1 IF CONJUGATE DIRECTIONS ARE DESIRED.
 = 0 OTHERWISE.
 IDIAG = 1 IF DIAGONAL DIRECTION ARE DESIRED.
 = 0 OTHERWISE.
 ITMAX = THE MAXIMUM # OF ITERATIONS.
 KFIL = 1 IF THE OBJECTIVE FUNCTION IS LINEAR.
 = 0 OTHERWISE.
 KLIN = 1 IF ALL OF THE CONSTRAINTS ARE LINEAR.
 = 0 OTHERWISE.
 NCO = 0 FOR LINEAR PROGRAMMING.
 GREATER THAN THE # OF CONSTRAINTS FOR NONLINEAR PROGRAMMING.
 ITSOR = THE ITERATION WHICH RECORDING OF THE INTERMEDIATE OUT PUT STARTS.
 ISOLSR = THE SOLUTION IS PRINTED OUT EVERY ISOLSR ITERATIONS.
 EPSIL = A CRITERION FOR CHOICE OF A PIVOT IN THE CHANGING AND INVERSION OF A BASIS.
 EPSILO = A STOPPING CRITERIA FOR NEWTON'S METHOD.
 EPSIL1 = A STOPPING CRITERIA IF THE PROBLEM IS DECLARED CONVEX.
 EPSIL2 = A STOPPING CRITERION BY USING THE NORM OF THE REDUCED GRADIENT.
 PC = 0 IF PROBLEM IS NON-CONVEX.
 = 1 IF PROBLEM IS CONVEX.
 NJ = # OF VARIABLES AT THEIR LOWER OR UPPER BOUNDS.
 LC = # OF CONJUGATE CYCLES WHEN USING CONJUGATE DIRECTION.
 VN = NORM OF THE REDUCED GRADIENT.
 DELTAFL = DEFINED BY STOPPING CONDITION 3.
 ETA = DEFINED BY STOPPING CONDITION 2.
 NCDB = # OF CHANGE OF BASIS DURING THE ITERATION.
 NCN = # OF CALLS TO THE NEWTON METHOD.
 NITN = # OF ITERATIONS OF THE NEWTON METHOD DURING THE COURSE OF NCN CALLS.

Table 2-2a Definition of variables.

STOPPING CONDITIONS

```

# 0 IF THE MAXIMUM # OF ITERATIONS AUTHORIZED BY THE USER ARE REACHED.
# 1 IF THE NORM OF THE PROJECTED REDUCED GRADIENT, CALCULATED FROM THE
  LAST ITERATION IS SMALLER THAN EPSIL TIMES THE NORM OF THE FIRST
  ITERATION AND IF THERE ARE NO MORE ARTIFICIAL VARIABLES.
# 2 LET  ETAL(1)=ABS(GAMA(1))*XS(1)-X(1)) IF GAMA.GT.0.0 I=1,NK
  ETAL(1)=ABS(GAMA(1))*XI(1)-X(1)) IF GAMA.LT.0.0 I=1,NK
  ETA=MAX(ETAL(1),...,ETAL(NK)) NK=# OF INDEPENDENT VARIABLES.
  EPSIL3 =EPSIL1/ETA
  DELT13 =ETAL(1)+.....+ETAL(NK)
  IF (ETA.LT.EPSIL3).
# 3 IF THE FUNCTION DECLARED AS CONVEX (PC=1.0) THE PROGRAME STOPS IF
  (DELT13.LT.EPSIL1).
# 4 IF (C*Y.LT.0.0) THE PROCEDURE IS AS FOLLOWS.
  IF Y IS A CONJUGATE DIRECTION WE REPEAT THE ITERATION AND GIVE Y
  THE VALUE OF GRADIENT. IF AGAIN (C*Y.LT.0.0) WE EFFECT AN INVERSION
  FOLLOWED BY A NEW SEARCH OF BASIS IF ONCE MORE (C*Y.LT.0.0) WE
  STOP THE PROGRAME.
# 5 IF IN THE COURSE OF A REENTRY INTO THE DOMAIN" THE NEWTON METHOD
  GIVE A POINT SATISFYING THE CONSTRAINTS, BUT SUCH THAT THE FUNCTION
  VALUE AT THIS POINT IS LOWER THAN THAT OBTAINED AT THE PREVIOUS
  POINT, THEN WE MODIFY THE REENTRY POINT BY CHOOSING A TETA SUCH
  THAT. TETA=MAX(TETA/10.0,TGT6) WHERE TETA BEING THE STEP COND-
  UCTED DURING A FAILURE, TGT6 BEING A INTERPOLATED BETWEEN (0,TETA)
  SUCH A REDUCTION IS NORMALLY AUTHORIZED ONLY TWICE AFTER WHICH WE
  CARRY OUT AN INVERSION FOLLOWED BY A NEW SEARCH FOR BASIS. IF AFTER
  THIS WE STILL HAVE AN IDENTICAL RESULT THEN THE PROGRAME GIVE US
  THE OUTPUT STATEMENT # 7.
# 6 THIS MESSAGE CAN NOT BE OBTAINED IF THE BASIS IS PRIVILEGED. IF IT
  IS OBTAINED, IT IS BECAUSE THE DIRECTION Y LINEARISE THE CONSTRAINTS
  AT THE PREVIOUS POINT X(K) AND THE STEP TETA AT THE POINT X(K+1)=
  X(K)+TETA*Y DOES NOT SUFFICIENTLY IMPROVE THE COST FUNCTION. THE
  ACCURACY OF THE CALCULATION IS TESTED BY.
  ABS((PHI(X(K+1))-PHI(X(K)))).LT.(10.00-12*ABS(PHI(X(K))))
# 7 HERE THE BASIS IS NO MORE PRIVILEGED BUT WE CAN FIND THE SAME CON-
  DITIONS AS IN OUT PUT 6.
  ABS((PHI(X(K+1))-PHI(X(K)))).LT.(10.00-12*ABS(PHI(X(K))))
  WE SEEK AN INVERSE FOLLOWED BY A NEW SEARCH FOR BASIS. AFTER EVENT-
  UALLY PASSING TO A DIRECTION CONJUGATE TO THE GRADIENT. AS A FINAL
  RECURSE WE TAKE THE SAME DIRECTION OF STEPPING AS THE PROJECTED
  REDUCED GRADIENT, I.E., WE DO NOT TAKE INTO ACCOUNT SOME OF THE
  ELEMENTS OF GAMA(K). AN INSUFFICIENT STEP IN THE LINEARIZED DIRECTION
  DOES NOT ALLOW A SUCCESSFUL REENTRY INTO THE DOMAIN. WE KNOW IN THIS
  CASE THAT IT HAS BEEN REDUCED LIKE TETA (IN STEP 5). OR THAT THE
  NEWTON DOES NOT GIVE A SOLUTION AND THAT TETA IS REDUCED TO A CERTAIN
  NUMBER OF SOME POWER OF 10.
# 8 AS IN THE PRECEDING OUTPUTS WE HAVE
  ABS((PHI(X(K+1))-PHI(X(K)))).LT.(10.00-12*ABS(PHI(X(K))))
  BUT ANY TENTATIVE REENTRY IN TO THE DOMAIN HAS NOT BEEN EFFECTED IN
  THIS ITERATION. IN CONTRAST HOWEVER WE HAVE TRIED MANY TIMES TO
  DOUBLE TETA. WE HAVE EFFECTED AN INVERSION FOLLOWED BY A SEARCH FOR
  A BASIS, AND THEN HAVE TAKEN THE PROJECTED REDUCED GRADIENT AS THE
  STEPPING DIRECTION BEFORE COMING TO STOP.

```

Table 2-2b Definition of the stopping conditions codes.

PARAMETERS	
NV	2
NIN	1
NEG	0
NEVL	20
NTU	6
ITET	20
ICONJ	0
IDIAG	0
ITMAX	50
KFIL	0
KLIN	0
NCO	10
ITSOR	1
ISUISUR	50
EPSIL	0.10 00
EPSILO	0.10-06
EPSILI	0.10-04
EPSIL2	0.10-04
PC	0.10 01

Table 2-2c Parameter echo check.

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 2
 TOTAL NUMBER OF VARIABLES 3
 NUMBER OF CONSTRAINTS 1
 EPSILON FOR NEWTON'S METHOD 0.1000D-06
 EPSILON FOR GRADIENT TEST 0.1000D-04

OBJECTIVE FUNCTION 0.225000000000000D 01

	LOWER BOUND	STARTING POINT	UPPER BOUND
X1(1)	0.0	0.500000000000000D 00	0.100000000000000D 02
X1(2)	0.0	0.500000000000000D 00	0.100000000000000D 02

VALUE OF CONSTRAINTS

C(1) -0.500000000000000D 00

Table 2-2d Starting conditions.

IT 1	PHI	0.276464843750000	01	DIR. GRAD.	NO	0	YN	0.290	01	DELTA FI	0.380	02	ETA	0.240	02	NCDB	0	NCN	2	NITN	2
IT 2	PHI	0.307512739833300	01	DIR. GRAD.	NO	0	YN	0.270	01	DELTA FI	0.350	02	ETA	0.220	02	NCDB	0	NCN	2	NITN	2
IT 3	PHI	0.307740328134170	01	DIR. GRAD.	NO	0	YN	0.260	01	DELTA FI	0.330	02	ETA	0.210	02	NCDB	0	NCN	2	NITN	2
IT 4	PHI	0.307751861152010	01	DIR. GRAD.	NO	0	YN	0.260	01	DELTA FI	0.330	02	ETA	0.210	02	NCDB	1	NCN	3	NITN	4
IT 5	PHI	0.310553814386000	01	CIR. GRAD.	NO	1	YN	0.500	00	DELTA FI	0.590	00	ETA	0.590	00	NCDB	0	NCN	3	NITN	20
IT 6	PHI	0.310553822138260	01	DIR. GRAD.	NO	1	YN	0.720	00	DELTA FI	0.400	00	ETA	0.660	01	NCDB	0	NCN	21	NITN	167
	PHI	0.310553822138260	01				YN	0.720	00	DELTA FI	0.400	00	ETA	0.660	01						

EXECUTION TIME 613 CENTISECONDES

Table 2-2e Intermediate output.

PRIMAL VARIABLE		ASSOCIATED REDUCED GRADIENT VALUE	
X(1) =	0.55246026171653D 00	V(1) =	0.
	X(2) = 0.83353923106716D 00		V(2) = 0.71892103064937D 00
CONSTRAINT		LAGRANGE MULTIPLIER	
CONTRAINTE 1	-0.94960673074906D-08	U(1) =	0.86830930346867D 00

Table 2-2f Final out-put.

CHAPTER THREE

NUMERICAL TESTING OF THE GREG DOUBLE PRECISION PROGRAM

3.1 INTRODUCTION

The original GREG code was developed on the CDC6600 computer which has fifteen digit single precision arithmetic in its FORTRAN compiler. Later, the GREG code was adopted on the IBM system 360/50 computer [17] which has seven digit single precision arithmetic in its FORTRAN compiler.

Williams [17] solved a series of problem by GREG single precision code on the IBM 360/50 system. In performing the numerical study he compared GREG code with the other available techniques, in terms of reliability, limitation, accuracy and speed of solutions. He concluded that the GREG code was very successful at solving nonlinear programming problems.

In this chapter, the GREG double precision program is used for solving the same problems which were already solved by the GREG single precision program. The solution at each problem is compared with the solution obtained by the GREG single precision program in terms of the value of the objective function, the value of the variables, the number of iteration, and the speed of solution. The speed of solution is measured as the time of execution of the algorithm. It does not include compiling of the sub-routines and initialization of the problem. Initialization involves, basically, generating slack and artificial variables and adding them into

the objective function and constraints. All of the pertinent programming material is given in Appendix 2.

3.2 PROBLEM NUMBER ONE

Example 1 of chapter 2 is the first problem which has been solved by both GREG single and double precision codes. The problem definition is:

Maximize

$$f_0(\bar{X}) = (2x_1 - \frac{1}{2} x_1^2) + (3x_2 - \frac{1}{2} x_2^2)$$

subject to

$$f_1(\bar{X}) = x_1^2 + x_2^2 - 1 \leq 0$$

$$x_i \geq 0, \quad i = 1, 2$$

The solution of this problem by both GREG single and double precision codes are given in Table 3.1.

	GREG single precision	GREG double precision
Objective function $f_0 =$	3.1055507659	3.1055382212
Variables $x_1 =$ $x_2 =$	0.5545919537 0.8321228027	0.5524602617 0.8335392310
No. of iterations IT =	7	6
Stopping condition #	8	7
Execution time, in seconds.	2.50	6.13

Table 3.1 The solution of the problem number one for both GREG single and double precision.

3.3 PROBLEM NUMBER TWO

This problem concerns the reliability of a complex system. The model of this reliability problem has been proposed by Tillman, Hwang, Fan, and Lai and it has been solved by GREG single precision by Williams [11].

The statement of the problem is

Maximize

$$R_s = 1 - R_3((1-R_1)(1-R_4))^2 - (1-R_3)(1-R_2(1-(1-R_1)(1-R_4)))^2$$

subject to

$$f_1(\bar{R}) = K_1 R_1^{\alpha_1} + K_2 R_2^{\alpha_2} + K_3 R_3^{\alpha_3} + K_4 R_4^{\alpha_4} \leq C$$

$$R_i \geq R_{i,\min}, \quad i = 1, 2, 3, 4$$

The numerical values for the parameters are:

$$K_1 = K_2 = K_3 = 200, \quad K_4 = 300, \quad C = 800$$

$$(\alpha_i = 0.6, \quad R_{i,\min} = 0.5, \quad i = 1, \dots, 4)$$

The solution of this problem is shown in Table 3.2.

Objective function	GREG single precision	GREG double precision
$R_s =$	1.0	1.0
Variables		
$R_1 =$	1.0	1.0
$R_2 =$	1.0	1.0
$R_3 =$	0.5394473075	0.5394488375
$R_4 =$	0.7974168658	0.7974155742
No. of iterations		
IT =	5	5
Stopping condition		
#	3	1
Execution time, in seconds	0.78	1.23

Table 3.2. The solution of the problem number two by both GREG single and double precision codes.

3.4 PROBLEM NUMBER THREE

This problem involves a quadratic objective function with six quadratic constraints. The problem defined as:

Minimize:

$$f_0(\bar{X}) = a_1 x_3^2 + a_2 x_1 x_5 + a_3 x_1 - a_4$$

subject to:

$$f_1(\bar{X}) = b_1 + b_2 x_2 x_5 + b_3 x_1 x_4 - b_4 x_3 x_5 \geq 0$$

$$f_2(\bar{X}) = b_1 + b_2 x_2 x_5 + b_3 x_1 x_2 - b_4 x_3 x_5 \leq 92$$

$$f_3(\bar{X}) = c_1 + c_2 x_2 x_5 + c_3 x_1 x_2 + c_4 x_3^2 \geq 90$$

$$f_4(\bar{X}) = c_1 + c_2 x_2 x_5 + c_3 x_1 x_2 + c_4 x_3^2 \leq 110$$

$$f_5(\bar{X}) = d_1 + d_2 x_3 x_5 + d_3 x_1 x_3 + d_4 x_3 x_4 \geq 20$$

$$f_6(\bar{X}) = d_1 + d_2 x_3 x_5 + d_3 x_1 x_3 + d_4 x_3 x_4 \leq 25$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_3 \leq 45$$

$$27 \leq x_4 \leq 45$$

$$27 \leq x_5 \leq 45$$

The numerical values for the parameters are:

$$a_1 = 5.3578547, \quad a_2 = 0.8356891, \quad a_3 = 37.293239, \quad a_4 = 40792.141$$

$$b_1 = 85.334407, \quad b_2 = 0.0056858, \quad b_3 = 0.0006262, \quad b_4 = 0.0022053$$

$$c_1 = 80.51249, \quad c_2 = 0.0071317, \quad c_3 = 0.0029955, \quad c_4 = 0.0021813$$

$$d_1 = 9.300961, \quad d_2 = 0.0047026, \quad d_3 = 0.0012547, \quad d_4 = 0.0019085$$

In this problem the optimum solution was not reached by the normal procedure. To solve the problem a constant was subtracted from the objective function to scale it.

The constant value was 30373.94. The final solution of this problem is given in Table 3.3

	GREG single precision	GREG double precision
Objective function $f_0 =$	$-(292.0117+30373.94)$ $=-30665.9517$	$-(292.0150+30373.94)$ $=-30665.9550$
Variables		
$x_1 =$	78.0	78.0
$x_2 =$	33.0	33.0
$x_3 =$	29.9938964843	29.9939426347
$x_4 =$	45.0	45.0
$x_5 =$	36.7761230468	36.7758856848
No. of iterations IT =	8	5
Stopping condition #	7	1
Execution time, in seconds	1.65	2.51

Table 3.3. The solution of the problem number three by both single and double precision codes.

3.5. PROBLEM NUMBER FOUR

Maximize:

$$f_0(\bar{X}) = \sum_{i=1}^{10} b_i x_{i+5} - \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_i x_j - 2 \sum_{j=1}^5 d_j x_j^3$$

Subject to:

$$f_i(\bar{X}) = \sum_{j=1}^{10} a_{i,j+5} x_{j+5} - e_i - 2 \sum_{j=1}^5 c_{ji} x_j - 3d_i x_i^2 = 0$$

$$i = 1, 2, 3$$

$$f_i(\bar{X}) = \sum_{j=1}^{10} a_{i,j+5} x_{j+5} - e_i - 2 \sum_{j=1}^5 c_{ji} x_j - 3d_i x_i^2 + x_{i+12} = 0$$

$$i = 4, 5$$

$$0 \leq x_k \leq 100, \quad k = 1, \dots, 17$$

The numerical values for the parameters are given in Table 3.4.

The solution of this problem is given in Table 3.5.

3.6 PROBLEM NUMBER FIVE

This problem is a geometric programming example. The statement of the problem is:

Minimize:

$$f_0(\bar{X}) = 4x_1 + 10x_2 + 4x_3 + 2(x_1^2 + x_2^2)^{1/2}$$

subject to

$$f_1(\bar{X}) = x_1 x_2 x_3 \geq 100$$

$$x_i \geq 0, \quad i = 1, 2, 3$$

The solution of this problem is given in Table 3.6.

Table 3.4 gives the parameter values.

i	1	2	3	4	5
d_i	4	8	10	6	2
e_i	-15	-27	-36	-18	-12

i	1	2	3	4	5	6	7	8	9	10
b_i	-40	-2	$-\frac{1}{4}$	-4	-4	-1	-40	-60	5	1

value of c_{ij} :

$i \backslash j$	1	2	3	4	5
1	30	-20	-10	32	-10
2	-20	39	-6	-31	32
3	-10	-6	10	-6	-10
4	32	-31	-6	39	-20
5	-10	32	-10	-20	30

Table 3.4 Parameter values for problem four.

value of a_{ij} :

$i \backslash j$	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	-16	2	0	1	0
7	0	-2	0	0.4	2
8	-3.5	0	2	0	0
9	0	-2	0	-4	-1
10	0	-9	-2	1	-2.8
11	2	0	-4	0	0
12	-1	-1	-1	-1	-1
13	-1	-2	-3	-2	-1
14	1	2	3	4	5
15	1	1	1	1	1

Table 3.4 (Continued)

Objective function f_0	GREG single precision	GREG double precision
	-32.3486785806	-33.9621492223
Variables		
$x_1 =$	0.2994560599	0.5000984677
$x_2 =$	0.3332918286	0.3051029097
$x_3 =$	0.4006066918	0.3221967842
$x_4 =$	0.4287636280	0.2176330865
$x_5 =$	0.2243463993	0.1421636603
$x_6 =$	0.0	0.0
$x_7 =$	0.0	0.0
$x_8 =$	5.1755962371	3.6602819223
$x_9 =$	0.0	0.0000074672
$x_{10} =$	3.0615062713	3.2542665094
$x_{11} =$	11.8326454162	11.6137452756
$x_{12} =$	0.0	0.0
$x_{13} =$	0.0	0.0
$x_{14} =$	0.1027765870	0.0276781631
$x_{15} =$	0.0	0.0
$x_{16} =$	0.0	0.0
$x_{17} =$	0.0	0.0
No. of iteration IT =	36	42
Stopping condition #	1	8
Execution time, in seconds	12.88	58.54

Table 3.5 The solution of the problem number four by both single and double precision codes.

Objective function $ f_0 =$	GREG single precision	GREG double precision
	87.987747192	87.9877697106
Variables		
x_1	5.0872011184	5.0859008797
x_2	2.6814708709	2.6820680504
x_3	7.3307437896	7.3309854769
No of iteration IT =	9	9
Stopping condition #	1	1
Execution time, in seconds	0.83	2.27

Table 3.6 The solution of the problem number five by both single and double precision.

3.7 CONCLUDING REMARKS

As Williams concluded [17], the GREG code is very successful at solving nonlinear programming problems. It does have some faults. One is that it is very sensitive to numerical error and behaves very erratically without any indication of what the problems are. Usage of the GREG codes could be aided by the implementation of an error routine.

The solutions obtained by GREG single and double precision codes have been compared for each problem separately. In some of the problems, double precision version of GREG has been improved the objective function but in other problems the objective function has not been improved. The reason could be found by checking the stopping condition code. For example problem number one was stopped by stopping condition number 8 while it was running by GREG single precision, but the stopping condition

was number 7 by the double precision version. In some of the nonlinear programming problems when the primal variables are not limited at their upper bounds the more accurate result could be obtained by defining the upper bounds closer to the optimal vlaues of the variables. This could be accomplished by running the program two or three times, reducing each time the region between upper and lower bounds. The other approach is changing the starting point. In some of the cases, the problem would converge to a better solution if the starting point is defined equal to the lower bounds of all the variables. The GREG double precision codes takes more time to execute the problem.

REFERENCES

1. Abadie, J., Solution des questions de degenerescence dans la methode GREG. EDF* note HI 143/00 du 25 September 1969.
2. Abadie, J., Integer and Nonlinear Programming, ed. J. Abadie, North Holland Publishing Co., Amsterdam, 1970.
3. Abadie, J. and J. Carpentier, Generalisation de la methode du gradient reduit de Wolfe au cas de Contraintes non-lineaires, EDF* note HR 6678, du 27 Octobre 1965.
4. Abadie, J. and J. Carpentier, Generalisation de la method du gradient reduit de Wolfe au cas de constraints non-lineaires, in: Proc. IFORS Conf., eds. D. B. Hertz and J. Melese, Wiley, New York, pp. 1041-1053, 1966.
5. Abadie, J. and J. Carpentier, Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints, In: Optimization, ed. R. Fletcher, Academic Press, New York, 1969.
6. Abadie, J. and J. Guigou, Gradient reduit generalise, EDF* note HI 069/2 du Avril 1969.
7. Abadie, J. and J. Guigou, Numerical experiments with the GRG method, in: Integer and Nonlinear Programming, ed. J. Abadie, North Holland Publishing Co., Amsterdam, 1970.

8. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963.
9. Fletcher, R., and C. H. Reeves, Function minimization by conjugate gradients, the computer journal, Vol. 7, July 1964, pp. 149-154.
10. Fletcher, R., and M. J. D. Powell, A rapidly convergent descent method for minimization, the computer Journal, Vol. 6, July 1963, pp. 163-168.
11. Guigou, J., Presentation et utilisation due code GREG, EDF* note HI 102/02 du 9 Juin 1969.
12. Guigou, J., Presentation et utilisation due code GREG, EDF* note HI 502/2 due 25 Mai 1971.
13. Himmelblau, D. M., Applied Nonlinear Programming, McGraw-Hill Book Company, New York, 1972.
14. Hwang, J. L, J. L. Williams, and L. T. Fan, Introduction to the Generalized Reduced Gradient Method, Report No. 39, Institute For Systems Design and Optimization, Kansas State University, Manhattan, Kansas, 1972.
15. Kuhn, H. W., and A. W. Tucker, Nonlinear Programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Calif., pp. 481-492, 1951.

16. Kuhn, H. W., and A. W. Tucker, Linear Inequalities and Related Systems, in: Annals of Mathematics Studies Number 38, Princeton University Press, Princeton, N. J., 1956.
17. Williams, J., Optimization of Industrial Systems with the Separable Programming and Generalized Reduced Gradient Methods, ed. J. Williams, A Master's Thesis, Kansas State University., Manhattan, Kansas, 1972.
18. Wolfe, P., Methods of Nonlinear Programming, in: Recent Advances in Mathematical Programming eds. R. Graves and P. Wolfe, McGraw-Hill New York, 1963.
19. Wolfe, P., Methods for linear constraints, in: Nonlinear Programming, ed. J. Abadie, North Holland Publishing Co., Amsterdam, 1967.

APPENDIX 1

A BRIEF SUMMARY OF GREG ALGORITHM

The following GREG algorithm is for the nonlinear programming problem defined specifically as

Maximize:

$$f_0 = f(\bar{X})$$

subject to:

$$v_i = f_i(\bar{X}) \leq 0, \quad i = 1, \dots, m$$

where

m = number of constraints.

GREG ALGORITHM

Step 1

Define the problem according to the following substeps:

Step 1.1

Identify the boundaries of the variables $(\bar{a}, \bar{b})$, and define the starting point $(\bar{X}^0)$ such that

$$a_i \leq x_i \leq b_i, \quad i = 1, \dots, n$$

Step 1.2

Define the problem as follows:

Identify the objective function by f_0 and constraints by $(v_i, i = 1, \dots, m)$. Let $(C_i = \frac{\partial f_0}{\partial x_i}, i = 1, \dots, n)$ and $(A_{ij} = \frac{\partial v_i}{\partial x_j}, i = 1, \dots, m \text{ and } j = 1, \dots, n)$.

Step 2

Generate a sufficient additional variables as follows:

Step 2.1

Check the selected starting point satisfies each constraint or not as follows:

$$v_i \begin{cases} \text{inequality type} \begin{cases} v_i > \varepsilon_0 & \longrightarrow \text{constraint not satisfied.} \\ v_i \leq \varepsilon_0 & \longrightarrow \text{constraint satisfied.} \end{cases} \\ \text{equality type} \begin{cases} |v_i| > \varepsilon_0 & \longrightarrow \text{constraint not satisfied.} \\ |v_i| \leq \varepsilon_0 & \longrightarrow \text{constraint satisfied.} \end{cases} \end{cases}$$

$$i = 1, \dots, m$$

Let

m_1 = number of inequality constraints not satisfied.

m_2 = number of inequality constraints satisfied.

m_3 = number of equality constraints not satisfied.

If $m_1 = 0$ and $m_3 = 0$, go directly to step 2.3.

Step 2.2

Calculate the penalty for the artificial variables as follows:

$$\frac{P}{2} = \max [f_0(\bar{a}), f_0(\bar{b}), f_0(\bar{X}^0)] \quad \text{and if } P < 10^3 \text{ take } P = 10^3$$

Step 2.3

Generate the slack and artificial variables as follows:

A slack variables for each inequality constraint not satisfied [$x_i = 0.$,

$a_i = 0.$, $b_i = \infty$, and $C_i = 0.$, $i = (n+1), \dots, (n+m_1)]$.

An artificial variables for each inequality constraint not satisfied

$[x_i = -v_j$, $a_i = -\infty$, $b_i = 0.$ and $C_i = P$, $i = (n+m_1+1), \dots, (n+2m_1)]$.

A slack variable for each inequality constraint satisfied $[x_i = -v_j$,

$a_i = 0.$, $b_i = \infty$ and $C_i = 0.$, $i = (n+2m_1+1), \dots, (n+2m_1 + m_2)]$.

An artificial variable for each equality constraint not satisfied [if

$v_j > \epsilon_0$ then $(x_i = -v_j$, $a_i = -\infty$, $b_i = 0.$ and $C_i = -P)$, if $v_j < \epsilon_0$ then

$(x_i = -v_j$, $a_i = 0.$, $b_i = \infty$, $C_i = -P)$, $i = (n+2m_1+m_2+1), \dots,$

$(n+2m_1+m_2+m_3)]$.

Step 2.4

Add the additional variables into the constraints. For inequality constraints not satisfied:

$$v_j = v_j + x_{n+j} + x_{n+m_1+j}, \quad j = 1, \dots, m_1$$

For inequality constraints satisfied

$$v_j = v_j + x_{n+2m_1+j} + \epsilon_0, \quad j = (m_1+1), \dots, (m_1+m_2)$$

For equality constraints not satisfied

$$v_j = v_j + x_{n+2m_1+m_2+j}, \quad j = (m_1+m_2+1), \dots, (m_1+m_2+m_3)$$

Step 2.5

Add the artificial variables to the objective function

$$\phi_0 = f_0 + \sum_{i=1}^{m_1} C_{n+m_1+i} x_{n+m_1+i} + \sum_{j=1}^{m_3} C_{n+2m_1+m_2+j} x_{n+2m_1+m_2+j}$$

Step 2.6

Divide the variables in terms of dependent and independent variables. Let N be the total number of variables including the primal and additional variables. Therefore:

$$N = n + 2m_1 + m_2 + m_3$$

Select the basis (dependent variables) among the additional variables.

Apply the following changes:

Identify $[x_i, a1_i, b1_i, C1_i \text{ and } A1_{ji}, i = 1, \dots, (N-m) \text{ and } j = 1, \dots, m]$ for independent variables.

Identify $[y_i, a2_i, b2_i, C2_i \text{ and } A2_{ji}, i = 1, \dots, m \text{ and } j = 1, \dots, m]$ for dependent variables.

Set the iteration counter equal zero. $IT = 0$

Step 3

Calculate the reduced gradient according to the following substeps.

Set $IT = IT + 1$.

Step 3.1

Compute the following:

ϕ_0^0 = The value of objective function at this stage.

$$C1_i = \frac{\partial \phi_0^0}{\partial x_i}, \quad i = 1, \dots, (N-m).$$

$$A1_{ji} = \frac{\partial f_j}{\partial x_i}, \quad i = 1, \dots, (N-m) \text{ and } j = 1, \dots, m.$$

$$C2_i = \frac{\partial \phi_0}{\partial y_i}, \quad i = 1, \dots, m.$$

$$A2_{ji} = \frac{\partial f_i}{\partial y_i}, \quad i = 1, \dots, m \text{ and } j = 1, \dots, m.$$

Step 3.2

Compute the inverse of the matrix $\bar{A}2$ and identify it by $\bar{\alpha}$.

Step 3.3

Compute the reduced gradient as follows:

Let

$$u_i = - \sum_{j=1}^m C2_i \alpha_{ji}, \quad i = 1, \dots, m$$

And then

$$g_i = C1_i + \sum_{j=1}^m u_j A1_{ji}, \quad i = 1, \dots, (N-m)$$

Step 3.4

Compute the projected reduced gradient.

$$\gamma_i = \begin{cases} 0 & \text{if } g_i = 0 \\ 0 & \text{if } g_i < 0 \text{ and } x_i = al_i \\ 0 & \text{if } g_i > 0 \text{ and } x_i = bl_i \\ g_i & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (N-m)$$

Step 4

Check the stopping conditions.

Step 4.1

Calculate the norm of the projected reduced gradient.

$$\Gamma = \left(\sum_{i=1}^{N-m} (\gamma_i)^2 \right)^{1/2}$$

Step 4.2

Calculate $\Delta\phi$ and τ

$$\tau_i = \begin{cases} 0 & \text{if } \gamma_i = 0 \\ \gamma_i (b1_i - x_i) & \text{if } \gamma_i > 0 \\ \gamma_i (a1_i - x_i) & \text{if } \gamma_i < 0 \end{cases}$$

$$i = 1, \dots, (N-m)$$

$$\Delta\phi = \sum_{i=1}^{N-m} \tau_i$$

And

$$\tau = \max [\tau_1, \tau_2, \dots, \tau_{N-m}]$$

Let

$$\epsilon_3 = \frac{\epsilon_1}{n}$$

Step 4.3

Check for stopping condition

If $IT \geq K$ go to STOP

If $\Delta\phi \leq \epsilon_1$ go to STOP

If $\tau \leq \epsilon_3$ go to STOP

Step 5

Calculate the direction of movement.

let

$$\epsilon_4 = 10^{10} \quad \text{and} \quad \epsilon_5 = \frac{0.01}{n}$$

Then

$$\epsilon_u = \frac{1}{10} \epsilon_4 \quad \text{such that } \tau \leq \epsilon_u$$

and

$$\epsilon_6 = \epsilon_u \cdot \epsilon_5$$

Step 5.1

Direction of movement for independent variables.

$$h_i = \begin{cases} 0 & \text{if } \gamma_i > 0 \quad \text{and } \tau_i \leq \epsilon_6 \text{ and } x_i \text{ is an artificial variable} \\ 0 & \text{if } \gamma_i > 0 \quad \text{and } \tau_i \geq \epsilon_6 \text{ and } x_i \text{ is an artificial variable} \\ 0 & \text{if } \gamma_i = 0 \\ \gamma_i & \text{otherwise} \end{cases}$$

Step 5.2

Calculate the norm of the direction of movement for independent variables

$$H = \left(\sum_{i=1}^{N-m} (h_i)^2 \right)^{1/2} \quad \text{and if } H = 0 \text{ go to step 5}$$

Step 5.3

Direction of movement for dependent variables.

Let

$$z_i = \sum_{j=1}^{N-m} A1_{ij} \cdot h_j, \quad i = 1, \dots, m$$

And then

$$k_i = - \sum_{\ell=1}^m \alpha_{j\ell} z_\ell, \quad i = 1, \dots, m \text{ and } j = 1, \dots, m$$

Step 5.4

Calculate the product of the gradient and the direction of movements.

$$S = \sum_{i=1}^{N-m} C1_i h_i + \sum_{j=1}^m C2_j k_j$$

Step 6

Calculate the θ .

Step 6.1

Calculate θ' for dependent variables.

$$\theta'1_i = \begin{cases} \frac{b2_i - y_i}{k_i} & \text{if } k_i > 0 \\ \frac{a2_i - y_i}{k_i} & \text{if } k_i < 0 \end{cases}$$

$$i = 1, \dots, m$$

Then

$$\theta'1 = \min [\theta'1_1, \theta'1_2, \dots, \theta'1_m]$$

Step 6.2

Calculate θ' for independent variables.

$$\theta'2_i = \begin{cases} \frac{b1_i - x_i}{h_i} & \text{if } h_i > 0 \\ \frac{a1_i - x_i}{h_i} & \text{if } h_i < 0 \end{cases}$$

$$i = 1, \dots, (N-m)$$

Then

$$\theta'2 = \min [\theta'2_1, \theta'2_2, \dots, \theta'2_{N-m}]$$

Step 6.3

$$\theta' = \min [\theta'1, \theta'2]$$

Step 6.4

Maximize θ over the segment $(0, \theta')$ by the parabolic interpolation

Step 6.4.1

Calculate $x1_i = x_i + h_i \theta'$, $i = 1, \dots, (N-m)$

and $y1_i = y_i + k_i \theta'$, $i = 1, \dots, m$

$$x1_i = \begin{cases} a1_i & \text{if } x1_i < a1_i \\ b1_i & \text{if } x1_i > b1_i \\ x1_i & \text{otherwise} \end{cases} \quad y1_i = \begin{cases} a2_i & \text{if } y1_i < a2_i \\ b2_i & \text{if } y1_i > b2_i \\ y1_i & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (N-m) \quad i = 1, \dots, m$$

And

ϕ_0^1 = The value of the objective function with respect to these variables.

Step 6.4.2

Let $\theta'' = \frac{\theta'}{2}$, then calculate

$x2_i = x_i + h_i \theta''$, $i = 1, \dots, (N-m)$

$y2_i = y_i + k_i \theta''$, $i = 1, \dots, m$

$$x2_i = \begin{cases} a1_i & \text{if } x2_i < a1_i \\ b1_i & \text{if } x2_i > b1_i \\ x2_i & \text{otherwise} \end{cases} \quad y2_i = \begin{cases} a2_i & \text{if } y2_i < a2_i \\ b2_i & \text{if } y2_i > b2_i \\ y2_i & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (N-m) \quad i = 1, \dots, m$$

And

ϕ_0^2 = The value of the objective function with respect to these variables.

Step 6.4.3

$$\text{Let } F1 = (\phi_0^1 + 3\phi_0^0 - 4\phi_0^2) \cdot \frac{\theta'}{2}$$

$$\text{And } F2 = 2 \cdot (\phi_0^1 + \phi_0^0 - 2\phi_0^2)$$

$$\text{Then } F = F1/F2$$

If $|F2| \leq 10^{-6}$ go to step 6.4.4

If $F \leq 0$. go to step 6.4.4

If $F \geq \theta'$ go to step 6.4.4

otherwise calculate

$$x3_i = x_i + h_i F, \quad i = 1, \dots, (N-m)$$

$$y3_i = y_i + k_i F, \quad i = 1, \dots, m$$

$$x3_i = \begin{cases} a1_i & \text{if } x3_i < a1_i \\ b1_i & \text{if } x3_i > b1_i \\ x3_i & \text{otherwise} \end{cases} \quad i = 1, \dots, (N-m)$$

$$y3_i = \begin{cases} a2_i & \text{if } y3_i < a2_i \\ b2_i & \text{if } y3_i > b2_i \\ y3_i & \text{otherwise} \end{cases} \quad i = 1, \dots, m$$

And

ϕ_0^3 = The value of the objective function with respect to these values.

If $\phi_0^3 \leq \phi_0^2$ go to step 6.4.4

Otherwise take $\phi_0^2 = \phi_0^3$ and $\theta'' = F$

Step 6.4.4

$$\text{PHI} = \max(\phi_0^1, \phi_0^2) \text{ and } \theta = \begin{cases} \theta' & \text{if } \text{PHI} = \phi_0^1 \\ \theta'' & \text{if } \text{PHI} = \phi_0^2 \end{cases}$$

If $\text{PHI} \geq \phi_0^0$ go to step 6.4.5

Otherwise do some modification in order to obtain some value for

θ (i.g. $\theta = \frac{\theta'}{4}$) such that $\text{PHI} \geq \phi_0^0$, if this is not possible change the basis. The best candidates for bringing into the basis are those variables which are at their upper or lower bounds.

Step 6.4.5

If $\theta = \theta'$, change the value of the related variable to the upper or lower bound, if it's direction of movement is greater or less than zero respectively.

Step 7

Set up the new value for the variables.

$$x_{l_i} = x_i + \theta h_i, \quad i = 1, \dots, (N-m)$$

$$y_{l_i} = y_i + \theta k_i, \quad i = 1, \dots, m$$

$$x_{l_i} = \begin{cases} a_{l_i} & \text{if } x_{l_i} < a_{l_i} \\ b_{l_i} & \text{if } x_{l_i} > b_{l_i} \\ x_{l_i} & \text{otherwise} \end{cases} \quad y_{l_i} = \begin{cases} a_{2_i} & \text{if } y_{l_i} < a_{2_i} \\ b_{2_i} & \text{if } y_{l_i} > b_{2_i} \\ y_{2_i} & \text{otherwise} \end{cases}$$

$$i = 1, \dots, (N-m) \quad i = 1, \dots, m$$

Step 7.1

Check the constraints by the Newton's method

Let

$$R_0 = \max[|v_1|, |v_2|, \dots, |v_m|]$$

If $R_0 \leq \epsilon_0$, all the constraints are satisfied and go to step 8.

Otherwise go to step 7.2.

Step 7.2

Calculate

$$DEL_j = - \sum_{k=1}^m \alpha_{jk} v_i, \quad j = 1, \dots, m$$

And

$$y_{l_i} = y_{l_i} + DEL_i, \quad i = 1, \dots, m$$

If $a_{2_i} < y_{l_i} < b_{2_i}$ go to step 7.1

Otherwise set $\theta = \frac{\theta}{2}$ and go to step 7.

If after several reduction on θ , still the constraints are not satisfied, change the basis.

Step 8

Calculate the value of the objective function.

If $\text{PHI} > \phi_0$ then set

$$x_i = x_{1_i} \quad i = 1, \dots, (N-m)$$

$$y_i = y_{1_i} \quad i = 1, \dots, m \quad \text{and go to step 3}$$

Otherwise change the basis, set $\text{IT} = \text{IT}-1$ and go to step 3

A NUMERICAL EXAMPLE

Maximize:

$$f_0(\bar{X}) = (2x_1 - \frac{1}{2} x_1^2) + (3x_2 - \frac{1}{2} x_2^2)$$

Subject to:

$$f_1(\bar{X}) = x_1^2 + x_2^2 - 1 \leq 0$$

$$x_1, x_2 \geq 0$$

Step 1.1

$$a_1 = 0., \quad b_1 = 10., \quad x_1 = 0.5$$

$$a_2 = 0., \quad b_2 = 10., \quad x_2 = 0.5$$

Step 1.2

$$f_0 = 2x_1 - \frac{1}{2} x_1^2 + 3x_2 - \frac{1}{2} x_2^2$$

$$v_1 = x_1^2 + x_2^2 - 1.$$

$$c_1 = 2. - x_1$$

$$C_2 = 3. - x_2$$

$$A_{11} = 2x_1$$

$$A_{12} = 2x_2$$

$$M = 2$$

Step 2.1

$$v_1 = -0.5 < \epsilon_0 = .01, \text{ so the constraint satisfied}$$

Then

$$m_1 = 0, \quad m_2 = 1, \quad m_3 = 0$$

Step 2.3

$$x_3 = 0.5, \quad a_3 = 0., \quad b_3 = \infty, \quad C_3 = 0., \quad A_{13} = 1.0$$

Step 2.4

$$v_1 = x_1^2 + x_2^2 - 1 + x_3 + \epsilon_0$$

Step 2.5

$$\phi_0 = 2x_1 - \frac{1}{2} x_1^2 + 3x_2 - \frac{1}{2} x_2^2$$

Step 2.6

$$N = n + m_2 = 2 + 1 = 3$$

$$\bar{X} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{and} \quad \bar{Y} = [x_3]$$

Then

$$x_1 = x_1, \quad a1_1 = a_1, \quad b1_1 = b_1, \quad C1_1 = C_1 \quad \text{and} \quad A1_{11} = A_{11}$$

$$x_2 = x_2, \quad a1_2 = a_2, \quad b1_2 = b_2, \quad C1_2 = C_2 \quad \text{and} \quad A1_{12} = A_{12}$$

$$y_1 = x_3, \quad a2_1 = a_3, \quad b2_1 = b_3, \quad C2_1 = C_3 \quad \text{and} \quad A2_{11} = A_{13}$$

$$IT = 0$$

Step 3

$$IT = IT + 1 = 1$$

Step 3.1

$$\phi_0^0 = 2.25$$

$$C1_1 = 1.5, \quad C1_2 = 2.5$$

$$A1_{11} = 1.0, \quad A1_{12} = 1.0$$

$$C2_1 = 0.$$

$$A2_{11} = 1.0$$

Step 3.2

$$\alpha_{11} = 1.0$$

Step 3.3

$$u_1 = -C2_1 \alpha_{11} = 0.$$

$$g_1 = 1.5, \quad g_2 = 2.5$$

Step 3.4

$$\gamma_1 = 1.5, \quad \gamma_2 = 2.5$$

Step 4.1

$$r = [(1.5)^2 + (2.5)^2]^{1/2} = 2.9$$

Step 4.2

$$\tau_1 = 1.5(10. - 0.5) = 14.25$$

$$\tau_2 = 2.5(10. - 0.5) = 23.75$$

$$\Delta\phi = 14.25 + 23.75 = 38.0$$

$$\tau = \max [14.25, 23.75] = 23.75$$

$$\epsilon_3 = \frac{.01}{2} = .005$$

Step 4.3

$$IT = 1 < K = 10, \text{ then continue}$$

$$\Delta\phi = 38.0 > \epsilon_1 = .01, \text{ then continue}$$

$$\tau = 23.75 > \epsilon_3 = .005, \text{ then continue}$$

Step 5

$$\epsilon_4 = 10^{10}, \quad \epsilon_5 = \frac{.01}{n} = \frac{.01}{2} = .005$$

$$\epsilon_u = 10., \quad \epsilon_6 = 10(.005) = .05$$

Step 5.1

$$h_1 = 1.5, \quad h_2 = 2.5$$

Step 5.2

$$H = [(1.5)^2 + (2.5)^2]^{1/2} = 2.9$$

Step 5.3

$$z_1 = A_{11}h_1 + A_{12}h_2 = 1.5 + 2.5 = 4.0$$

$$k_1 = -4$$

Step 5.4

$$S = (1.5)(1.5) + (2.5)(2.5) + 0(-4) = 8.5$$

$$S = 8.5 > 0.$$

Step 6.1

$$\theta'_{11} = \frac{0.0-0.5}{-4.0} = 0.125 \text{ and } \theta'_{12} = 0.125$$

Step 6.2

$$\theta'_{21} = \frac{10.-0.5}{1.5} = 6.3$$

$$\theta'_{22} = \frac{10.-0.5}{2.5} = 3.8$$

Therefore

$$\theta' = \min [6.3, 3.8] = 3.8$$

Step 6.3

$$\theta' = \min [0.125, 3.8] = 0.125$$

$$x1_1 = 0.5 + 1.5(0.125) = 0.6875$$

$$x2_2 = 0.5 + 2.5(0.125) = 0.8125$$

$$y1_1 = 0.5 + (-4)(0.125) = 0.0$$

And

$$\phi_0^1 = 3.2461$$

Step 6.4.2

$$\theta'' = \frac{\theta'}{2} = \frac{0.125}{2} = 0.0625$$

$$x2_1 = 0.5 + 1.5(0.0625) = 0.59375$$

$$x2_2 = 0.5 + 2.5(0.0625) = 0.65625$$

$$y2_1 = 0.5 + (-4)(0.0625) = 0.25$$

And

$$\phi_0^2 = 2.7646$$

Step 6.4.3

$$F1 = [3.2461 + 3(2.25) - 4(2.7646)](0.0625)$$

$$= -0.065$$

$$F2 = 2[3.2461 + 2.25 - 2(2.7646)]$$

$$= 0.1338$$

$$F = \frac{F1}{F2} = -0.48 < 0$$

Step 6.4.4

$$\text{PHI} = \max(3.2461, 2.7646) = 3.2461$$

$$\text{And } \theta = 0.125$$

$$\text{PHI} > \phi_0^0 \text{ then go to step 7}$$

Step 7

$$x1_1 = 0.6875$$

$$x1_2 = 0.8125$$

$$y1_1 = 0.0$$

Step 7.1

$$v_1 = 0.14281$$

$$R0 = 0.14281 > \varepsilon_0 = 0.01 \text{ the constraint is not satisfied.}$$

Step 7.2

$$\text{DEL}_1 = -0.14281$$

$$y1_1 = 0. - 0.14281 = -0.14281$$

The value of $y1_1$ is less than it's lower bound so, we
reduce the value of θ

$$\theta = \frac{\theta}{2} = \frac{0.125}{2} = 0.0625$$

Step 7

$$x1_1 = 0.5 + 1.5(0.0625) = 0.50375$$

$$x_{1_2} = 0.5 + 2.5(0.0625) = 0.65625$$

$$y_{1_1} = 0.5 + (-4)(0.0625) = 0.25$$

Step 7.1

$v_1 = 0.043272 > \varepsilon_0 = 0.01$, the constraint is not satisfied.

$$RO = .043772$$

Step 7.2

$$DEL_1 = -0.043$$

$$y_{1_1} = .25 - 0.043 = 0.027 > a_{2_1} = 0.$$

Step 8

$$PHI = 2.7646 > \phi_0^0 = 2.25$$

Then:

$$x_1 = 0.05375$$

$$x_2 = 0.65625$$

$$y_1 = 0.207$$

Then the next iteration starts. Table 1 summarized the numerical calculation of this example.

IT	$\bar{X}^0$	$\bar{Y}^0$	Γ	$\Delta\Phi$	Z	Φ_0	$\bar{X}^1$	$\bar{Y}^1$	Φ_1
1	$\begin{bmatrix} X1=0.5 \\ X2=0.5 \end{bmatrix}$	$\begin{bmatrix} Y1=0.5 \end{bmatrix}$	2.9	38.	37.	2.2500	$\begin{bmatrix} X1=0.5937 \\ X2=0.6562 \end{bmatrix}$	$\begin{bmatrix} Y1=0.207 \end{bmatrix}$	2.7646
2	$\begin{bmatrix} X1=0.5937 \\ X2=0.6562 \end{bmatrix}$	$\begin{bmatrix} Y1=0.207 \end{bmatrix}$	2.7	35.	22.	2.7646	$\begin{bmatrix} X1=0.6252 \\ X2=0.7088 \end{bmatrix}$	$\begin{bmatrix} Y1=0.106 \end{bmatrix}$	2.9303
3	$\begin{bmatrix} X1=0.6252 \\ X2=0.7088 \end{bmatrix}$	$\begin{bmatrix} Y1=0.106 \end{bmatrix}$	2.7	34.	21.	2.9303	$\begin{bmatrix} X1=0.6400 \\ X2=0.7334 \end{bmatrix}$	$\begin{bmatrix} Y1=0.052 \end{bmatrix}$	3.0065
4	$\begin{bmatrix} X1=0.6400 \\ X2=0.7334 \end{bmatrix}$	$\begin{bmatrix} Y1=0.052 \end{bmatrix}$	2.6	34.	21.	3.0065	$\begin{bmatrix} X1=0.6541 \\ X2=0.7563 \end{bmatrix}$	$\begin{bmatrix} Y1=0.0 \end{bmatrix}$	3.0774
5	$\begin{bmatrix} X1=0.6541 \\ X2=0.7563 \end{bmatrix}$	$\begin{bmatrix} Y1=0.0 \end{bmatrix}$	0.6	0.4	0.4	3.0774	$\begin{bmatrix} X1=0.5534 \\ X2=0.8328 \end{bmatrix}$	$\begin{bmatrix} Y1=0.0 \end{bmatrix}$	3.1055
6	$\begin{bmatrix} X1=0.5534 \\ X2=0.8328 \end{bmatrix}$	$\begin{bmatrix} Y1=0.0 \end{bmatrix}$	0.006	0.06	0.06	3.1055	$\begin{bmatrix} X1=0.5550 \\ X2=0.8318 \end{bmatrix}$	$\begin{bmatrix} Y1=0.0 \end{bmatrix}$	3.1055

Table 1. Numerical results of example 1.

APPENDIX 2

Computer Programs for Numerical Examples in Chapter 3

Example 1 of Chapter 3

Problem:

maximize

$$f_0(\bar{X}) = (2x_1 - \frac{1}{2} x_2^2) + (3x_2 - \frac{1}{2} x_2^2)$$

subject to

$$f_1(\bar{X}) = x_1^2 + x_2^2 - 1 \leq 0$$

$$x_i \geq 0, \quad i = 1, 2$$

Computer program and the results of this problem is presented in the following pages (p 90 through p 98)

PAGE 0001

```

FORTRAN IV G LEVEL 21          PHIX          DATE = 73111          05/53/50          GEGK 10
0001      SUBROUTINE PHIX          GEGK 20
0002      DIMENSION R(100)          GEGK 30
0003      1,Y(150),C(150),VC(50),IRAS(50),IHR(100),IVC(50),IVA(100)
0004      DOUBLE PRECISION A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFI,DTP,DELTA
          1X,O1,D,EPSIL,EPSILO,ETA,EPSIL1,EPSIL2,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
          2EPSILU,EPSIL6,EPSIL9,EPSI1,F11,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
          3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
          4RAPG,RJB,RQ,RB,RA,SCAL,TR,TD,TC,TREN,TRK,TRL,TETA,TETAT,TQTG,TQ
          5TG1,TETAI,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TQO,TET1,TI
0005      DOUBLE PRECISION TRR,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VCI,VCL,XI,
          1XNV,XNU,XSH,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSBI,XSB2,XMK,XIE1,XIE2
          2,XSEL,XSE2,XTR1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSCRT0,YR0,YR
          3,Z,Z1,X,XC,XS,XI,X2,XR,B
          DOUBLE PRECISION DIT,DISOLS
          DOUBLE PRECISION DMOD,DSORT,DABS,DMIN1,DMAX1
          COMMON R,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
          COMMON F11,PHI,PSI,PSI3,TR,TD,TC,EPSIL,EPSILO,EPSIL2,YSCRT
          COMMON NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEGA
          1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA
          2GB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
          COMMON KFOHC,KGRAD,KCDBAL,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
          1REN1,KREN2,KINV,KCDBAL,KREN11,KREN21,IDIREC,JKJ,LC
          PHI=2.00*XC(1)-0.500*XC(1)**2+3.00*XC(2)-0.500*XC(2)**2
          RETURN
          END
0006
0007
0008
0009
0010
0011
0012
0013
0014      GEGK 250
          GEGK 260

```


PAGE 0001

FORTRAN IV G LEVEL 21

DATE = 73111

05/53/50

JACOR

GEGM 10

GEGM 20

GEGM 30

SUBROUTINE JACOR

DIMENSION B(100)

1,Y(150),C(150),VC(150),IBAS(50),IHB(100),IVC(150),IVA(100)

DOUBLE PRECISION A,ALFA,ALC,ALA,ALB,TRA,C,DELTF1,DELTF2,DELTA

1X,D1,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,

2EPSILU,EPSIL6,EPSIL9,EPS,EPS11,F11,F10,F20,F1,F2,G,GAMA,GNORM,GKS,

3CK,PHI,PSI,PSI3,PC,PHI0,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,

4RAPG,ROB,RO,RE,RA,SCAL,TR,TD,TC,TREN,TRE,TR,TRI,TETA,TETAT,TQTG,TQ

5TG1,TETAL,TET2,TET3,TET4,TET5,TET,TET2,TQ,TQ1,TQ0,TET1,T1

DOUBLE PRECISION TRR,T2,T22,TEX,TE,U,VC,VU,V2,V1,V3,V,VCL,XI,

1XNV,XNO,XSB,XSA,XNORM,XNOR,XINDRM,XMAJ,XSC,XSBI,XSB2,XMK,XIE1,XIE2

2,XSEL,XSE2,XTR1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORT0,YRO,YR

3,Z,Z1,X,XC,XS,X1,X2,XR,B

DOUBLE PRECISION DIT,DISOLS

DOUBLE PRECISION OMUR,DSORT,DABS,DMIN1,DMAX1

COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTF1,ETA

COMMON F11,PHI,PSI,PSI3,IB,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT

COMMON IV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEGA

1N4,NVN1N1,NVNIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA

2DR,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB

COMMON KFONC,KGRAD,KCGNT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEA

1REN1,KREN2,KINV,KCDBAL,KREN11,KREN21,IDIREC,JKO,LC

A(1,1)=2.D0*XC(2)

A(1,2)=2.D0*XC(2)

RETURN

END

0001

0002

0003

0004

0005

0006

0007

0008

0009

0010

0011

0012

0013

0014

0015

GEGM 180

GEGM 190

PARAMETERS

NV	2
NIN	1
NEG	0
NEVL	20
NT0	6
ITET	20
ICONJ	0
IDIAG	0
ITMAX	50
KFIL	0
KLIN	0
NCO	10
ITSOR	1
ISOLSOR	50
EPSIL	0.1D 00
EPSILO	0.1D-06
EPSILI	0.1D-04
EPSIL2	0.1D-04
PC	0.1D 01

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 2
 TOTAL NUMBER OF VARIABLES 3
 NUMBER OF CONSTRAINTS 1
 EPSILON FOR NEWTON'S METHOD 0.1000D-06
 EPSILON FOR GRADIENT TEST 0.1000D-04

OBJECTIVE FUNCTION 0.225000000000000D 01

	LOWER BOUND	STARTING POINT	UPPER BOUND
X(1)	0.0	0.500000000000000D 00	0.100000000000000D 02
X(2)	0.0	0.500000000000000D 00	0.100000000000000D 02

VALUE OF CONSTRAINTS

C(1) -0.500000000000000D 00

IT 1	PHI 0.276464843750000 01	DIR. GRAD.	NO 0	YN 0.290 01	DELTA FI 0.380 02	ETA 0.240 02	NCDB 0	NCN 2	NITN 2
IT 2	PHI 0.307512739833300 01	DIR. GRAD.	NO 0	YN 0.270 01	DELTA FI 0.350 02	ETA 0.220 02	NCDB 0	NCN 2	NITN 2
IT 3	PHI 0.307740328134170 01	DIR. GRAD.	NO 0	YN 0.260 01	DELTA FI 0.330 02	ETA 0.210 02	NCDB 0	NCN 2	NITN 2
IT 4	PHI 0.307751861152010 01	DIR. GRAD.	NO 0	YN 0.260 01	DELTA FI 0.330 02	ETA 0.210 02	NCDB 1	NCN 3	NITN 4
IT 5	PHI 0.310553814386000 01	DIR. GRAD.	NO 1	YN 0.900 00	DELTA FI 0.590 00	ETA 0.590 00	NCDB 0	NCN 3	NITN 20
IT 6	PHI 0.310553822138260 01	DIR. GRAD.	NO 1	YN 0.720 00	DELTA FI 0.400 00	ETA 0.660 01	NCDB 0	NCN 21	NITN167
	PHI 0.310553822138260 01			YN 0.720 00	DELTA FI 0.400 00	ETA 0.660 01			

EXECUTION TIME 613 CENTISECONDES

PRIMAL VARIABLE	ASSOCIATED REDUCED GRADIENT VALUE
X(1) = 0.55246026171653D 00	V(1) = 0.
X(2) = 0.83353923106716D 00	V(2) = 0.71892103064937D 00
CONSTRAINT	LAGRANGE MULTIPLIER
CONTRAINTE 1 -0.94960673074906D-08	U(1) = 0.86830930346867D 00

STOPPING CONDITION	7		
NUMBER OF ITERATIONS	6		
NUMBER OF TIMES THE CONSTRAINTS WERE CALCULATED	806		
NUMBER OF TIMES THE OBJECTIVE FUNCTION WAS CALCULATED	218		
NUMBER OF TIMES THE GRADIENT WAS CALCULATED	10		
NUMBER OF INVERSE CALCULATIONS WITH THE GIVEN BASE	3		
NUMBER OF INVERSE CALCULATIONS WITH ANOTHER BASE	4		
NUMBER OF BASE CHANGES	1		
NUMBER OF TIMES THE JACOBIEN OF THE CONSTRAINTS WAS CALCULATED	8		
NUMBER OF MAXIMUMS FOUND WITHOUT BISECTION	18		
NUMBER OF MAXIMUMS FOUND BY BISECTION	0		
NUMBER OF REENTRIES IN TO THE FEASIBLE REGION	101		
TOTAL NUMBER OF NEWTON,S ITERATIONS			723

Example 2 of Chapter 3

Problem:

maximize

$$R_5 = 1 - R_3((1-R_1)(1-R_4))^2 - (1-R_3)(1-R_2(1-(1-R_1)(1-R_4)))^2$$

subject to

$$K_1 R_1^{\alpha_1} + K_2 R_2^{\alpha_2} + K_3 R_3^{\alpha_3} + K_4 R_4^{\alpha_4} \leq C$$

$$R_i \geq R_{i,\min}, \quad i = 1, 2, 3, 4$$

Computer program and the results of this problem is presented in the following pages (p100through p108).

PAGE 0001

```

FORTRAN IV G LEVEL 21          CPHI          DATE = 73060          03/22/00          GEGl 10
0001      SURROUTINE CPHI
0002      DIMENSION A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)          GEGl 20
0003      1,X(150),C(150),VC(150),THAS(50),THR(100),IVC(50),IVA(100)          GEGl 30
0004      DOUBLE PRECISION A,ALFA,ALC,ALA,ALA,IPA,C,DELTF1,DELTF2,DELTF3,DELTF4,DTP,DELTA
1X,O1,D,EPSIL,EPSILO,EPSIL2,EYA,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
2EPSILU,EPSIL6,EPSIL9,EPS,EPS11,F11,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSI,TQP,PSI,PENAL,PE1,PE2,PE3,
4RAG,ROR,PO,PR,RA,SCAL,TR,TD,TC,TREN,TRE,TR,TRI,TETA,TETAT,TQIG,TQ
5IG1,TETAL,TET2,TET3,TET4,TET5,TET,TFT2,TQ,TQ1,TQ0,TET1,T1
0005      DOUBLE PRECISION TRB,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VCI,VCL,XI,
1XNV,XNO,XS,XSA,XNORM,XNDR,XINORM,XMAJ,XSC,XSRI,XSB2,XMK,XIE1,XIE2
2,XSEI,XSE2,XTR1,XTR,XTR2,XI,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YR0,YR
3,Z,Z1,X,XC,XS,X1,X2,XR,B
0006      DOUBLE PRECISION OIT,DISOLS
0007      DOUBLE PRECISION DMOD,DSORT,DABS,DMINI,DMAX1
0008      COMMON B,A,ALFA,A,XC,XI,XS,Y,C,VC,DELTF1,ETA
0009      COMMON F11,PHI,PSI,PSI3,TR,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
0010      COMMON NV,NC,NK,NEG,NIN,NTV,NV1,REV,NEVL,NTQ,NIN1,NIN2,NIN3,NIGEGA 50
1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IBP,ICDR,JCDR,KCEGA 60
2DB,KFIL,KLIN,KREN,KD,IBAS,IHR,IVC,IVA,IVR
COMMON KFGNC,KGRAD,KCONT,KINVI,KINV2,KCDRA,KJACO,KMAX1,KMAX2,KGEGA 80
1REN1,KREN2,KINV,KCDRAL,KREN11,KREN21,IOIREG,JKO,LC
VC(1)=200.D0*XC(1)**0.6D0+200.D0*XC(2)**0.6D0+200.D0*XC(3)**0.6D0+
1300.D0*XC(4)**0.6D0-800.D0
RETURN
END
GEGl 570
GEGl 580

```


PAGE 0001

```

FORTRAN IV G LEVEL 21          GRADFI          DATE = 73060          03/22/00          GEGN 10
0001      SUBROUTINE GRADFI          GEGN 20
0002      DIMENSION A(100)          GEGN 30
0003      1,Y(150),C(150),ALFA(50),X(150),XI(150),XS(150),IVAL(100)
0004      DOUBLE PRECISION A,ALFA,ALC,ALA,ALB,TRA,C,DELTA,DELTA,DELTA,DELTA,DELTA
          1X,D1,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
          2EPSILU,EPSIL6,EPSIL9,EPSI1,F11,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
          3GK,PHI,PSI,PSI3,PC,PHI0,PHI1,PHI2,PSI,TOP,PSI4,PENAL,PE1,PE2,PE3,
          4RPG,ROR,RQ,RR,RA,SCAL,TR,TD,TC,TREN,TPE,TR,TRI,TETA,TETAT,TOTG,TQ
          5TGL,TETAL,TET2,TET3,TET4,TET5,TET,TET2,TQ,TQ1,TQ0,TEI1,TI
          6DOUBLE PRECISION TRR,T2,T22,TEX,TE,J,VC,VO,V2,VI,V3,V,VC1,VCL,XI,
          7XNV,XND,XSR,XSA,XNORM,XNOR,XINDRM,XMAJ,XSC,XS81,XS82,XMK,XIE1,XIE2
          8XSEL,XSEF2,XTP1,XTR,XTR2,XI,XIT,Y,YSDRT,YSDRT1,YNORM,YSDRT0,YRO,YR
          9Z,ZI,X,XC,XS,XI,X2,XR,R
          0DOUBLE PRECISION DIT,DISOLS
          1DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
          2COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTA,ETA
          3COMMON F11,PHI,PSI,PSI3,TR,TD,TC,EPSIL,EPSILO,EPSIL2,YSDRT
          4COMMON NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NFVL,NT0,NIN1,NIN2,NIN3,NIGEGA
          5IN4,NVIN1,NVNTN2,NVIN3,INDEX,I1,IR,IR1,IS,ISI,IT,I8P,ICDB,JCDB,KCGEGA
          6ZDB,KFIL,KLIN,KREN,KD,IPAS,IHR,IVC,IYA,IVR
          7COMMON KCONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
          8IREN1,KREN2,KINV,KCDBAL,KREN11,KREN21,IOIREC,JKO,LC
          9B(1)=1.D0-XC(1)
          0B(2)=1.D0-XC(3)
          1B(3)=1.D0-XC(4)
          2C(1)=2.D0*XC(3)*B(1)*B(3)*B(3)+2.D0*B(2)*(1.D0-XC(2))*(1.D0-B(1))*B(
          3))*XC(2)*B(3)
          4C(2)=2.D0*B(2)*(1.D0-XC(2))*(1.D0-B(1))*B(3))*(1.D0-B(1))*B(3))
          5C(3)=-B(1)*B(3)**2+(1.D0-XC(2))*(1.D0-B(1))*B(3))**2
          6C(4)=2.D0*XC(3)*B(1)*B(3)*B(1)+2.D0*B(2)*(1.D0-XC(2))*(1.D0-B(1))*B(
          713))*XC(2)*B(1)
          8RETURN
          9END
0019      GEGN 280
0020      GEGN 290

```

PARAMETERS

NV	4
NIN	1
NEG	0
NFVL	20
NT0	6
TTET	20
TCONJ	1
IDIAG	0
ITMAX	50
KFIL	0
KLIN	0
NCO	10
ITSOR	1
ISOLSOR	50
EPSIL	0.10 00
EPSILO	0.10-05
EPSILI	0.10-05
EPSIL2	0.10-05
PC	0.10 01

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 4
 TOTAL NUMBER OF VARIABLES 5
 NUMBER OF CONSTRAINTS 1
 EPSILON FOR NEWTON'S METHOD 0.1000D-05
 EPSILON FOR GRADIENT TEST 0.1000D-05

OBJECTIVE FUNCTION 0.8862336000000000 00

	LOWER ROUND	STARTING POINT	UPPER BOUND
X(1)	0.5000000000000000 00	0.6000000000000000 00	0.1000000000000000 01
X(2)	0.5000000000000000 00	0.6000000000000000 00	0.1000000000000000 01
X(3)	0.5000000000000000 00	0.6000000000000000 00	0.1000000000000000 01
X(4)	0.5000000000000000 00	0.6000000000000000 00	0.1000000000000000 01

VALUE OF CONSTRAINTS

C(1) -0.13758026946395D 03

IT 1	PHI 0.99400900747759D 00	DIR. GRAD.	NO 0	YN 0.47D 00	DELTA FI 0.36D 00	ETA 0.13D 00	NCDB 0	NCN 2	NITN 2
IT 2	PHI 0.99592933988700D 00	DIR. CONJ.	LC 4				NCDB 0	NCN 2	NITN 2
IT 3	PHI 0.99593604408432D 00	DIR. CONJ.	LC 4				NCDB 0	NCN 2	NITN 1
IT 4	PHI 0.99671324495106D 00	ITERATION	SPECIALE	BASE DEGENEREE			NCDB 1	NCN 1	NITN 4
IT 5	PHI 0.100000000000000D 01	DIR. GRAD.	NO 2	YN 0.21D-01	DELTA FI 0.81D-02	ETA 0.52D-02	NCDB 0	NCN 1	NITN 5
	PHI 0.100000000000000D 01			YN 0.0	DELTA FI 0.0	ETA 0.0			

EXECUTION TIME 123 CENTISECONDES

ASSOCIATED REDUCED GRADIENT VALUE

PRIMAL VARIABLE

$x(1) = 0.100000000000000001 = xs(1)$
 $x(2) = 0.100000000000000001 = xs(2)$
 $x(3) = 0.53944883753526000$
 $x(4) = 0.79741557422451000$

LAGRANGE MULTIPLIER

CONSTRAINT

$u(1) = 0.0$

CONTRAINTE 1 -0.106921152109860-05

STOPPING CONDITION	1		
NUMBER OF ITERATIONS	5		
NUMBER OF TIMES THE CONSTRAINTS WERE CALCULATED	20		
NUMBER OF TIMES THE OBJECTIVE FUNCTION WAS CALCULATED	19		
NUMBER OF TIMES THE GRADIENT WAS CALCULATED	8		
NUMBER OF INVERSE CALCULATIONS WITH THE GIVEN BASE	2		
NUMBER OF INVERSE CALCULATIONS WITH ANOTHER BASE	0		
NUMBER OF BASE CHANGES	1		
NUMBER OF TIMES THE JACOBIEN OF THE CONSTRAINTS WAS CALCULATED	7		
NUMBER OF MAXIMUMS FOUND WITHOUT BISECTION	5		
NUMBER OF MAXIMUMS FOUND BY BISECTION	0		
NUMBER OF REENTRIES IN TO THE FEASIBLE REGION	8		
TOTAL NUMBER OF NEWTON,S ITERATIONS	14		

Example 3 of Chapter 3

Problem:

$$f_0(\bar{X}) = A_1 x_3^2 + A_2 x_1 x_5 + A_3 x_1 - A_4$$

subject to

$$\begin{aligned} B_1 + B_2 x_2 x_5 + B_3 x_1 x_4 - B_4 x_1 x_3 &\geq 0 \\ &\leq 92 \end{aligned}$$

$$\begin{aligned} C_1 + C_2 x_2 x_5 + C_3 x_1 x_2 + C_4 x_3^2 &\geq 90 \\ &\leq 110 \end{aligned}$$

$$\begin{aligned} D_1 + D_2 x_3 x_5 + D_3 x_1 x_3 + D_4 x_3 x_4 &\geq 20 \\ &\leq 25 \end{aligned}$$

$$78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45$$

$$27 \leq x_3, x_4, x_5 \leq 45$$

Computer program and the results of this problem is presented in the following pages (p110through p118)

06/01/18

DATE = 73111

CPH1

FORTRAN IV G LEVEL 71

```

0001 SUBROUTINE CPH1
0002 DIMENSION R(100)
0003 1, Y(150), C(150), VC(50), IRAS(50), IH3(100), IVC(50), IVA(100)
0004 DOUBLE PRECISION A, ALFA, ALC, ALA, ALR, TRA, C, DELTFA, DTP, DELTA
0005 1X, D1, EPSIL, EPSILO, EPSIL2, ETA, EPSIL3, EPSIL4, EPSIL5, EPSIL7,
0006 2X, EPSILO, EPSIL9, EPSI10, EPSI11, EPSI12, EPSI13, EPSI14, EPSI15, EPSI16, EPSI17,
0007 3X, PH1, PSI, PSI3, PC, PH10, PH11, PH12, PSIT, TQP, PSI4, PENAL, PEL, PE2, PE3,
0008 4RPG, KQB, RD, RA, RA, SCAL, FB, TD, TC, TREN, TPE, TR, TRI, TETA, TETAI, TOTG, TO
0009 5TGI, TETAI, TET2, TET3, TET4, TET5, TETI, TETA2, TQ, TQ1, TQ2, TET1, T1
0010 DOUBLE PRECISION TR0, T2, T22, TEX, TEU, VC, VU, V2, VI, V3, V, VCI, VCL, XI,
0011 1XNV, XNU, XSO, XSA, XNORM, XNUR, XINORM, XMAJ, XSC, XSRI, XSB2, XMK, XIE1, XIE2
0012 2, XSE1, XSE2, XTR1, XTR2, XT, XIT, Y, YSUPT, YSUPT1, YNORM, YSORTO, YRO, YR
0013 3, Z, Z1, Z, XC, XS, X1, X2, XR, R
0014 DOUBLE PRECISION DIT, DISOLS
0015 DOUBLE PRECISION DMUD, DARS, DMINI, DMAX1
0016 COMMON B, A, ALFA, X, XC, XI, XS, Y, C, VC, DELTFA, ETA
0017 COMMON F11, PH1, PSI, PSI3, TR, TD, TC, EPSIL, EPSILO, EPSIL2, YSORT
0018 COMMON NV, NC, NK, NEG, NIN, NIV, NV1, NEV, NEVL, NTO, NINI, NIN2, NIN3, NIGEGA
0019 1N4, NVNINI, NVNIN2, NVNIN3, INDEX, I1, IR, IRI, IS, IS1, IT, IBP, ICDB, JCDB, KCGEA
0020 2DH, KFIL, KLIN, KREN, KD, IRAS, IHR, IVC, IVA, IVB
0021 COMMON KFOC, KGRAD, KCONT, KINVI, KINV2, KCDBA, KJACO, KMAX1, KMAX2, KGEA
0022 1REN1, KREN2, KINV, KCDBA1, KREN11, KREN21, IDIREC, JKO, LC
0023 IF (IT) 100, 110, 110
0024 100 B(5)=0.853344102
0025 B(6)=0.00568600
0026 R(7)=0.00062600
0027 B(8)=0.00220500
0028 B(9)=0.805124902
0029 R(10)=0.00713200
0030 R(11)=0.00299600
0031 R(12)=0.00218100
0032 B(13)=0.93096101
0033 B(14)=0.00476300
0034 B(15)=0.0125500
0035 B(16)=0.00190800
0036 110 VC(1)=3(5)+D(6)*XC(2)+XC(5)+8(7)*XC(1)*XC(4)+8(8)*XC(3)*XC(5)
0037 VC(2)=-VC(1)
0038 VC(11)=VC(1)-92.00
0039 VC(3)=4(9)+R(10)*XC(2)+XC(5)+B(11)*XC(1)*XC(2)+B(12)*XC(3)+*2
0040 VC(4)=-VC(3)+90.00
0041 VC(3)=VC(3)-110.00
0042 VC(5)=B(13)+B(14)*XC(3)+XC(5)+B(15)+XC(1)*XC(3)+B(16)+XC(3)*XC(4)
0043 VC(6)=-VC(5)+20.00
0044 VC(5)=VC(5)-25.00
0045 RETURN
0046 END
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200

```


PARAMETERS	
NV	5
NIN	6
NLG	0
NFVL	20
NT0	6
ITET	20
ICCNJ	1
IDTAG	0
ITMAX	50
KFIL	0
KLIN	0
NGO	10
ITSOR	1
ISOLSOR	50
EPSIL	0.10 00
EPSIL0	0.10-04
EPSIL1	0.10-02
EPSIL2	0.10-02
PC	0.0

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 5
 TOTAL NUMBER OF VARIABLES 11
 NUMBER OF CONSTRAINTS 6
 EPSILON FOR NEWTON'S METHOD 0.1000D-04
 EPSILON FOR GRADIENT TEST 0.1000D-02

OBJECTIVE FUNCTION 0.46606719911006D 01

	LOWER BOUND	STARTING POINT	UPPER BOUND
X(1)	0.78000000000000D 02	0.78610000000000D 02	XS(1)
X(2)	0.33000000000000D 02	0.33430000000000D 02	XS(2)
X(3)	0.27000000000000D 02	0.31060000000000D 02	XS(3)
X(4)	0.27000000000000D 02	0.44090000000000D 02	XS(4)
X(5)	0.27000000000000D 02	0.35210000000000D 02	XS(5)

VALUE OF CONSTRAINTS

C(1) -0.21454337980001D 00
 C(2) -0.91795456620200D 02
 C(3) -0.11115297278000D 02
 C(4) -0.88847027220000D 01
 C(5) -0.48786077860000D 01
 C(6) -0.12139221399999D 00

IT 1	PHI 0.11736756393584D 03	DIR.GRAD.	NO 2	YN 0.35D 03	DELTA FI 0.19D 04	ETA 0.14D 04	NCDB 0	NCN 2	NITN 2
IT 2	PHI 0.11746680177866D 03	DIR.CONJ.	LC 3	BASE DEGENEREE			NCDB 0	NCN 2	NITN 1
IT 3	PHI 0.12252417813077D 03	ITERATION	SPECIALE				NCDB 1	NCN 1	NITN 6
IT 4	PHI 0.2541435325688D 03	DIR.GRAD.	NO 2	YN 0.95D 02	DELTA FI 0.76D 03	ETA 0.69D 03	NCDB 1	NCN 5	NITN 8
IT 5	PHI 0.29201503819868D 03	DIR.GRAD.	NO 3	YN 0.98D 02	DELTA FI 0.38D 02	ETA 0.37D 02	NCDB 0	NCN 7	NITN 13
	PHI 0.29201500819868D 03			YN 0.0	DELTA FI 0.0	ETA 0.0			

EXECUTION TIME 251 CENTISECONDES

PRIMAL VARIABLE

X(1) = 0.7800000000000000 02 = X1(1)
 X(2) = 0.3300000000000000 02 = X1(2)
 X(3) = 0.299939426347300 02
 X(4) = 0.4500000000000000 02 = X5(4)
 X(5) = 0.367758856648170 02

ASSOCIATED REDUCED GRADIENT VALUE

V(1) = -0.489174205531080 02
 V(2) = -0.843107323443360 02
 V(3) = 0.
 V(4) = 0.266323969121650 02
 V(5) = 0.

CONSTRAINT

CONTRAINTE 1 -0.100031580529960-04
 CONTRAINTE 2 -0.51959899568420 02
 CONTRAINTE 3 -0.111582732355590 02
 CONTRAINTE 4 -0.88417267644120 01
 CONTRAINTE 5 -0.49998999326430 01
 CONTRAINTE 6 -0.100067357493570-04

LAGRANGE MULTIPLIER

U(1) = 0.403192890493630 03
 U(2) = 0.0
 U(3) = 0.0
 U(4) = 0.0
 U(5) = 0.0
 U(6) = 0.809378995961580 03

STOPPING CONDITION	1	
NUMBER OF ITERATIONS	5	
NUMBER OF TIMES THE CONSTRAINTS WERE CALCULATED	40	
NUMBER OF TIMES THE OBJECTIVE FUNCTION WAS CALCULATED	36	
NUMBER OF TIMES THE GRADIENT WAS CALCULATED	16	
NUMBER OF INVERSE CALCULATIONS WITH THE GIVEN BASE	4	
NUMBER OF INVERSE CALCULATIONS WITH ANOTHER BASE	0	
NUMBER OF BASE CHANGES	2	
NUMBER OF TIMES THE JACOBIEN OF THE CONSTRAINTS WAS CALCULATED	8	
NUMBER OF MAXIMUMS FOUND WITHOUT BISECTION	5	
NUMBER OF MAXIMUMS FOUND BY BISECTION	0	
NUMBER OF REENTRIES IN TO THE FEASIBLE REGION	17	
TOTAL NUMBER OF NEWTON,S ITERATIONS	24	

Example 4 of Chapter 3

Problem:

maximize

$$f_0(\bar{X}) = \sum_{i=1}^{10} b_i x_{i+5} - \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_i x_j - 2 \sum_{j=1}^5 d_j x_j^3$$

subject to

$$f_i(\bar{X}) = \sum_{j=1}^{10} a_{i,j+5} x_{j+5} - e_i - 2 \sum_{j=1}^5 c_{ji} x_j - 3d_i x_i^2 \leq 0$$

$$i = 1, 2, 3$$

$$f_i(\bar{X}) = \sum_{j=1}^{10} a_{i,j+5} x_{j+5} - e_i - 2 \sum_{j=1}^5 c_{ji} x_j - 3d_i x_i^2 + x_{i+12} = 0$$

$$i = 4, 5$$

$$0 \leq x_k \leq 100, \quad k = 1, \dots, 17$$

Computer program and the results of this problem is presented in the following pages (p120through p129).

```

0001 SUBROUTINE PHIX
0002 DIMENSION B(100)
0003
0004 DIMENSION A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
      1,XI(50),CI(50),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)
      DOUBLE PRECISION A,ALFA,ALC,ALA,ALR,TRA,C,DELTA,DELTA,DTP,DELTA
      1X,D1,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
      2EPSIL6,EPSIL9,EPS,EPS1,FIL,FLO,F20,F1,F2,G,GAMA,GNORM,GKS,
      3GK,PHI,PSI,PSI3,PC,PHIO,PHIL,PHI2,PSIT,TOP,PSI4,PENAL,PEL,PE2,PE3,
      4RAPG,ROR,RD,RB,RA,SCAL,TQ,TD,TC,TREN,TRE,TR,TSL,TETA,TETAT,TOTG,TQ
      5TG1,TETAL,TET2,TET3,TET4,TET5,TET,TETAZ,TQ,TOL,TQQ,TET1,TI
      DOUBLE PRECISION TRB,I2,I22,TEX,TF,U,VCAVO,VZ,VI,V3,V,VC1,VCL,XI,
      1XNV,XNO,XSB,XSA,XNDRM,XNDR,XINPRM,XMAJ,XSC,XSRI,XSR2,XMK,XIE1,XIE2
      2,XSEL,XSF2,XTRI,XTR,XTR2,XT,XIT,Y,YSORT,YSORTI,YNDRM,YSDORTO,YRO,YR
      3,Z,Z1,X,XC,XS,XI,X2,XP,B
      DOUBLE PRECISION DIT,DITSOLS
      DOUBLE PRECISION DMOD,DSORT,DABS,DMIN1,DMAX1
      COMMON R,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTA,ETA
      COMMON FILL,PHI,PSI,PSI3,TQ,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
      CMMON NV,NC,NK,NEG,NIN,NTV,NVL,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEGA
      1NG,NVIN1,NVIN2,NVIN3,INDEX,I,IIR,Irl,IS,ISI,IT,IBP,ICDR,JCD8,KCEGA
      20,KFIL,KLIN,KREN,KDI,RAS,IHB,IVC,IVA,IVB
      CMMON KCONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEKA
      1KEN1,KREN2,KINV,KCDRA1,KREN11,KREN21,IDIRECT,KJO,LC
      IF (IT.GE.0) GO TO 100
      R(1)=-40.00
      R(2)=-2.00
      R(3)=-0.2500
      R(4)=-6.00
      R(5)=-4.00
      R(6)=-1.00
      R(7)=-40.00
      R(8)=-60.00
      R(9)=6.00
      R(10)=1.00
100 CONTINUE
      PHI = -8.D0*XC(1)**3-16.D0*XC(2)**3-20.D0*XC(3)**3-12.D0*XC(4)**3-GEK 170
      14.D0*XC(5)**3-30.D0*XC(1)**2-39.D0*XC(2)**2-10.D0*XC(3)**2-39.D0*XGFGK 180
      2C(4)**2-30.D0*XC(5)**2+40.D0*XC(1)*XC(2)+20.D0*XC(1)*XC(3)-64.D0*XGEGK 190
      3C(1)*XC(4)+20.D0*XC(1)*XC(5)+12.D0*XC(2)*XC(3)+62.D0*XC(2)*XC(4)-6GEGK 200
      44.D0*XC(2)*XC(5)+12.D0*XC(3)*XC(4)+20.D0*XC(3)*XC(5)+40.D0*XGEGK 210
      5C(4)*XC(5)
      DO 20 I=6,15
        PHI = PHI+B(I-5)*XC(I)
20 CONTINUE
      RETURN
      END
0025
0026
0027
0028
0029

```

PAGE 0001

03/24/07

DATE = 73114

СРНИ

FOPTAN IV C LEVEL 21

0001	SUPROUTINE CPHI		GEGL 10
0002	DIMENSION B(100)		GEGL 20
0003	1,Y(150),VC(50),IBAS(50),IHR(100),IVC(50),IVA(100)		GEGL 30
0004	DOUBLE PRECISION A,ALFA,ALCA,ALB,TRA,C,DELTEI,DELTEA,DTP,DELTA IX,OI,OPSIL,EPSILO,EPSIC2,ETA,EPSIL3,EPSIL4,EPSIL5,EPSIL7, ZEPSIU,EPSIL9,EPSI10,EPSI11,FII,FIG,F20,F1,F2,G,GAMA,GNORM,GKS, 3GK,PHI,PSI3,PE,PHIO,PHI1,PHI2,PSIT,TOP,PSI4,PENAL,PE1,PE2,PE3, GRAPG,ROR,RQ,PR,PA,SCAL,TR,ID,TC,TREN,TRF,IR,IIR,I,ETA,IETAT,TOTG,TQ STGI,TETAI,TET2,TET3,TET4,TET5,TET,TETA2,TOT,TOL,TOD,TETI,TI DOUBLE PRECISION TRA,T2,T22,TEX,TE,U,VG,VU,V2,VI,V3,V,VCI,VCL,XI, 1XNV,XNO,XSA,XSA,XNDNM,XNOP,XINDRM,XMAJ,XSB1,XSR2,XWK,XIE1,XIE2 2,XSEL,XSF2,XTRI,XTR,XTR2,XI,XIT,Y,YSORT,YSORTI,XNDRM,YSORTO,YRO,YR 3,Z,ZI,X,XC,XS,XI,X2,XP,R		
0005	DOUBLE PRECISION DIT,DISOLS		
0006	DOUBLE PRECISION DMOD,DSORT,DABS,DMINI,DMAXI		
0007	COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTEI,ETA		
0008	COMMON FII,PHI,PSI,PSI3,TB,TOT,OPSIL,EPSILO,EPSIL2,YSORT		
0009	COMMON NV,NMC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTG,NINI,NIN2,NIN3,NIGEGA	50	
0010	1N4,NVINI,NVNIN2,NVNIN3,INDEX,IIF,IR,IIR,IIS,I51,IT,I8P,ICDB,JCDR,KCGEGA	60	
0011	2PH,KFIL,KLIN,KREN,KDI,BAS,IHB,IVC,IVA,IVR		
0012	COMMON KEONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAXI,KMAX2,KCEGA	80	
0013	1RCNI,KLEN2,KINV,KCDBAI,KREN11,KREN21,IDIREC,JKO,LC		
0014	IF ((I+GE .0) GO TO 103		
0015	A(1,2) = 40.00	GEGL 110	
0016	A(1,3) = 20.00	GEGL 170	
0017	A(1,4) = -64.00	GEGL 180	
0018	A(1,5) = 20.00	GEGL 190	
0019	A(1,6) = -16.00	GEGL 200	
0020	A(1,8) = -3.500		
0021	A(1,11) = 2.00		
0022	A(1,12) = -1.00		
0023	A(1,13) = -1.00		
0024	A(1,14) = 1.00		
0025	A(1,15) = 1.00		
0026	A(2,1) = 40.00	GEGL 210	
0027	A(2,3) = 12.00	GEGL 220	
0028	A(2,4) = 62.00	GEGL 230	
0029	A(2,5) = -64.00	GEGL 240	
0030	A(2,6) = 2.00		
0031	A(2,7) = -2.00		
0032	A(2,9) = -2.00		
0033	A(2,10) = -9.00		
0034	A(2,12) = -1.00		
0035	A(2,13) = -2.00		
0036	A(2,14) = 2.00		
0037	A(2,15) = 1.00		
0038	A(3,1) = 20.00	GEGL 250	
0039	A(3,2) = 12.00	GEGL 260	
0040	A(3,4) = 12.00	GEGL 270	
0041	A(3,5) = 20.00	GEGL 280	
0042	A(3,8) = 2.00		
0043	A(3,10) = -2.00		
0044	A(3,11) = -4.00		
0045	A(3,12) = -1.00		
0046	A(3,13) = -3.00		
0047	A(3,14) = 3.00		
0048	A(3,15) = 1.00		

```

FORTRAN IV G LEVEL 21                                CPHI                                DATE = 73114                                03/24/07                                PAGE 0002

0047      A(4,1) = -64.00                                GEGL 290
0048      A(4,2) = 62.00                                GEGL 300
0049      A(4,3) = 12.00                                GEGL 310
0050      A(4,5) = 40.00                                GEGL 320
0051      A(4,6) = 1.00
0052      A(4,7) = 0.400
0053      A(4,9) = -4.00
0054      A(4,10) = 1.00
0055      A(4,12) = -1.00
0056      A(4,13) = -2.00
0057      A(4,14) = 4.00
0058      A(4,15) = 1.00
0059      A(4,16) = 1.00
0060      A(5,1) = 20.00
0061      A(5,2) = -64.00
0062      A(5,3) = 20.00
0063      A(5,4) = 40.00
0064      A(5,7) = 2.00
0065      A(5,9) = -1.00
0066      A(5,10) = -2.800
0067      A(5,12) = -1.00
0068      A(5,13) = -1.00
0069      A(5,14) = 5.00
0070      A(5,15) = 1.00
0071      A(5,17) = 1.00
0072
0073      103 CONTINUE
0074      10*XC(5) = 15.00-60.00*XC(1)+40.00*XC(2)+20.00*XC(3)-64.00*XC(4)+20.00*XC(5)-12.00*XC(1)**2
0075      10*XC(5) = 27.00+40.00*XC(1)-78.00*XC(2)+12.00*XC(3)+62.00*XC(4)-64.00*XC(5)-24.00*XC(2)**2
0076      10*XC(5) = 36.00+20.00*XC(1)+12.00*XC(2)-20.00*XC(3)+12.00*XC(4)+20.00*XC(5)-30.00*XC(3)**2
0077      10*XC(5) = 19.00-64.00*XC(1)+62.00*XC(2)+12.00*XC(3)-78.00*XC(4)+40.00*XC(5)-18.00*XC(4)**2
0078      10*XC(5) = 12.00+20.00*XC(1)-64.00*XC(2)+20.00*XC(3)+40.00*XC(4)-60.00*XC(5)-6.00*XC(5)**2
0079      DO 30 I=1,5
0080      R(11)=0.00
0081      DO 25 J=6,15
0082      B(11)=R(11)+A(1,J)*XC(J)
0083      25 CONTINUE
0084      VC(11)=VC(1)+R(11)
0085      30 CONTINUE
0086      VC(4) = VC(4)+XC(16)
0087      VC(5) = VC(5)+XC(17)
0088      RETURN
      END

GEGL 150
GEGL 330
GEGL 340
GEGL 350
GEGL 360

GEGL 160
GEGL 370
GEGL 380
GEGL 390
GEGL 410
GEGL 420
GEGL 430
GEGL 440
GEGL 450
GEGL 460
GEGL 470
GEGL 480

GEGL 500
GEGL 520
GEGL 540
GEGL 560
GEGL 570
GEGL 580

```

```

FORTRAN IV G LEVEL 21          JACOB          DATE = 73114          03/24/07          PAGE 0000

SUBROUTINE JACOR
DIMENSION R(100)
1,Y(150),C(150),VC(150),IHR(100),IVC(50),IVA(100)
DOUBLE PRECISION A,ALEA,AIC,ALA,ALB,TFA,C,DELTEL,DELTFA,DTP,DELIA
1X,OL,D,EPSIL,EPSILO,EPSIL2,FITA,EPSILI,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
2EPSIU,EPSIL6,EPSIL9,FPS,EPSII,FPI,F10,F20,F1,F2,G,GAMA,QNORM,GKS,
3GK,PHI,PS1,PS13,PC,PHIO,PHI1,PHI2,PSIT,TOP,PSI4,PENAL,PE1,PE2,PE3,
4GRG,ROR,B,C,RB,PA,SICAL,IH,TD,TC,TREN,TPE,TR,TRI,TFTA,TETAT,TOIG,TO
5GT1,TETAT,TETZ,TET3,TET4,TET5,TFT,TETA2,TQ,TQT,QOQ,TET1,TI
DOUPLE PRECISION FRR,T2,T22,TEX,TE,U,AVC,V0,V2,V1,V3,V,VCI,VCL,X1,
1XNV,XNO,XSR,XSA,XUIRM,XNDR,XINURM,XMAJ,XSC,XSRI,XSR2,XMK,XIEL,XIE2,
2,XSE1,XSF2,XTH1,XTR,XTR2,XTAXIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
3,Z,Z1,X,XC,XS,X1,X2,XR,n
DOUBLE PRECISION D1,DISOLS
DOUBLE PRECISION DMDD,USORT,DABS,DMINI,DMAXI
COMMON B,A,ALFA,X,KC,XI,XS,Y,C,AVC,DELTFI,ETA
COMMON F11,PHI,PSI,PS13,TR,TD,TC,EPSIL,EPSILO,EPSIL2,YSDRT
COMMON NV,NC,NK,NFG,MIN,NTV,MVL,NEV,NEVL,NTG,NIN1,NIN2,NIN3,NIGEGA
1N4,NVIN1,NVIN2,NVIN3,INOEX,I1,IIR,IP1,IS,IS1,IT,ISP,ICD8,JCD8,KCGEGA
2DR,KFIL,KLIN,KREN,KC,IRAS,IHB,IVC,IVA,IVB
COMMON KFCNC,KCRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KCEGA
1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
A(1,1) = -24.D0*XC(11)-60.D0
A(2,2) = -48.D0*XC(12)-78.D0
A(3,3) = -60.D0*XC(13)-20.D0
A(4,4) = -36.D0*XC(14)-78.D0
A(5,5) = -12.D0*XC(15)-60.D0
RETURN
END

```

FORTRAN IV G LEVEL 21 GRADFI DATE = 73114 03/24/07 PAGE 0001

```

0001 SUBROUTINE GRADFI
0002 DIMENSION B(100)
0003 1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)
0004 DOUBLE PRECISION A,ALFA,ALC,ALA,ALR,TRA,C,DELTFI,DELTFADTP,DELTA
      IX,D1,D,EPSIL,EPSILO,EPSIL2,ETA,EPFIL,EPFIL3,EPFIL4,EPFIL5,EPFIL7,
      2EPSILU,EPFIL6,EPFIL8,EPFIL9,EPFIL10,EPFIL11,EPFIL12,EPFIL13,EPFIL14,EPFIL15,EPFIL16,EPFIL17,
      3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,PSIT,PSI4,PENAL,PE1,PE2,PE3,
      4RAPH,ROB,RO,RA,SCAL,TR,TC,TPEN,TRE,TR,TRI,TETA,TETAT,TOTG,TQ
      5IG1,TETAL,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TQ2,TET1,T1
0005 DOUBLE PRECISION TRB,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VC1,VCL,XI,
      1XNV,XNO,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSRI,XSB2,XMK,XIEL,XIE2
      2,XSE1,XSE2,XTRI,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
      3,Z,Z1,X,XC,XS,X1,X2,XP,B
0006 DOUBLE PRECISION DIT,DISOLS
0007 DOUBLE PRECISION DMOD,DSORT,DARS,DMIN1,DMAX1
0008 COMMON B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
0009 COMMON FII,PHI,PSI,PSI3,TR,TD,TC,EPFIL,EPFIL2,YSORT
0010 COMMON NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEKA 50
      1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDR,KCEGA 60
      2DB,KFIL,KLIN,KRFN,KD,IBAS,IHB,IVC,IVA,IVB
0011 COMMON KFOC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEKA 80
      1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREF,JKO,LC
      IF (IT.GE.0) GO TO 100
0012 DO 1J 1=6,15
0013 C(1) = B(1-5)
0014 10 CONTINUE
0015 100 CONTINUE
0016 C(1) = -24.D0*XC(1)**2-60.D0*XC(1)+40.D0*XC(2)+20.D0*XC(3)-64.D0*XC(4)+20.D0*XC(5)
0017 C(2) = -48.D0*XC(2)**2+40.D0*XC(1)-78.D0*XC(2)+12.D0*XC(3)+62.D0*XC(4)+20.D0*XC(5)
0018 C(3) = -60.D0*XC(3)**2+20.D0*XC(1)+12.D0*XC(2)-20.D0*XC(3)+12.D0*XC(4)+20.D0*XC(5)
0019 C(4) = -36.D0*XC(4)**2-64.D0*XC(1)+62.D0*XC(2)+12.D0*XC(3)-78.D0*XC(4)+20.D0*XC(5)
0020 C(5) = -12.D0*XC(5)**2+20.D0*XC(1)-64.D0*XC(2)+20.D0*XC(3)+40.D0*XC(4)-60.D0*XC(5)
0021 RETURN
0022 END
0023
      GEGN 10
      GEGN 20
      GEGN 30
      GEGN 130
      GEGN 140
      GEGN 150
      GEGN 160
      GEGN 170
      GEGN 180
      GEGN 190
      GEGN 200
      GEGN 210
      GEGN 220
      GEGN 230
      GEGN 240
      GEGN 250
      GEGN 260
      GEGN 270
      GEGN 280
      GEGN 290

```

PARAMETERS

NV	17
NTN	3
NEG	2
NEVL	20
NTD	6
ITFT	20
ICONJ	1
IQIAG	0
ITMAX	50
KFIL	0
KLIN	0
NCO	10
ITSOR	1
ISCLSOR	50
EPSIL	0.10-00
EPSILU	0.10-03
EPSILI	0.10-03
EPSIL2	0.10-03
PC	0.0

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 17
 TOTAL NUMBER OF VARIABLES 23
 NUMBER OF CONSTRAINTS 5
 EPSILON FOR NEWTON'S METHOD 0.1000D-03
 EPSILON FOR GRADIENT TEST 0.1000D-03

OBJECTIVE FUNCTION -0.8000000000000000 01

	LOWER BOUND	STARTING POINT	UPPER BOUND
X(1)	0.0	0.0	0.1000000000000000 03
X(2)	0.0	0.0	0.1000000000000000 03
X(3)	0.0	0.0	0.1000000000000000 03
X(4)	0.0	0.0	0.1000000000000000 03
X(5)	0.0	0.1000000000000000 01	0.1000000000000000 03
X(6)	0.0	0.0	0.1000000000000000 03
X(7)	0.0	0.0	0.1000000000000000 03
X(8)	0.0	0.0	0.1000000000000000 03
X(9)	0.0	0.0	0.1000000000000000 03
X(10)	0.0	0.0	0.1000000000000000 03
X(11)	0.0	0.0	0.1000000000000000 03
X(12)	0.0	0.0	0.1000000000000000 03
X(13)	0.0	0.0	0.1000000000000000 03
X(14)	0.0	0.5000000000000000 01	0.1000000000000000 03
X(15)	0.0	0.1000000000000000 01	0.1000000000000000 03
X(16)	0.0	0.0	0.1000000000000000 03
X(17)	0.0	0.2800000000000000 02	0.1000000000000000 03

VALUE OF CONSTRAINTS

C(1) 0.4100000000000000 02
 C(2) -0.2600000000000000 02
 C(3) 0.7200000000000000 02
 C(4) 0.7900000000000000 02
 C(5) 0.0

IT 1	PHI-0.693820426925710	10	DIR. GRAD.	NO 6	YN 0.78D 10	DELTA FI 0.11D 13	ETA 0.54D 12	NCOR 0	NCN 2	NITN 2
IT 2	PHI-0.582604536900750	10	DIR. CONJ.	LC 12				NCOR 0	NCN 2	NITN 2
IT 3	PHI-0.582374357588210	10	DIR. CONJ.	LC 12				NCOR 0	NCN 2	NITN 1
IT 4	PHI-0.515332434191710	10	ITERATION	SPECIAL				NCOR 1	NCN 1	NITN 1
IT 5	PHI-0.487327138252220	10	DIP. GRAD.	NO 6	RASF DEGENEREE	DELTA FI 0.74D 12	ETA 0.29D 12	NCOR 1	NCN 3	NITN 7
IT 6	PHI-0.235819115495400	10	DIR. GRAD.	NO 5	YN 0.19D 10	DELTA FI 0.74D 12	ETA 0.24D 18	NCOR 1	NCN 5	NITN 18
IT 7	PHI-0.173961078582620	10	DIR. GRAD.	NO 5	YN 0.14D 10	DELTA FI 0.66D 12	ETA 0.29D 12	NCOR 1	NCN 2	NITN 10
IT 8	PHI-0.107977191398700	10	DIP. GRAD.	NO 5	YN 0.40D 10	DELTA FI 0.66D 12	ETA 0.37D 12	NCOR 2	NCN 5	NITN 68
IT 9	PHI-0.115910635002670	03	CIP. GRAD.	NO 17	YN 0.42D 08	DELTA FI 0.11D 10	ETA 0.11D 10	NCOR 0	NCN 5	NITN 57
LES VARIABLES ARTIFICIELLES SONT TOUTES ANNULEES										
IT 10	PHI-0.315141862909520	02	DIP. GRAD.	NO 8	YN 0.92D 02	DELTA FI 0.12D 05	ETA 0.56D 04	NCOR 1	NCN 19	NITN 16
IT 11	PHI-0.624503693487200	02	DIR. GRAD.	NO 11	YN 0.72D 02	DELTA FI 0.72D 04	ETA 0.49D 04	NCOR 1	NCN 4	NITN 19
IT 12	PHI-0.48145406261518D	02	DIP. GRAD.	NO 13	YN 0.93D 01	DELTA FI 0.97D 03	ETA 0.60D 03	NCOR 1	NCN 5	NITN 40
IT 13	PHI-0.344894826432100	02	DIP. GRAD.	NO 15	YN 0.41D 01	DELTA FI 0.25D 03	ETA 0.18D 03	NCOR 0	NCN 3	NITN 25
IT 14	PHI-0.39581432670930	02	DIR. GRAD.	NO 16	YN 0.12D 02	DELTA FI 0.19D 03	ETA 0.18D 03	NCOR 0	NCN 2	NITN 5
IT 15	PHI-0.33496133261020	02	DIR. GRAD.	NO 16	YN 0.61D 01	DELTA FI 0.17D 03	ETA 0.17D 03	NCOR 0	NCN 2	NITN 2
IT 16	PHI-0.329490164453200	02	CIP. GRAD.	NO 16	YN 0.59D 01	DELTA FI 0.17D 03	ETA 0.17D 03	NCOR 0	NCN 2	NITN 0
IT 17	PHI-0.339489195441640	02	CIP. GRAD.	NO 16	YN 0.58D 01	DELTA FI 0.17D 03	ETA 0.17D 03	NCOR 0	NCN 2	NITN 0
IT 18	PHI-0.339428897452810	02	DIR. GRAD.	NO 17	YN 0.51D 02	DELTA FI 0.17D 03	ETA 0.33D 10	NCOR 1	NCN 2	NITN 2
IT 19	PHI-0.3393348860782750	02	DIP. GRAD.	NO 13	YN 0.14D 02	DELTA FI 0.17D 03	ETA 0.24D 03	NCOR 1	NCN 2	NITN 2
IT 20	PHI-0.418730872845070	02	DIR. GRAD.	NO 16	YN 0.40D 01	DELTA FI 0.17D 03	ETA 0.17D 03	NCOR 1	NCN 1	NITN 7
IT 21	PHI-0.40846620643330	02	DIP. GRAD.	NO 15	YN 0.33D 01	DELTA FI 0.17D 03	ETA 0.11D 03	NCOR 0	NCN 1	NITN 4
IT 22	PHI-0.4384389311104640	02	DIP. GRAD.	NO 15	YN 0.36D 01	DELTA FI 0.12D 03	ETA 0.11D 03	NCOR 0	NCN 2	NITN 2
IT 23	PHI-0.40841189132428D	02	DIP. GRAD.	NO 15	YN 0.36D 01	DELTA FI 0.12D 03	ETA 0.11D 03	NCOR 0	NCN 2	NITN 2
IT 24	PHI-0.409364482221490	02	DIP. GRAD.	NO 15	YN 0.34D 01	DELTA FI 0.12D 03	ETA 0.11D 03	NCOR 0	NCN 2	NITN 2
IT 25	PHI-0.409139193133390	02	DIP. GRAD.	NO 15	YN 0.33D 01	DELTA FI 0.12D 03	ETA 0.11D 03	NCOR 0	NCN 2	NITN 2
IT 26	PHI-0.402087150293290	02	DIP. GRAD.	NO 15	YN 0.31D 01	DELTA FI 0.12D 03	ETA 0.11D 03	NCOR 0	NCN 1	NITN 4
IT 27	PHI-0.397160068337270	02	DIR. GRAD.	NO 15	YN 0.31D 01	DELTA FI 0.12D 03	ETA 0.12D 03	NCOR 0	NCN 1	NITN 4
IT 28	PHI-0.396658446873770	02	DIP. GRAD.	NO 15	YN 0.30D 01	DELTA FI 0.12D 03	ETA 0.88D 02	NCOR 0	NCN 2	NITN 4
IT 29	PHI-0.387283414581670	02	DIP. GRAD.	NO 15	YN 0.28D 01	DELTA FI 0.12D 03	ETA 0.83D 02	NCOR 0	NCN 1	NITN 3
IT 30	PHI-0.386448514630090	02	DIP. GRAD.	NO 15	YN 0.52D 01	DELTA FI 0.12D 03	ETA 0.78D 02	NCOR 0	NCN 2	NITN 4
IT 31	PHI-0.386323486591920	02	DIR. GRAD.	NO 15	YN 0.38D 01	DELTA FI 0.11D 03	ETA 0.74D 02	NCOR 0	NCN 2	NITN 2
IT 32	PHI-0.38623536312998D	02	DIP. GRAD.	NO 15	YN 0.35D 01	DELTA FI 0.11D 03	ETA 0.73D 02	NCOR 0	NCN 2	NITN 2
IT 33	PHI-0.3861805000713D	02	DIR. GRAD.	NO 15	YN 0.33D 01	DELTA FI 0.11D 03	ETA 0.72D 02	NCOR 0	NCN 2	NITN 2
IT 34	PHI-0.386135482820070	02	DIP. GRAD.	NO 15	YN 0.31D 01	DELTA FI 0.11D 03	ETA 0.71D 02	NCOR 0	NCN 2	NITN 2
IT 35	PHI-0.386084917022710	02	DIR. GRAD.	NO 15	YN 0.30D 01	DELTA FI 0.11D 03	ETA 0.70D 02	NCOR 0	NCN 2	NITN 2
IT 36	PHI-0.38599697112050	02	DIR. GRAD.	NO 15	YN 0.29D 01	DELTA FI 0.11D 03	ETA 0.69D 02	NCOR 0	NCN 2	NITN 2
IT 37	PHI-0.385445275158370	02	DIP. GRAD.	NO 15	YN 0.28D 01	DELTA FI 0.11D 03	ETA 0.68D 02	NCOR 0	NCN 1	NITN 2
IT 38	PHI-0.33986672296784D	02	DIR. CONJ.	LC 3				NCOR 0	NCN 4	NITN 31
IT 39	PHI-0.33967062199493D	02	DIR. GRAD.	NO 14	YN 0.37D 01	DELTA FI 0.11D 03	ETA 0.19D 03	NCOR 0	NCN 2	NITN 3
IT 40	PHI-0.33963270576766D	02	DIR. GRAD.	NO 14	YN 0.33D 01	DELTA FI 0.11D 03	ETA 0.15D 03	NCOR 0	NCN 2	NITN 2
IT 41	PHI-0.33962982089110D	02	DIR. GRAD.	NO 14	YN 0.32D 01	DELTA FI 0.11D 03	ETA 0.14D 03	NCOR 0	NCN 2	NITN 1
IT 42	PHI-0.33962149222362D	02	DIR. GRAD.	NO 13	YN 0.72D 01	DELTA FI 0.11D 03	ETA 0.11D 03	NCOR 1	NCN 3	NITN 2
	PHI-0.33962149222362D	02			YN 0.30D 01	DELTA FI 0.11D 03	ETA 0.13D 03			

EXECUTION TIME 5854 CENTISECONDES

ASSOCIATED REDUCED GRADIENT VALUE

PRIMAL VARIABLE

X(1) =	0.500098467702240 00	V(1) =	0.
X(2) =	0.305102909730340 00	V(2) =	0.
X(3) =	0.322196784226880 00	V(3) =	0.
X(4) =	0.217633086756550 00	V(4) =	0.
X(5) =	0.142163660325480 00	V(5) =	0.
X(6) =	0.0	V(6) =	-0.328262634229810 02
X(7) =	0.0	V(7) =	-0.176117473589290 01
X(8) =	0.366028192239330 01	V(8) =	0.855951068504080 00
X(9) =	0.746722347451920-05	V(9) =	-0.237709817318760 01
X(10) =	0.325426650947420 01	V(10) =	-0.429255081818430 00
X(11) =	0.116137452756930 02	V(11) =	-0.711409798496970 00
X(12) =	0.0	V(12) =	-0.385128050912590 02
X(13) =	0.0	V(13) =	-0.573456755263180 02
X(14) =	0.276781631083630-01	V(14) =	0.134175471150280 01
X(15) =	0.0	V(15) =	-0.487194908741480 00
X(16) =	0.0	V(16) =	-0.217633086756550 00
X(17) =	0.0	V(17) =	-0.142163660325480 00

CONSTRAINT

LAGRANGE MULTIPLIER

CONTRAINTE 1	-0.100076951962790-03	U(1) =	0.500098467702240 00
CONTRAINTE 2	-0.113768192974820-03	U(2) =	0.305102909730340 00
CONTRAINTE 3	-0.999867346500590-04	U(3) =	0.322196784226880 00
CONTRAINTE 4	-0.680271472486280-07	U(4) =	0.217633086756550 00
CONTRAINTE 5	-0.324953539720950-05	U(5) =	0.142163660325480 00

STOPPING CONDITION	8	
NUMBER OF ITERATIONS	42	
NUMBER OF TIMES THE CONSTRAINTS WERE CALCULATED	475	
NUMBER OF TIMES THE OBJECTIVE FUNCTION WAS CALCULATED	474	
NUMBER OF TIMES THE GRADIENT WAS CALCULATED	95	
NUMBER OF INVERSE CALCULATIONS WITH THE GIVEN BASE	43	
NUMBER OF INVERSE CALCULATIONS WITH ANOTHER BASE	3	
NUMBER OF BASE CHANGES	13	
NUMBER OF TIMES THE JACOBIEN OF THE CONSTRAINTS WAS CALCULATED	47	
NUMBER OF MAXIMUMS FOUND WITHOUT BISECTION	37	
NUMBER OF MAXIMUMS FOUND BY BISECTION	10	
NUMBER OF REENTRIES IN TO THE FEASIBLE REGION	113	
TOTAL NUMBER OF NEWTON-S ITERATIONS	368	

Example 5 of Chapter 3

Problem:

minimize

$$f_0(\bar{X}) = 4x_1 + 10x_2 + 4x_3 + 2(x_1^2 + x_2^2)^{1/2}$$

subject to

$$f_1(\bar{X}) = x_1 x_2 x_3 \geq 100$$

$$x_i \geq 0, \quad i = 1, 2, 3$$

Computer program and the results of this problem is presented in the following pages (p131 through p139).

ILLEGIBLE DOCUMENT

**THE FOLLOWING
DOCUMENT(S) IS OF
POOR LEGIBILITY IN
THE ORIGINAL**

**THIS IS THE BEST
COPY AVAILABLE**

```

FORTRAN IV G LEVEL 31      PHIX      DATE = 73667      01/30/16      PAGE 0707
      SUBROUTINE PHIX
      DIMENSION B(100)
      DIMENSION A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
      1,Y(150),C(150),VC(150),IBAS(50),IUB(100),IVC(50),IVA(100)
      DOUBLE PRECISION A,ALFA,ALC,ALA,ALP,TPAC,C,DELIFT,DELIFT2,OTP,OTFLTA
      1X,OL,OT,PSI1,PSI12,PSI13,PSI14,PSI15,PSI16,PSI17,PSI18,PSI19,
      2PSI20,PSI21,PSI22,PSI23,PSI24,PSI25,PSI26,PSI27,PSI28,PSI29,
      3PSI30,PSI31,PSI32,PSI33,PSI34,PSI35,PSI36,PSI37,PSI38,PSI39,
      4PSI40,PSI41,PSI42,PSI43,PSI44,PSI45,PSI46,PSI47,PSI48,PSI49,
      5PSI50,PSI51,PSI52,PSI53,PSI54,PSI55,PSI56,PSI57,PSI58,PSI59,
      6PSI60,PSI61,PSI62,PSI63,PSI64,PSI65,PSI66,PSI67,PSI68,PSI69,
      7PSI70,PSI71,PSI72,PSI73,PSI74,PSI75,PSI76,PSI77,PSI78,PSI79,
      8PSI80,PSI81,PSI82,PSI83,PSI84,PSI85,PSI86,PSI87,PSI88,PSI89,
      9PSI90,PSI91,PSI92,PSI93,PSI94,PSI95,PSI96,PSI97,PSI98,PSI99,
      10PSI100,PSI101,PSI102,PSI103,PSI104,PSI105,PSI106,PSI107,PSI108,PSI109,
      11PSI110,PSI111,PSI112,PSI113,PSI114,PSI115,PSI116,PSI117,PSI118,PSI119,
      12PSI120,PSI121,PSI122,PSI123,PSI124,PSI125,PSI126,PSI127,PSI128,PSI129,
      13PSI130,PSI131,PSI132,PSI133,PSI134,PSI135,PSI136,PSI137,PSI138,PSI139,
      14PSI140,PSI141,PSI142,PSI143,PSI144,PSI145,PSI146,PSI147,PSI148,PSI149,
      15PSI150,PSI151,PSI152,PSI153,PSI154,PSI155,PSI156,PSI157,PSI158,PSI159,
      16PSI160,PSI161,PSI162,PSI163,PSI164,PSI165,PSI166,PSI167,PSI168,PSI169,
      17PSI170,PSI171,PSI172,PSI173,PSI174,PSI175,PSI176,PSI177,PSI178,PSI179,
      18PSI180,PSI181,PSI182,PSI183,PSI184,PSI185,PSI186,PSI187,PSI188,PSI189,
      19PSI190,PSI191,PSI192,PSI193,PSI194,PSI195,PSI196,PSI197,PSI198,PSI199,
      20PSI200,PSI201,PSI202,PSI203,PSI204,PSI205,PSI206,PSI207,PSI208,PSI209,
      21PSI210,PSI211,PSI212,PSI213,PSI214,PSI215,PSI216,PSI217,PSI218,PSI219,
      22PSI220,PSI221,PSI222,PSI223,PSI224,PSI225,PSI226,PSI227,PSI228,PSI229,
      23PSI230,PSI231,PSI232,PSI233,PSI234,PSI235,PSI236,PSI237,PSI238,PSI239,
      24PSI240,PSI241,PSI242,PSI243,PSI244,PSI245,PSI246,PSI247,PSI248,PSI249,
      25PSI250,PSI251,PSI252,PSI253,PSI254,PSI255,PSI256,PSI257,PSI258,PSI259,
      26PSI260,PSI261,PSI262,PSI263,PSI264,PSI265,PSI266,PSI267,PSI268,PSI269,
      27PSI270,PSI271,PSI272,PSI273,PSI274,PSI275,PSI276,PSI277,PSI278,PSI279,
      28PSI280,PSI281,PSI282,PSI283,PSI284,PSI285,PSI286,PSI287,PSI288,PSI289,
      29PSI290,PSI291,PSI292,PSI293,PSI294,PSI295,PSI296,PSI297,PSI298,PSI299,
      30PSI300,PSI301,PSI302,PSI303,PSI304,PSI305,PSI306,PSI307,PSI308,PSI309,
      31PSI310,PSI311,PSI312,PSI313,PSI314,PSI315,PSI316,PSI317,PSI318,PSI319,
      32PSI320,PSI321,PSI322,PSI323,PSI324,PSI325,PSI326,PSI327,PSI328,PSI329,
      33PSI330,PSI331,PSI332,PSI333,PSI334,PSI335,PSI336,PSI337,PSI338,PSI339,
      34PSI340,PSI341,PSI342,PSI343,PSI344,PSI345,PSI346,PSI347,PSI348,PSI349,
      35PSI350,PSI351,PSI352,PSI353,PSI354,PSI355,PSI356,PSI357,PSI358,PSI359,
      36PSI360,PSI361,PSI362,PSI363,PSI364,PSI365,PSI366,PSI367,PSI368,PSI369,
      37PSI370,PSI371,PSI372,PSI373,PSI374,PSI375,PSI376,PSI377,PSI378,PSI379,
      38PSI380,PSI381,PSI382,PSI383,PSI384,PSI385,PSI386,PSI387,PSI388,PSI389,
      39PSI390,PSI391,PSI392,PSI393,PSI394,PSI395,PSI396,PSI397,PSI398,PSI399,
      40PSI400,PSI401,PSI402,PSI403,PSI404,PSI405,PSI406,PSI407,PSI408,PSI409,
      41PSI410,PSI411,PSI412,PSI413,PSI414,PSI415,PSI416,PSI417,PSI418,PSI419,
      42PSI420,PSI421,PSI422,PSI423,PSI424,PSI425,PSI426,PSI427,PSI428,PSI429,
      43PSI430,PSI431,PSI432,PSI433,PSI434,PSI435,PSI436,PSI437,PSI438,PSI439,
      44PSI440,PSI441,PSI442,PSI443,PSI444,PSI445,PSI446,PSI447,PSI448,PSI449,
      45PSI450,PSI451,PSI452,PSI453,PSI454,PSI455,PSI456,PSI457,PSI458,PSI459,
      46PSI460,PSI461,PSI462,PSI463,PSI464,PSI465,PSI466,PSI467,PSI468,PSI469,
      47PSI470,PSI471,PSI472,PSI473,PSI474,PSI475,PSI476,PSI477,PSI478,PSI479,
      48PSI480,PSI481,PSI482,PSI483,PSI484,PSI485,PSI486,PSI487,PSI488,PSI489,
      49PSI490,PSI491,PSI492,PSI493,PSI494,PSI495,PSI496,PSI497,PSI498,PSI499,
      50PSI500,PSI501,PSI502,PSI503,PSI504,PSI505,PSI506,PSI507,PSI508,PSI509,
      51PSI510,PSI511,PSI512,PSI513,PSI514,PSI515,PSI516,PSI517,PSI518,PSI519,
      52PSI520,PSI521,PSI522,PSI523,PSI524,PSI525,PSI526,PSI527,PSI528,PSI529,
      53PSI530,PSI531,PSI532,PSI533,PSI534,PSI535,PSI536,PSI537,PSI538,PSI539,
      54PSI540,PSI541,PSI542,PSI543,PSI544,PSI545,PSI546,PSI547,PSI548,PSI549,
      55PSI550,PSI551,PSI552,PSI553,PSI554,PSI555,PSI556,PSI557,PSI558,PSI559,
      56PSI560,PSI561,PSI562,PSI563,PSI564,PSI565,PSI566,PSI567,PSI568,PSI569,
      57PSI570,PSI571,PSI572,PSI573,PSI574,PSI575,PSI576,PSI577,PSI578,PSI579,
      58PSI580,PSI581,PSI582,PSI583,PSI584,PSI585,PSI586,PSI587,PSI588,PSI589,
      59PSI590,PSI591,PSI592,PSI593,PSI594,PSI595,PSI596,PSI597,PSI598,PSI599,
      60PSI600,PSI601,PSI602,PSI603,PSI604,PSI605,PSI606,PSI607,PSI608,PSI609,
      61PSI610,PSI611,PSI612,PSI613,PSI614,PSI615,PSI616,PSI617,PSI618,PSI619,
      62PSI620,PSI621,PSI622,PSI623,PSI624,PSI625,PSI62
```


PARAMETERS

NV	3
NTN	1
NEG	0
NEVL	20
NFC	6
ITET	20
ICONJ	1
ITAG	0
ITMAX	50
KEL	0
KLIN	0
NCO	10
ITSOR	1
ISOLSOR	50
EPSIL	0.10 00
EPSILO	0.10-04
EPSILI	0.10-02
EPSILZ	0.10-02
PC	0.10 01

GENERALIZED REDUCED GRADIENT

NUMBER OF PRIMAL VARIABLES 3
 TOTAL NUMBER OF VARIABLES 4
 NUMBER OF CONSTRAINTS 1
 EPSILON FOR NEWTON'S METHOD 0.10000-04
 EPSILON FOR GRADIENT TEST 0.10000-02

OBJECTIVE FUNCTION -0.209284271247460 03

	LOWER BOUND	STARTING POINT	UPPER BOUND
X(1)	0.0	0.100000000000000 02	0.100000000000000 03
X(2)	0.0	0.100000000000000 02	0.100000000000000 03
X(3)	0.0	0.100000000000000 02	0.100000000000000 03

VALUE OF CONSTRAINTS

C(1) -0.900000000000000 03

IT 1	PHI-0.132777690214430 03	DIR.GRAD.	NO 0	YN 0.130 02	DELTAFI 0.210 03	ETA 0.110 03	NCOR 0	NCN 2	LITN 2
IT 2	PHI-0.983778272470720 02	DIP.CONJ.	LC 3				NCOR 0	NCN 2	NITN 2
IT 3	PHI-0.893485427300490 02	DIP.CONJ.	LC 3				NCOR 0	NCN 2	NITN 2
IT 4	PHI-0.803136972316050 02	DIP.CONJ.	LC 3				NCOR 1	NCN 3	NITN 10
IT 5	PHI-0.887971893324040 02	DIP.GRAD.	NO 1	YN 0.280 01	DELTAFI 0.210 03	ETA 0.240 03	NCOR 0	NCN 3	NITN 22
IT 6	PHI-0.879957808645300 02	DIP.CONJ.	LC 2				NCOR 0	NCN 5	NITN 47
IT 7	PHI-0.879903545808250 02	DIP.CONJ.	LC 2				NCOR 0	NCN 3	NITN 28
IT 8	PHI-0.879877915379920 02	DIP.GRAD.	NO 0	YN 0.290 00	DELTAFI 0.390 00	ETA 0.240 00	NCOR 0	NCN 7	NITN 9
IT 9	PHI-0.879877697106300 02	DIP.GRAD.	NO 1	YN 0.260-01	DELTAFI 0.390 00	ETA 0.200 01	NCOR 0	NCN 3	NITN 9
IT 10	PHI-0.879877697106300 02	DIP.GRAD.	NO 1	YN 0.360-02	DELTAFI 0.190-01	ETA 0.180-01	NCOR 0	NCN 3	NITN 5

EXECUTION TIME 227 CENTISECONDES

ASSOCIATED REDUCED GRADIENT VALUE

$V(1) = -0.355924839571880-02$
 $V(2) = 0.$
 $V(3) = -0.147183280286180-03$

LAGRANGE MULTIPLIER

$U(1) = 0.2932228599558380 00$

PRIMAL VARIABLE

$X(1) = 0.508590067973640 01$
 $X(2) = 0.268206805044030 01$
 $X(3) = 0.733098547692200 01$

CONSTRAINT

CONTRAINTE 1 $-0.100718688074150-04$

STOPPING CONDITION	1		
NUMBER OF ITERATIONS	9		
NUMBER OF TIMES THE CONSTRAINTS WERE CALCULATED	160		
NUMBER OF TIMES THE OBJECTIVE FUNCTION WAS CALCULATED	94		
NUMBER OF TIMES THE GRADIENT WAS CALCULATED	19		
NUMBER OF INVERSE CALCULATIONS WITH THE GIVEN BASE	7		
NUMBER OF INVERSE CALCULATIONS WITH ANOTHER BASE	1		
NUMBER OF BASE CHANGES	1		
NUMBER OF TIMES THE JACOBIEN OF THE CONSTRAINTS WAS CALCULATED	12		
NUMBER OF MAXIMUMS FOUND WITHOUT BISECTION	8		
NUMBER OF MAXIMUMS FOUND BY BISECTION	2		
NUMBER OF REENTRIES IN TO THE FEASIBLE REGION	30		
TOTAL NUMBER OF NEWTON'S ITERATIONS	138		

A STUDY OF THE GENERALIZED
REDUCED GRADIENT METHOD

by

YOUSEF SHAFII

B.S., TEHRAN POLYTECHNIC , TEHRAN , IRAN , 1961

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the
requirement for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1973

This report reviews the mathematical theory of generalized reduced gradient method (GRG). Its computer program via GREG code is described and a summary of GREG algorithm along with a numerical example is presented. The double precision version of GREG code is introduced and the procedure of using this program is given. By a series of numerical examples GREG double precision code is compared with the GREG single precision code.