USER'S GUIDE TO SUBROUTINE 'EVOP'

FORTRAN –IV Version

Sayeed N Ghani PhD (Univ London), DIC (Imperial College), CEng (UK), MIEE (UK), Quality Six Sigma Green Belt Certified (USA)

Contents

APPENDIX - A
FORTRAN – IV Code

USER'S GUIDE TO SUBROUTINE 'EVOP' --- Fortran-IV Version

ABSTRACT

This is the user's manual for a new optimization algorithm EVOP coded in FORTRAN-IV language. Few current methods cope with real world problems involving discontinuous objective and constraining functions where there is a mix of continuous, discrete and integer arguments and global minimum is sought. For noisy data, solutions are possible with genetic algorithms but costly parallel processing would be needed to locate the global minimum. Solutions remain illusive with genetic algorithms for problems with hard real-time constraints. The author's robust algorithm EVOP surmounts these difficulties with a much faster and more accurate solution. No gradient information is needed, and there is a high probability of locating the global minimum with a small number of automatic restarts as specified by the user. It appears EVOP is the 'silver bullet' that has succeeded in slaying the 'dragon' of dimensionality in multiple minima bound objective functions.

Audience: **Virtually from any applied discipline. Optimization is addressed in all spheres of human enterprise from natural sciences and engineering of whatever discipline, through economics, econometrics, statistics and operations research to management science. Students and practitioners of mathematical programming who require global optimization methods in such diverse technological application areas as, nucleonic, mechanical, civil, structural, electrical, electronic, chemical engineering, high performance control systems, fuzzy-logic systems, metallurgy, space technology, chip design, networks and transportation, databases, image processing, molecular biology, environmental engineering, finance and other such quantitative topics will immensely benefit from the knowledge of using global optimizer EVOP. Further optimization is one of the core concepts used in DFSS (Design for Six Sigma) for product to service and for manufacturing to transaction.**

1 Introduction

This is the user's manual for a numerical optimization software EVOP (Evolutionary Operation) written in FORTRAN-IV language. The software minimizes the following parameter optimization problem [1].

Minimize an objective function $F_o(x) = F_o(x_1, x_2, \ldots\ldots. x_n)$ where $F_o(x)$ is a function, of arbitrary complexity, of n independent variables $(x_1, x_2, \ldots\ldots. x_n)$.

The n independent parameters or design variables x_i 's (i = 1 …… n) are subject to explicit constraints

$$l_i <= x_i <= g_i$$

where l_i 's and g_i 's are lower and upper bounds on the n independent parameters. They can be constants or functions, of arbitrary complexity, of n independent variables $(x_1, x_2, \ldots\ldots. ,x_n)$. In the latter case the explicit constraints constitute moving boundaries.

These n independent parameters x_i 's are also subject to m numbers of implicit constraints

$$L_j <= f_j(x) <= G_j \quad \text{where } j = 1, 2 \ldots. m$$

L_j 's and G_j 's are lower and upper bounds on the m implicit constraints. They may be constants or functions, of arbitrary complexity, of n independent variables $(x_1, x_2, \ldots\ldots. x_n)$. Functions $f_i(x)$ are also of arbitrary complexity of n independent variables $(x_1, x_2, \ldots\ldots. , x_n)$.

Additionally, the optimization can also be subject to q numbers of equality constraints

$$h_k(x) = 0 \quad \text{where } k = 1, 2 \ldots.. q.$$

Functions $h_k(x)$ are also of arbitrary complexity of n independent variables $(x_1, x_2, \ldots\ldots. , x_n)$.

Selection of an optimum parameter set is of fundamental importance. It is addressed in all spheres of human enterprise from engineering and natural sciences through economics and econometrics to operational research and management science. Further, optimization is one of the core concepts used in DFSS (Design for Six Sigma) for product to service and for manufacturing to transaction. This is to name but a few of the application areas for this software.

2 Realism

A truly versatile minimization program for realistic problems should possess at the very least, the following [2]:

(i) The capability to deal with a possible finite number of discontinuities in nonlinear objective and constraining functions.
(ii) The ability to minimize directly an objective function without requiring information on gradients or subgradients.
(iii) The ability to deal with objective functions with a mix of continuous, discrete and integer variables as arguments.
(iv) No requirement for the scaling of objective and constraining functions.
(v) The capability to locate directly with high probability the global minimum.
(vi) The capability for optimization even when there are more than one of the above difficulties present simultaneously.
(vii) The facility for an automatic restart to check that the previously obtained value is the global minimum.

Further requirements on such an algorithm for optimum and adaptive control of physical systems in real or accelerated time are:

(viii) The objective function should never be evaluated in the infeasible region (as a consequence the safety of the plant will never be in jeopardy because of optimization).
(ix) Gradients or subgradients should never be used (this will ensure that the noise in the measurement will not be accentuated to adversely effect the optimization process).
(x) An inherent ability to cope with the realistic hard time constraint requirement imposed by real-time.

3 Global Minimum

Global optimization, so far, has been a rather difficult and illusive problem. It has not yet reached its full maturity, and consequently the literature is not so rich compared to that for local optimization [3, pages 1] and [4]. Results from interactive global optimization methods are as yet very prefatory, and require visualization of data and reduction of the dimensionality of the parameter space. Algorithms developed so far succeed in giving a rather rough estimate of the global minimum [3, p.171]. Algorithms based on random search techniques have been developed by Palosaari, Parviainen et.al. [3, p.171], and applied to models for optimization containing both inequality and equality constraints. However, some work [4] has been done to formalize the subject on a strict mathematical basis. In this reference development and analysis of algorithms are presented using familiar geometric concepts for ease in visualization and comprehension. The book is ideally suited for beginners to obtain a sound foothold on the subject and underlying concepts as well as to learn the mathematical language. But no algorithm has been presented or cited that will efficiently solve realistic global optimization problems encountered in real-life situation [such as ‘mm7_v1’ or ‘mm7m_v1’ cited herein].

Floudas and Gounaris [5] present an excellent survey paper that covers rapid developments during the first decade of 21st century. They cite methods that are, however, restricted to dealing with binary, integer and continuous variables only. But they do not discuss any algorithm which will efficiently solve realistic global optimization problems encountered in real-life situation that contain, in addition, discrete variables as well. It appears EVOP is the ‘silver bullet’ that has succeeded in slaying the ‘dragon’ of dimensionality in multiple minima objective functions *as well as* in dealing with mixed integer, discrete and continuous variables that plague multiple minima objective functions. This ability to treat mixed integer, discrete and continuous variables is not covered in current literature, and is claimed to be a distinctly unique feature of EVOP. Researchers so far have claimed success in locating global minimum of objective functions limited to integer and continuous variables only [5]. With a user specified number of automatic restarts, the EVOP subroutine at best locates progressively lower local minima, or at worst relocates the lowest local minimum obtained so far. It is intrinsically unable to climb up to a higher local minimum. Once the global minimum has been reached any further restarts will relocate that global minimum repeatedly. A commercial website [6] is also now available.

Solutions remain elusive with genetic algorithms [7, 8] as well. They are inefficient and significantly slow. Therefore, for problems with hard real-time constraints they are practically useless.

4. The Subroutine EVOP

The algorithm is fully described in a companion report [1]. However, some further work [2] has been accomplished since the publication of the report that remains yet to be presented in relevant journals.

All variables are in double precision.

4.1 Subroutine Call Statement

To call the subroutine use the following statement.

```
      CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
```

All variables are described below.

4.2 User Written Subroutines

Three user written subroutines must be provided. They are as follows.

4.2.1 Function Value

```
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(1)
```

All variables are described below.

Given the coordinates of a point in an n-dimensional space the subroutine calculates the objective function value. If the function value at the minimum is zero, add 1.0 in order to speed up convergence with specified degree of accuracy. Otherwise ‘evop’ will seek zero going through a monotonically decreasing sequence like 10^{-6} , 10^{-20}, 10^{-25} , 10^{-32} etc. Convergence will not be obtained and number of calls to ‘func’ excessive.

4.2.2 Explicit Constraints

```
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
```

All variables are described below.

Given the coordinates of a point the subroutine calculates the upper and the lower limits of the explicit constraints.

For problems with mixed variables rounding off to integer or discrete variables takes place in this subroutine. The method is as follows. Narrow feasible strips are defined around the integer and discrete variables to force these variables to take up values near to the specified integer or discrete values as appropriable. This would allow all variables to be of continuous values, and the procedure will circumvent collapse of the complex vertices to a sub-space. The width of the strip is defined by assigning a small fractional value, and then calling subroutine `DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))` (Problems mv1_v1 to mv4m_v1).

4.2.3 Implicit Constraints

```
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
```

All variables are described below.

Given the coordinates of a point the subroutine calculates the implicit constraints, and their upper and lower limits.

4.2.4 Equality Constraints

The procedure is unable to directly deal with equality constraints of the form $h_j(x) = 0$, $j = 1, 2, \ldots K$. However, they can be satisfied in two ways.

(i) If the equality constraints are soft ie a small deviation D_j from 0 is permissible, then the equality constraints can be introduced as inequalities and handled as usual.

$$-D_j <= h_j(x) >= D_j, \; j = 1, 2, \ldots K$$

(ii) Hard equality constraints can be indirectly satisfied by defining an augmented objective function

$$F_0(x, L) = F_0(x_1, x_2 \ldots\ldots x_{n;} L_1, L_2, \ldots L_{K)}$$
$$= f_0(x) + L_1 h_1(x) + L_2 h_2(x) + \ldots\ldots + L_K h_K(x)$$

where Lj ‘s are the weighting factors such that all $h_j(x)$ vanish at the minimum of $F_0(x, L)$.

5 Description of Variables

Only those variables relevant to the usage of the subroutine are discussed.

5.1 Integer quantities

5.1.1 Input

icon --- Consecutive number of times first convergence test ‘test_1’ was made. Typically 5.

imv --- Always set to 0.

ijk --- For first entry, this variable should always be set to 1. It will subsequently be changed by ‘evop’.

iprint --- Flag for printing.
= -1 or 0 for no printing.
= 1 for printing of output summary.
= 2 for printing of output summary, and both initial and final ‘complex’ configurations.

k --- Number of ‘complex’ vertices. If ‘n’ is the dimension of the parameter space then, for n <= 5 k = 2n; and for n > 5 k >= (n + 1).

knt --- Number of consecutive times the subroutine ‘func’ is called after which tests are conducted for convergence. Typically 25.

limit --- Maximum number of times the three user written subroutines ‘cons_expl’, ‘cons_impl’ and ‘func’ can be collectively called.

n --- Dimension of the parameter space.

nic --- Number of implicit constraints.

nrstrt --- Number of ‘evop’ restarts.

5.1.2 Output

ier --- Error flag.
= 1 indicates user provided starting point is violating upper limit of an explicit constraint.

= 2 indicates user provided starting point is violating lower limit of an explicit constraint.

= 3 indicates user provided starting point is violating upper limit of an implicit constraint.

= 4 indicates user provided starting point is violating the lower limit of an implicit constraint.

= 5 indicates randomly generated (k – 1) tests points not obtainable in the ‘limit’ to which the user written subroutines 'cons_expl', 'cons_impl' and 'func' can be collectively called.

= 6 indicates minimum of the objective function not obtainable within the desired accuracy of convergence. The results are those obtained after exceeding 'limit'.

= 7 indicates final 'complex' has not reduced its size to satisfy convergence test2. Results are those obtained after exceeding 'limit'.

= 8 indicates minimum of the objective function has been located to the desired degree of accuracy to satisfy both convergence tests.

iflg --- 'evop' sets 'iflg' to 1 for checking the feasibility of centroid of the 'complex'. For problems with mixed continuous , discrete and/or integer variables 'iflg' is used in the user written subroutine 'expcons' to bypass the rounding off procedure when the centroid is involved. For testing feasibility of all other trial points 'evop' sets 'iflg' to 0 to activate the rounding off procedure. In problems with all variables continuous 'iflg' becomes redundant.

kkt --- Total number of times the user written subroutine 'expcon' was called.

kount --- Total number of times the user written subroutines 'expcon', 'impcon' and 'func' were collectively called.

kut --- Number of times the user written subroutine 'func' was called during the last convergence tests.

m --- Total number of times the user written subroutine 'func' was called.

5.2 Simple floating point quantities

5.2.1 Input

alpha --- Reflection coefficient. Typical value is 1.2, and range greater than 1.0 to less than 2.0.

beta --- Contraction coefficient. Typical value is 0.5, and range greater than 0.0 to less than 1.0.

del --- Explicit constraint retention coefficient. Typical value is 10^{-12} for double precision or 10^{-6} for single precision.

f --- Function value calculated and supplied by the user written subroutine 'func' for a given feasible trial point 'xt(n)' generated by subroutine 'evop'.

gama --- Expansion coefficient. Typical value is 2.0, and range greater than 1.0 upwards.

phi --- Accuracy parameter for convergence. Typical value for calculation in double precision is 10^{-10}. 'phi = 10^{-11}' will yield lesser accuracy for convergence (lower number of most significant digits) compared to phi = 10^{-10}.

For single precision a value of 10^{-2} may be used.

phicpx --- Parameter for determining collapse of a 'complex' in a subspace. If set two or three decades higher than 'phi' (ie if 'phi = 10^{-10}' and 'phicpx = 10^{-8}'), there is every possibility that the convergence criteria would be satisfied before collapse of a 'complex' can be detected. It is usual to set 'phicpx' a decade lower than 'phi' (ie if 'phi = 10^{-10}', then set 'phicpx = 10^{-11}') to begin with, and latter adjusted for minimum number of function evaluation. Even with 'phicpx' set a decade lower than 'phi' premature satisfaction of the convergence criteria may take place when an objective function or its first derivative (razor sharp valley) contain finite number of discontinuities. The minimum so predicted may be far away from the true minimum. Tell tale evidence for this is gradual creeping of the predicted minimum on each restarts towards the true minimum. However, this should not be confused with that exhibited by objective function containing multiple minima. Such multiple minima are usually widely separated and distinct. On the other hand too low a value (high negative exponent) for 'phicpx' will inhibit convergence, and cause error return with 'ier = 5 or 6'.

5.2.2 Output

f --- On return 'f' contains the function value of the minimum located.

5.3 Floating point arrays

5.3.1 Input

xmax(n) --- Array of dimension 'n' containing the upper limits of the explicit constraints. They are calculated and supplied by the user written subroutine 'expcon' for a given trial point provided by 'evop'.

xmin(n) --- Array of dimension 'n' containing the lower limits of the explicit constraints. They are calculated and supplied by the user written subroutine ‘expcon’ for a given trial point provided by ‘evop’.

For integer variables these limits should not be integer themselves. For example if the 2nd coordinate is an integer variable, and ‘xmax(2) = 10’ set ‘xmax(2) = 10.0001’. If ‘xmin(2) = 15’ set ‘xmin(2) = 14.9999’. Similarly, for discrete variables the limits should not be any one of the predefined discrete numbers. As for example, if the 7th coordinate is a discrete variable, ‘xmin(7) = -20.3’, and ‘-20.3’ happens to be one of the discrete values that this coordinate can take, set ‘xmin(7) = -20.3001’. If ‘xmax(7) = 25.6’, and ‘25.6’ happens to be a specified discrete value, set ‘xmax(7) = 25.6001’. The specified range of variables must lie just inside the upper and the lower limits. The subroutine demands that at least one variable should be continuous. For problems with mixed integer and discrete variables only, a dummy continuous variable should be introduced. It requires to be declared only in the ‘expcon’ subroutine. Therefore, its numerical value is of no consequence to the minimization process.

xt(n) --- Array of dimension ‘n’ containing the coordinates of the trial point. On first entry ‘xt(n)’ contains the feasible trial point, and at the end of minimization it returns with the coordinates of the minimum located.

xx(nic) --- Array of dimension ‘nic’ containing the implicit constraint function values. They are calculated and supplied by the user written subroutine ‘inpcon’ for a given trial point ‘xt(n)’ provided by ‘evop’.

xxmax(nic) --- Array of dimension ‘nic’ containing the upper limit of the implicit constraints. They are calculated and supplied by the user written subroutine ‘impcon’, for a given trial point ‘xt(n)’ provided by ‘evop’.

xxmin(nic) --- Array of dimension ‘nic’ containing the lower limit of the implicit constraints. They are calculated and supplied by the user written subroutine ‘impcon’, for a given trial point ‘xt(n)’ provided by ‘evop’.

The bounds xmax(n), xmin(n), xxmax(nic), xxmin(nic) can be just constants as well.

5.3.2 Output

On return xmax(n), xmin(n), xt(n), xx(nic), xxmax(nic) and xxmin(nic) contain the corresponding values at the minimum located.

5.3.3 Floating point arrays

c(n) --- Array of dimension ‘n’ containing the coordinates of the centroid.

ff(k) --- Array of dimension ‘k’ containing the function values at the k vertices of a ‘complex’.

h(n*k) --- Array of dimension ‘n*k’ containing ‘n’ coordinates of each of the ‘k’ vertices of a ‘complex’

oldcc(n) --- Array of dimension ‘n’ containing the coordinates of the centroid of the first ‘complex’ to collapse in a row.

xg(n) --- Array of dimension ‘n’ containing the coordinates of a ‘complex’ vertex ‘ng’ which has the highest function value. It is also used as a workspace.

xup(n) --- Array of dimension ‘n’ containing the upper limits of the explicit constraints for a smaller ‘complex’.

xdn(n) --- Array of dimension ‘n’ containing the lower limits of the explicit constraints for a smaller ‘complex’.

6 Values for ‘alpha’, ‘beta’, ‘gama’, ‘phi’ and ‘phicpx’

The following procedure for selecting suitable values for the above control parameters is recommended for first time users. This should provide a feel for the ‘evop’ program, its idiosyncrasies, and nature of the objective function. Experienced users may completely bypass this procedure.

(i) Initially set ‘nrstrt’ to a high integer value, say 10 or 20.

(ii) Set ‘phi’ for low convergence accuracy, say for first two digit consistency. This means set ‘phi = 10^{-6}’ for single precision, or ‘phi = 10^{-14}’ for double precision arithmetic for word lengths of 8 or 16 digits respectively. Initially, set ‘phicpx’ to 10^{-1} for, or 10^{-9} for double precision. There is every chance that the convergence criteria will be satisfied before the collapse of a ‘complex’ can be detected.

(iii) Set ‘alpha’, ‘beta’ and ‘gama’ to their default values of 1.2, 0.5 and 2.0 respectively, and run the program.

(iv) Keeping ‘beta’ and ‘gama’ fixed, vary ‘alpha’ from a value greater than 1.0 to a value less than 2.0 for convergence ‘ier = 8’, with lowest number of function evaluation ‘nfunc’, and lowest function value ‘f’. Increase ‘phi’ upto 10^{-10} for double precision or 10^{-2} for single precision in steps for tighter convergence, and set ‘phi’ to the highest value that would still yield ‘ier = 8’. Note ‘alpha’ is the most sensitive parameter.

(v) Keeping ‘alpha’ and ‘gama’ fixed vary ‘beta’ above 0.0 to less than 1.0 for the criterion set out in (iv) above.

(vi) Keeping ‘alpha’ and ‘beta’ fixed vary ‘gama’ from 2.0 upwards for the criterion set out in (iv) above.

(vii) Repeat from step (iv) only if lower values of ‘nfunc’ and ‘f’ required.

(viii) Change ‘phicpx’ from a value two decades higher to two decades lower compared to ‘phi’, and observe the effects on ‘nfunc’ and ‘f’. Chose ‘phicpx’ for least ‘nfunc’ and ‘f’.

(ix) Using optimum ‘xt(n)’ and corresponding ‘xmax(n)’, ‘xmin(n)’, ‘xxmax(nic)’, ‘xxminnic)’, from (viii) above, run the program with same values for ‘alpha’, ‘beta’, ‘gama’, ‘phi’ and ‘phicpx’. Check the minimum obtained. If the tell tale evidence of discontinuities in the objective function or its first derivative is observed, as discussed in Section 2.2.1:’phicpx’ decrease ‘phicpx’ further.

(x) Using optimum ‘xt(n)’ from (ix) above and considerably smaller search space , run the program with same values for ‘alpha’, ‘beta’, ‘gama’, ‘phi’ and ‘phicpx’. Check whether a better minimum is obtained.

7 General Discussion

If the program terminates with ‘ier = 5’ or 6’ increase ‘limit’ by a factor of 2, and check whether improvement in the objective function value is obtained. If so, increase limit till ‘ier = 8’ is obtained. Note ‘kount = nfunc + kkt + m’, and whenever ‘kount’ exceeds ‘limit’ the program will terminate with ‘ier < 8’.

Often a difficult problem can be made to converge faster if shrinking boundaries (variable upper and lower limits on explicit constraints) are introduced. This facility should be used with care for objective functions with multiple minima, because this technique reduces the probability of locating the global minimum when started from far away region. Multiple minima are detected in a descending order provided the global minimum was not obtained at the end of the first run.

In models for optimization containing a mixture of continuous, discrete and integer variables, the feasibility of the rounded off trial point should be checked in the user written subroutine ‘expcons’, since rounding off may cause violation of a constraint. If so, reset the particular coordinate of the trial point to an integer or a discrete value, as the case may be, within the particular bound violated. For variable bounds recompute the bounds in ‘expcons’ to check whether the resetted trial point is still feasible. Otherwise, repeat the procedure until feasibility of the trial point is obtained. For a discrete variable, with constant upper and lower limits, just a simple set of allowed discrete values within the bounds would suffice. Only for variable bounds, the above elaborate feasibility checking procedure becomes necessary. For integer variables even with constant limits, the feasibility should always be checked after rounding off. The subroutine ‘evop’ treats such objective functions as if they contain multiple minima. The above procedure for selection of ‘evop’ control parameters thoroughly searches the feasible parameter space for global minimum and is mandatory for such problems. Only certain set of values for ‘alpha’, ‘beta’ and ‘gama’ may locate the global minimum with lowest ‘nfunc’ at the end of first run.

The subroutine will cope with trial point coordinates as small as 10^{-60} If the minimum of an objective function has coordinates lower than this value, and if the compiler can deal with such a low number than it should replace 10^{-60} in subroutine ‘explicit_2’ in ‘evop’.

8 Test Problems

A total of 36 problems have been included (2 Demonstration + 34 Test Problems) to thoroughly test the subroutine, as well as to provide guidance on how to write the ‘main calling program’, and the user provided subroutines ‘func’ for objective function value evaluation, ‘expcon’ for evaluation of explicit constraints, and ‘impcon’ for evaluation of implicit constraints.

The test problems are of the following types.

8.1 Unconstrained Objective Functions

UC --- Unconstrained
M --- Moving boundaries or moving explicit constraints

UC1_V1: 2 Variables Problem
ROSENBROCK, H. H.: "AN AUTOMATIC METHOD FOR FINDING THE GREATEST OR THE

LEAST VALUE OF A FUNCTION", COMPUTER JOURNAL, 1960, 3, PP.175 - 184.(MODIFIED FORM)

UC1M_V1: Same as UC1_V1 with Moving Explicit Constraints to Expedite Convergence

UC2_V1: 2 Variables Problem Discontinuous Objective Function
Modified Version of UC1_V1

UC2M_V1: Same as UC2_V1 with Moving Explicit Constraints to Expedite Convergence

8.2 Constrained Objective Functions with Inequality Constraints

C --- Constrained
M --- Moving boundaries or moving explicit constraints

C1_V1: 2 Variables Problem
W. R. KLINGMAN AND D. M. HIMMELBLAU: "NONLINEAR PROGRAMMING WITH THE AID OF A MULTIPLE-GRADIENT SUMMATION TECHNIQUE", JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINERY, 1964, 11, (4), PP.400 - 415. (MODIFIED FORM)

C1M_V1: Same as C1_V1 with Moving Explicit Constraints to Expedite Convergence

C2_V1: 2 Variables Problem
H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING THE GREATEST OR THE LEAST VALUE OF A FUNCTION", COMPUTER J., 1960, VOL.3, PP.175-184.(MODIFIED FORM)

C2M_V1: Same as C2_V1 with Moving Explicit Constraints to Expedite Convergence

C3_V1: 20 Variables Problem With Discontinuous First Derivatives
SCHWEFEL, H: "NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY 1981, PAGE 327, PROBLEM 3.6. (MODIFIED FORM)

C3M_V1: Same as C3_V1 with Moving Explicit Constraints to Expedite Convergence

C4_V1: Another 20 Variables Problem With Discontinuous First Derivatives
SCHWEFEL, H: "NUMERICAL OPTIMIZATION OF COMPUTER MODELS",WILEY 1981, PAGE 326, PROBLEM 3.5. (MODIFIED FORM)

C4M_V1: Same as C4_V1 with Moving Explicit Constraints to Expedite Convergence

C5_V1: Another 20 Variables Problem With Discontinuous First Derivatives
SCHWEFEL, H: 'NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY 1981, PAGE 327, PROBLEM 3.6. (MODIFIED FORM --- AN IMPLICIT CONSTRAINT IS INTRODUCED WHICH IS ACTIVE AT ITS LOWER BOUND).

C5M_V1: Same as C4_V1 with Moving Explicit Constraints to Expedite Convergence

C6_V1: Another 20 Variables Problem With Discontinuous First Derivatives

H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER MODELS", JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES 306 - 326 RESPECTIVELY.(MODIFIED FORM --- THE ORIGINAL PROBLEMS ARE UNCONSTRAINED. ACTIVE EXPLICIT AND IMPLICIT CONSTRAINTS HAVE BEEN INTRODUCED BY DR. S. N. GHANI.)

C6M_V1: Same as C5_V1 with Moving Explicit Constraints to Expedite Convergence

8.3 Constrained Objective Functions with Both Equality and Inequality Constraints

EC --- Equality constrained
M --- Moving boundaries or moving explicit constraints

EC1_V1: 4 Variables Problem

K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.282, P.166 (A BOOK). THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR COMPATIBLITY WITH SUBROUTINE "EVOP".

EC1M_V1: Same as EC1_V1 with Moving Explicit Constraints to Expedite Convergence

EC2_V1: 5 Variables Problem

K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.282, P.173 (A BOOK). THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR COMPATIBILITY WITH SUBROUTINE "EVOP".

EC2M_V1: Same as EC2_V1 with Moving Explicit Constraints to Expedite Convergence

EC3_V1: 4 Variables Problem

K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.282, P.168 AND 220 (A BOOK). THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR COMPATIBILITY WITH SUBROUTINE "EVOP".

EC3M_V1: Same as EC3_V1 with Moving Explicit Constraints to Expedite Convergence

8.4 Global Optimization

MM --- Multiple minima
M --- Moving boundaries or moving explicit constraints

MM1_V1: 2 Variable Problem
A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA", MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115), PP.569-574.

MM1M_V1: Same as MM1_V1 with Moving Explicit Constraints to Expedite Convergence

MM2_V1: Another 2 Variable Problem
A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA", MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115), PP.569-574.

MM2M_V1: Same as MM2_V1 with Moving Explicit Constraints to Expedite Convergence

MM3_V1: Another 2 Variable Problem
L. C. W. DIXON AND G. P. SZEGO (EDS): "TOWARDS GLOBAL MINIMISATION", NORTH HOLLAND PUBLISHING CO., 1975 (TRECCANI).

MM3M_V1: Same as MM3_V1 with Moving Explicit Constraints to Expedite Convergence

MM4_V1: 2 Variable Problem
AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186, HE - HESSE [HESSE 1973], (N=6), VOLUME 350, SPRINGER-VERLAG, 1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.

MM4M_V1: Same as MM4_V1 with Moving Explicit Constraints to Expedite Convergence

MM5_V1: 2 Variable Problem
AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186, Gn - GRIEWANK [GRIEWANK 1981], (N=2), VOLUME 350, SPRINGER-VERLAG, 1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.

MM5M_V1: Same as MM5_V1 with Moving Explicit Constraints to Expedite Convergence

MM6_V1: 10 Variable Problem
AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186, Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.

MM6M_V1: Same as MM6_V1 with Moving Explicit Constraints to Expedite Convergence

MM7_V1: Another 10 Variable Problem
AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186, Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.

MM7M_V1: Same as MM7_V1 with Moving Explicit Constraints to Expedite Convergence

8.5 Constrained Objective Functions with Mixed Continuous, Integer and Discrete Variables

MV --- Mixed Variables
M --- Moving boundaries or moving explicit constraints

All variables treated as continuous with integer/discrete variables constrained to take values from a thin band centered on integer/discrete values.

MV1_V1: 2 Variable Problem

SAME AS PROBLEM "UC1_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED TO TAKE VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.

MV1M_V1: Same as MV1_V1 with Moving Explicit Constraints to Expedite Convergence

MV2_V1: Another 2 Variable Problem

SAME AS PROBLEM "C2_V1.FOR" BUT VARIABLES "XT(1)" AND "XT(2)" ALLOWED TO TAKE VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF INTEGER VALUES AND PREASSIGNED DISCRETE VALUES RESPECTIVELY. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.

MV2M_V1: Same as MV2_V1 with Moving Explicit Constraints to Expedite Convergence

MV3_V1: Another 2 Variable Problem Same as MV3_V1

SAME AS PROBLEM "MM1_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED TO TAKE ONLY INTEGER VALUES *FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI*

MV3M_V1: Same as MV8_V1 with Moving Explicit Constraints to Expedite Convergence

MV4_V1: Another 2 Variable Problem

SAME AS PROBLEM "C1_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED TO TAKE ONLY INTEGER VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.

MV4M_V1: Same as MV9_V1 with Moving Explicit Constraints to Expedite Convergence

REFERENCES

1 S. N. Ghani: 'A Versatile Procedure for Optimization of a Nonlinear Nondifferentiable Constrained Objective Function', AERE R 13714, Harwell Laboratory, United Kingdom Atomic Energy Authority (UKAEA), December 1989, HMSO Publication Centre, PO Box 276, London SW8 5DT, ISBN 0-7058-1566-8.

2 S. N. Ghani:`Performance of Global Optimisation Algorithm EVOP for Non-linear Non-differentiable Constrained Objective Functions', Proceedings of the IEEE International Conference on Evolutionary Computing, November 2, 1995 - December 2, 1995 (IEEE ICEC'95), The University of Western Australia, Perth, Western Australia, pp. 320 - 325.

3 Aimo Torn and Antanas Zilinskas: 'Global Optimization', Lecture Notes in Computer Science, No 350, Springer-Verlag, 1989, ISBN 3-540-50871-6 or ISBN 0-387-50871-6.

4 Reiner Horst, Panos M. Pardalos and Nguyen V. Thoai: 'Introduction to Global Optimization (Nonconvex Optimization and its Application), Kluwer Academic Publishers, 2002, ISBN 0-7923-6756-1 or ISBN 0-7923-6574-7.

5 C A Floudes and C E Gounaris: 'A Review of Recent Advances in Global Optimization", J Glob Optim, 45:3-38, 2009.

6 Website: http://www.solver.com/global-optimization#Solving%20GO%20Problems

7 H. P. Schwel and R. Manner: 'Parallel Problem Solving From Nature', Proceedings of 1st Workshop, PPSN1, Dortmund, FRG, October 1-3, 1990, Lecture Notes in Computer Science, No. 496, Springer-Verlag, 1991, ISBN 3-540-54148-9 or ISBN 0-387-54148-9.

8 J. Stender (Ed.): ‘Parallel Genetic Algorithms – Theory and Applications’, Brainware GmbH, Berlin, Germany, IOS Press, 1992, ISBN 90 5199 087 1.

APPENDIX -- A

FORTRAN – IV Code

These Test Problems Were Run on VAX/VMS Machine. For Other Platforms EVOP Control Parameters 'alpha', 'beta', 'gama', 'phi' and 'phicpx' May Need to Be Slightly Adjusted.

C Demonstration Example 1

C rectifier_filter_v1

```
C
C
C UNCONSTRAINED
C
C
C              MAIN PROGRAMME FOR PROBLEM "RECTIFIER_FILTER_V1.FOR"
C
C  UC - UNCONSTRAINED
C
C  GHANI, S. N.: OPTIMIZATION OF THE DESIGH OF AN AC TO DC POWER
C                SUPPLY RECTIFIER FILTER FOR MINIMUM COST
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
C
      XT(1) = 1.0D0
      XT(2) = -1.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
```

```
      K=4
      IPRINT=2
      NRSTRT=10
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM:
"RECTIFIER_FILTER_V1.FOR"
     1'//)
  998 FORMAT(1X,'PROBLEM DEVELOPED BY S. N. GHANI:'//)
      CONTINUE
      END
C
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F, T1, T2
      DOUBLE PRECISION XT(2)
C

C     N=N
C             ABOVE TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1

C
      T1 = XT(1) * XT(1) + XT(2) * XT(2)
      T2 = XT(1) * XT(1) * XT(2) * XT(2)
      F = 60.D0 + T1 + (346.D0/T2) * (5.D0 + 346.D0/T1)
      RETURN
      END
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C

      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
      XT(2)=XT(2)
C                ABOVE TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C

      KKT=KKT+1
      KOUNT=KOUNT+1
      XMIN(1) = -99999.D0
      XMIN(2) = -99999.D0
```

```
      XMAX(1) =  99999.D0
      XMAX(2) =  99999.D0
      RETURN
      END
C
C

C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)

      N=N
      NIC=NIC
C          TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      M=M+1
      XX(1) = XT(1) + XT(2)
      XXMAX(1) = 999999.D0
      XXMIN(1) = -999999.D0
      RETURN
      END
```

C Demonstration Example 2

C grain_silo_v1

```
C
C CONSTRAINED
C
C
C               MAIN PROGRAMME FOR PROBLEM "GRAIN_SILO_V1.FOR"
C
C  C - CONSTRAINED
C
C  GHANI, S. N.: OPTIMIZATION OF THE DESIGN OF A GRAIN SILO FOR MINUMUM
C                                        COST
C
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
C
      XT(1) = 100.0D0
```

```
      XT(2) = 100.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=10
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "GRAIN_SILO_V1.FOR"
     1'//)
  998 FORMAT(1X,'PROBLEM DEVELOPED BY S. N. GHANI:'//)
      CONTINUE
      END
C
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F, BASE, ROOF, WALLS
      DOUBLE PRECISION XT(2)
C

C     N=N
C            ABOVE TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1
C
      BASE = 5000.D0 +
     1        400.D0 * 3.1415D0/4.D0 * (XT(1) + 4.D0) * (XT(1) + 4.D0)
      ROOF = 1000.D0 +
     1              20.D0 * 3.1415D0/4.D0 * (XT(1) + 4.D0) * (XT(1) +
                                                                 4.D0)

      WALLS = 3000.D0 +
      1         40.D0 * 3.1415D0 * XT(1) * XT(2) + 1000.D0 * XT(2)
      F = BASE + ROOF + WALLS
      RETURN
      END
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
```

```
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C

      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
      XT(2)=XT(2)
C               ABOVE TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C

      KKT=KKT+1
      KOUNT=KOUNT+1
      XMIN(1) = 0.D0
      XMIN(2) = 0.D0
      XMAX(1) = 100000.D0
      XMAX(2) = 100000.D0
      RETURN
      END
C
C
C
C
C
C
C
C             SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)

      N=N
      NIC=NIC
C          TO STOP THE VAX/VMS COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      M=M+1
      XX(1) = XT(1) * XT(1) * XT(2)
      XXMAX(1) = 1000000.D0
      XXMIN(1) = 254.655D0
      RETURN
      END
```

C Test Examples

Unconstrained:
uc1_v1

```
C
C              MAIN PROGRAMME FOR PROBLEM "UC1_V1.FOR"
C
C  UC - UNCONSTRAINED
C
C  ROSENBROCK, H. H.: "AN AUTOMATIC METHOD FOR FINDING THE
C                      GREATEST OR THE LEAST VALUE OF A
C                      FUNCTION", COMPUTER JOURNAL, 1960, 3,
C                      PP.175 - 184.
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
C
      XT(1)=-1.2D0
      XT(2)=1.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "UC1_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING
THE'
     1,' GREATEST OR LEAST'/1X,'VALUES OF A FUNCTION`, COMPUTER J.,'
```

```
     2,' 1960, VOL.3, P.175 - 184.'//)
      END
C
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C

      N=N
C            ABOVE TO STOP THE COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=100.D0*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
      F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
      RETURN
      END
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C

      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
      XT(2)=XT(2)
C               ABOVE TO STOP THE COMPILER FROM COMPLAINING
C

      KKT=KKT+1
      KOUNT=KOUNT+1
      XMIN(1)=-100000.D0
      XMIN(2)=-100000.D0
      XMAX(1)=100000.D0
      XMAX(2)=100000.D0
      RETURN
      END
C
C
C             SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)

      N=N
      NIC=NIC
C          TO STOP THE COMPILER FROM COMPLAINING
```

```
C
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=100000.D0
      XXMIN(1)=-100000.D0
      RETURN
      END
```

uc1m_v1:

```
C
C
C               MAIN PROGRAMME FOR PROBLEM "UC1M_V1.FOR"
C
C  UC - UNCONSTRAINED
C
C     SAME AS “UC1_V1.FOR” BUT WITH MOVING EXPLICIT CONSTRAINTS TO
C     EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
C
      XT(1)=-1.2D0
      XT(2)=1.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
```

```
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "UC1M_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING
THE'
     1,' GREATEST OR LEAST'/1X,'VALUES OF A FUNCTION`, COMPUTER J.,
196'
     2,'0 VOL.3, P.175 - 184.'//)
      END
C
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C
      N=N
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=100.D0*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
      F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
      RETURN
      END
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C           ABOVE TO STOP THE COMPILER FROM COMPLAINING
C
      KKT=KKT+1
      KOUNT=KOUNT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMAX(1)=1.D5
        XMAX(2)=1.D5
        XMIN(1)=-1.D5
        XMIN(2)=-1.D5
      ELSE
        DO 120 I=1, 2
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -100000.D0) THEN
              XMIN(I)=-100000.D0
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
```

```
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 100000.D0) THEN
              XMAX(I)=100000.D0
            ENDIF
          ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
C
      NIC=NIC
      N=N
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=100000.D0
      XXMIN(1)=-100000.D0
      RETURN
      END

C
C
C
C
C
C
```

uc2_v1:

```
C
C
C               MAIN PROGRAMME FOR PROBLEM "UC2_V1.FOR"
C
C  UC - UNCONSTRAINED AND DISCONTINUOUS OBJECTIVE FUNCTION
C
C     SAME AS “UC1_V1.FOR” BUT WITH DISCONTINUOUS OBJECTIVE FUNCTION.
C     MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
```

```
C
C      STARTING POINT FOR OPTIMISATION
C
      XT(1)=-1.2D0
      XT(2)=1.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "UC2_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING
THE'
     1,' GREATEST OR LEAST'/1X,'VALUES OF A FUNCTION`, COMPUTER J.,
196'
     2,'0, VOL.3, P.175 - 184.'//)
      END
C
C           SUBROUTINE FOR FUNCTION VALUE
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION COEFF, F
      DOUBLE PRECISION XT(2)
C
      N=N
C               ABOVE TO STOP THE COMPILER FROM COMPLAINING

      KOUNT=KOUNT+1
      KUT=KUT+1
      IF (XT(1) .GT. 500.D0) THEN
        COEFF=1000.D0
      ELSE
        IF (XT(1) .GT. 100.D0) THEN
          COEFF=500.D0
        ELSE
          IF (XT(1) .GT. 10.D0) THEN
            COEFF=300.D0
          ELSE
            COEFF=100.D0
```

```
            ENDIF
          ENDIF
        ENDIF
        F=COEFF*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
        F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
        RETURN
        END
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
        SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
        INTEGER IFLG,ISKP,KKT,KOUNT,M,N
        DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
        IFLG=IFLG
        ISKP=ISKP
        M=M
        N=N
        XT(1)=XT(1)
        XT(2)=XT(2)
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING

        KKT=KKT+1
        KOUNT=KOUNT+1
        XMIN(1)=-100000.D0
        XMIN(2)=-100000.D0
        XMAX(1)=100000.D0
        XMAX(2)=100000.D0
        RETURN
        END
C
C
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
        SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
        INTEGER KOUNT,M,N,NIC
        DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
        N=N
        NIC=NIC
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING

        KOUNT=KOUNT+1
        M=M+1
        XX(1)=XT(1)+XT(2)
        XXMAX(1)=100000.D0
        XXMIN(1)=-100000.D0
        RETURN
        END
```

uc2m_v1:

```
C
C
C               MAIN PROGRAMME FOR PROBLEM "UC2M_V1.FOR"
C
C  UC - UNCONSTRAINED AND DISCONTINUOUS OBJECTIVE FUNCTION
C
C     SAME AS “UC2_V1.FOR” BUT WITH MOVING EXPLICIT CONSTRAINTS TO
C     EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
C
      XT(1)=-1.2D0
      XT(2)=1.0D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.0D0
      DEL=1.0D-12
      PHI=1.0D-10
      PHICPX=1.0D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "UC2M_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING
THE'
     1,' GREATEST OR LEAST'/1X,'VALUES OF A FUNCTION`, COMPUTER J.,
196'
     2,'0 VOL.3, P.175 - 184.'//)
      END
C
C            SUBROUTINE FOR FUNCTION VALUE
```

```
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION COEFF, F
      DOUBLE PRECISION XT(2)
C
      N=N
C              ABOVE TO STOP THE COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      IF (XT(1) .GT. 500.D0) THEN
        COEFF=1000.D0
      ELSE
        IF (XT(1) .GT. 100.D0) THEN
          COEFF=500.D0
        ELSE
          IF (XT(1) .GT. 10.D0) THEN
            COEFF=300.D0
          ELSE
            COEFF=100.D0
          ENDIF
        ENDIF
      ENDIF
      F=COEFF*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
      F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
      RETURN
      END
C
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KKT=KKT+1
      KOUNT=KOUNT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMAX(1)=1.D5
        XMAX(2)=1.D5
        XMIN(1)=-1.D5
        XMIN(2)=-1.D5
      ELSE
        DO 120 I=1, 2
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -100000.D0) THEN
              XMIN(I)=-100000.D0
            ENDIF
          ENDIF
```

```
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 100000.D0) THEN
              XMAX(I)=100000.D0
            ENDIF
          ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C
C             SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C               ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=100000.D0
      XXMIN(1)=-100000.D0
      RETURN
      END
```

CONSTRAINED

c1_v1:

```
C
C
C
C               MAIN PROGRAMME FOR PROBLEM "C1_V1.FOR"
C
C - CONSTRAINED
C
C   W. R. KLINGMAN AND D. M. HIMMELBLAU: "NONLINEAR PROGRAMMING
C           WITH THE AID OF A MULTIPLE-GRADIENT SUMMATION TECHNIQUE",
C           JOURNAL OF THE ASSOCIATION FOR COMPUTING MACHINERY, 1964,
C           11, (4), PP.400 - 415.
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
```

```
C
      WRITE(9,999)
      WRITE(9,998)

C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=1.D0
      XT(2)=1.D0
C
C     CONTROL PARAMETERS FOR "EVOP"
C
      ALPHA=1.3D0
      BETA=0.3D0
      GAMA=2.D0
      DEL=1.D-10
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMISATION OF TEST PROBLEM: "C1_V1.FOR"'//)
  998 FORMAT(1X,' W. R. KLINGMAN AND D. M. HIMMELBLAU: "NONLINEAR
PROGR'
     1,'AMMING WITH'/1X,'THE AID OF A MULTIPLE-GRADIENT SUMMATION
TECHN'
     2,'IQUE", J. ACM, 1964'/1X,'VOL.11, (4), PP.400 - 415.'//)
      END
C
C                 SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DEXP,F
      DOUBLE PRECISION XT(2)
C
      N=N
C           ABOVE TO STOP COMPILER FROM COMPLAINING
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)-1.D0)*(XT(1)-1.D0)
      F=F-(XT(2)*XT(2)-.5D0)*(XT(2)*XT(2)-.5D0)/.132D0
      F=-(DEXP(F))
      RETURN
      END
```

```
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      N=N
      XT(1)=XT(1)
      XT(2)=XT(2)
C                 ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=.2D0
      XMIN(2)=.2D0
      XMAX(1)=2.D0
      XMAX(2)=2.D0
      RETURN
      END
C
C
CC
C
C
C
C
C
C
C        SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
C             ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      NIC=NIC
      XX(1)=XT(1)*XT(1)+XT(2)*XT(2)
      XXMAX(1)=4.D0
      XXMIN(1)=-99999.D0
      RETURN
      END
```

```
C    c1m_v1:
C
C
C               MAIN PROGRAMME FOR PROBLEM "C1M_V1.FOR"
C
C
C - CONSTRAINED
C
C     SAME AS "C1_V1.FOR" EXCEPT FOR MOVING EXPLICIT CONSTRAINTS TO
C     EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY
C     DR. S. N. GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),XDN(2),XG(2),XMAX(2),XMIN(2),
     1XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1),OLDCC(2)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)

C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=1.D0
      XT(2)=1.D0
C
C     CONTROL PARAMETERS FOR "EVOP"
C
      ALPHA=1.3D0
      BETA=0.3D0
      GAMA=2.D0
      DEL=1.D-10
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMISATION OF TEST PROBLEM: "C1M_V1.FOR"'//)
  998 FORMAT(1X,' W. R. KLINGMAN AND D. M. HIMMELBLAU: "NONLINEAR
PROGR'
     1,'AMMING WITH'/1X,'THE AID OF A MULTIPLE-GRADIENT SUMMATION
TECHN'
     2,'IQUE", J. ACM, 1964'/1X,'VOL.11, (4), PP.400 - 415.'//)
      END
```

```
C
C                 SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DEXP,F
      DOUBLE PRECISION XT(2)
C
      N=N
C             ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)-1.D0)*(XT(1)-1.D0)
      F=F-(XT(2)*XT(2)-.5D0)*(XT(2)*XT(2)-.5D0)/.132D0
      F=-(DEXP(F))
      RETURN
      END
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMAX(1)=2.D0
        XMAX(2)=2.D0
        XMIN(1)=0.2D0
        XMIN(2)=0.2D0
      ELSE
        DO 10 I=1, N
          XMIN(I)=0.2D0
          XMAX(I)=1.001D0*XT(I)
          IF (XMAX(I) .GT. 2.D0) THEN
            XMAX(I)=2.D0
          ENDIF
   10   CONTINUE
      ENDIF
      RETURN
      END
C
```

```
C                SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                 ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)*XT(1)+XT(2)*XT(2)
      XXMAX(1)=4.D0
      XXMIN(1)=-99999.D0
      RETURN
      END
C
C
C
C
C
C
C
```

c2_v1:

```
C
C
C
C                 MAIN PROGRAMME FOR PROBLEM "C2_V1.FOR"
C
C  C - CONSTRAINED
C
C  H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING THE GREATEST OR
C                     THE LEAST VALUE OF A FUNCTION", COMPUTER J.,
1960,
C                     VOL.3, PP.175-184.
C
      DOUBLE PRECISION C(3),FF(6),H(18),XDN(3),XG(3),XMAX(3),XMIN(3),
     1XT(3),XUP(3),XX(1),XXMAX(1),XXMIN(1),OLDCC(3)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=10.D0
      XT(2)=10.D0
      XT(3)=10.D0
C
```

```
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-8
      ICON=5
      LIMIT=6000
      KNT=25
      N=3
      NIC=1
      K=6
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT( 5X,'OPTIMIZATION OF TEST PROBLEM: "C2_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING
THE'
     1,'GREATEST OR'/1X,'LEAST VALUES OF A FUNCTION", COMPUTER
J.,1960,'
     2,' VOL. 33,PP. 175-184.'//)
      END
C
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(3)
C
      N=N
C              ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)*XT(2)*XT(3))
      RETURN
      END
C
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(3),XMIN(3),XT(3)
C
      IFLG=IFLG
```

```
      ISKP=ISKP
      M=M
      N=N
      XT(1)=XT(1)
      XT(2)=XT(2)
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=0.D0
      XMIN(2)=0.D0
      XMIN(3)=0.D0
      XMAX(1)=20.D0
      XMAX(2)=11.D0
      XMAX(3)=42.D0
      RETURN
      END
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(3)
C
      N=N
      NIC=NIC
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(2)+XT(3)+XT(3)
      XXMAX(1)=72.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

c2m_v1:

```
C
C
C
C                 MAIN PROGRAMME FOR PROBLEM "C2M_V1.FOR"
C
C  C - CONSTRAINED
C
C     SAME AS “C2_V1.FOR” EXCEPT FOR MOVING EXPLICIT CONSTRAINTS TO
C     EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY
C     DR. S. N. GHANI.
C
C
      DOUBLE PRECISION C(3),FF(6),H(18),XDN(3),XG(3),XMAX(3),XMIN(3),
     1XT(3),XUP(3),XX(1),XXMAX(1),XXMIN(1),OLDCC(3)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=10.D0
      XT(2)=10.D0
      XT(3)=10.D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-8
      ICON=5
      LIMIT=6000
      KNT=25
      N=3
      NIC=1
      K=6
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT( 5X,'OPTIMIZATION OF TEST PROBLEM: "C2M_V1.FOR"'//)
  998 FORMAT(1X,'H. H. ROSENBROCK: "AN AUTOMATIC METHOD FOR FINDING THE
G
     1REATEST OR'/1X,'LEAST VALUES OF A FUNCTION", COMPUTER
J.,1960,VOL.
     233,PP. 175-184.'//)
      END
```

```
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(3)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)*XT(2)*XT(3))
      RETURN
      END
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(3),XMIN(3),XT(3)
C
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=0.D0
        XMIN(2)=0.D0
        XMIN(3)=0.D0
        XMAX(1)=20.D0
        XMAX(2)=11.D0
        XMAX(3)=42.D0
      ELSE
        IF (XT(1) .GT. 0.D0) THEN
          XMAX(1)=1.001D0*XT(1)
            IF (XMAX(1) .GT. 20.D0) THEN
              XMAX(1)=20.D0
            ENDIF
        ELSE
          XMAX(1)=1.D-1
        ENDIF
        IF (XT(2) .GT. 0.D0) THEN
          XMAX(2)=1.001D0*XT(2)
            IF (XMAX(2) .GT. 11.D0) THEN
              XMAX(2)=11.D0
            ENDIF
        ELSE
          XMAX(2)=1.D-1
        ENDIF
        IF (XT(3) .GT. 0.D0) THEN
          XMAX(3)=1.001D0*XT(3)
            IF (XMAX(3) .GT. 42.D0) THEN
              XMAX(3)=42.D0
            ENDIF
        ELSE
          XMAX(3)=1.D-1
        ENDIF
```

```
      ENDIF
      RETURN
      END
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(3)
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(2)+XT(3)+XT(3)
      XXMAX(1)=72.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

c3_v1:

```
C
C
C
C               MAIN PROGRAM FOR TEST FUNCTION: "C3_V1.FOR"
C
C  SCHWEFEL, H: "NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY
1981,
C                 PAGE 327, PROBLEM 3.6.
C
      DOUBLE PRECISION C(20),FF(21),H(420),OLDCC(20),
     1XDN(20),XG(20),XMAX(20),XMIN(20),XT(20),XUP(20),XX(1),
     2XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
      WRITE(9,997)
C
C
C         STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1,N
        XT(I)=1.D0
  100 CONTINUE
```

```
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=0.3D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-12
      PHICPX=1.D-15
      ICON=5
      LIMIT=50000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(5X,'OPTIMISATION OF TEST PROBLEM: "C3_V1.FOR"'//)
  998 FORMAT(1X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS."'/1X,'JOHN WILEY, 1981, PROBLEMS 2.22 AND 3.6 PAGES
306',
     2'306 AND 327 RESPECTIVELY.')
  997 FORMAT(1X,'THE ORIGINAL PROBLEMS ARE UNCONSTRAINED. DUMMY
EXPLIC',
     1'IT AND IMPLICIT'/1X,'CONSTRAINTS HAVE BEEN INTRODUCED FOR
COMPA',
     2'TABILITY WITH SUBROUTINE'/1X,'"EVOP". IN ORDER TO EXPEDITE
CONV',
     3'ERGENCE MOVING UPPER AND LOWER BOUNDARIES CONSTITUTE EXPLICIT',
     3' CONSTRAINTS.'//)
      END
C
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION DABS,F
C
      DOUBLE PRECISION XT(1)
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=DABS(XT(1))
      DO 100 I=2, N
        F=F*DABS(XT(I))
  100 CONTINUE
      DO 110 I=1, N
        F=F+DABS(XT(I))
  110 CONTINUE
      F=F+1.D0
      RETURN
      END
C
C
```

```
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
      XT(1)=XT(1)
      M=M
C                   ABOVE IS INCLUDED TO STOP THE COMPILER FROM
COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 120 I=1, N
        XMIN(I)=-1.01D0
        XMAX(I)=1.01D0
  120 CONTINUE
      RETURN
      END
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                 ABOVE IS INCLUDED TO STOP THE COMPILER FROM
COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=-21.D0
      RETURN
      END
```

c3m_v1:

```
C
C
C
C               MAIN PROGRAM FOR TEST FUNCTION: "C3M_V1.FOR"
C
C  SCHWEFEL, H: "NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY
1981,
C                 PAGE 327, PROBLEM 3.6.
C
      DOUBLE PRECISION C(20),FF(21),H(420),OLDCC(20),
     1XDN(20),XG(20),XMAX(20),XMIN(20),XT(20),XUP(20),XX(1),
     2XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
      WRITE(9,997)
C
C
C       STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1,N
        XT(I)=1.D0
  100 CONTINUE
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.3D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-16
      ICON=5
      LIMIT=120000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(5X,'OPTIMISATION OF TEST PROBLEM: "C3M_V1.FOR"'//)
  998 FORMAT(1X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS."'/1X,'JOHN WILEY, 1981, PROBLEMS 2.22 AND 3.6 PAGES ',
     2'306 AND 327 RESPECTIVELY.')
  997 FORMAT(1X,'THE ORIGINAL PROBLEMS ARE UNCONSTRAINED. DUMMY EXPLI',
     1'CIT AND IMPLICIT'/1X,'CONSTRAINTS HAVE BEEN INTRODUCED FOR COM',
```

```
     2'PATIBILITY WITH SUBROUTINE'/1X,'"EVOP". IN ORDER TO EXPEDITE C',
     3'ONVERGENCE MOVING UPPER AND LOWER BOUNDARIES CONSTITUTE EXPLIC',
     3'IT CONSTRAINTS.'//)
      END
C
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION DABS,F
C
      DOUBLE PRECISION XT(1)
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=DABS(XT(1))
      DO 100 I=2, N
        F=F*DABS(XT(I))
  100 CONTINUE
      DO 110 I=1, N
        F=F+DABS(XT(I))
  110 CONTINUE
      F=F+1.D0
      RETURN
      END
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
C                    ABOVE INCLUDED TO STOP THE COMPILER FROM
COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=1.01D0
          XMIN(I)=-1.01D0
  109   CONTINUE
      ELSE
        DO 120 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.01D0) THEN
                XMIN(I)=-1.01D0
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.01D0) THEN
              XMAX(I)=1.01D0
             ENDIF
```

```
          ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                   ABOVE INCLUDED TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=-21.D0
      RETURN
      END
```

C4_v1:

```
C
C               MAIN PROGRAM FOR PROBLEM "C4_V1.FOR"
C  SCHWEFEL, H: "NUMERICAL OPTIMIZATION OF COMPUTER MODELS",WILEY 1981,
C                 PAGE 326, PROBLEM 3.5.
C
      DOUBLE PRECISION C(20),FF(21),H(420),
     1XDN(20),XG(20),XMAX(20),XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),
     2XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
```

```
      N=20
      DO 100 I=1, N
        XT(I)=1.D0
  100 CONTINUE
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-15
      ICON=5
      LIMIT=200000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: "C4_V1.FOR"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES ',
     2'306 AND 326 RSPCTVLY'/1X,'THE ORIGINAL PROBLEMS ARE UNCONSTR',
     3'AINED. DUMMY EXPLICIT AND IMPLICIT'/1X,'CONSTRAINTS HAVE BEE',
     4'N INTRODUCED FOR COMPATIBILITY WITH THE SUBROUTINE.'//)
      END
C
C
C
C
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION AXT,DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-1.D30
      DO 100 I=1, N
        AXT=DABS(XT(I))
          IF (AXT .GT. F) THEN
            F=AXT
          ENDIF
  100 CONTINUE
      F=F+1.D0
      RETURN
      END
```

```
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,N
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      XT(1)=XT(2)
C                ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 100 I=1, N
        XMIN(I)=-1.01D0
        XMAX(I)=1.01D0
  100 CONTINUE
      RETURN
      END
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                 ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=-21.D0
      RETURN
      END
```

c4m_v1:

```
C               MAIN PROGRAM FOR PROBLEM "C4M_V1.FOR"
C
C  SCHWEFEL, H: "NUMERICAL OPTIMIZATION OF COMPUTER MODELS",WILEY 1981,
C                 PAGE 326, PROBLEM 3.5.
C
      DOUBLE PRECISION C(20),FF(21),H(420),
     1XDN(20),XG(20),XMAX(20),XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),
     2XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1, N
        XT(I)=1.D0
  100 CONTINUE
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-16
      ICON=5
      LIMIT=240000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,
     2XUP,XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: "C4M_V1.FOR"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES ',
     2'306 AND 326 RSPCTVLY'/1X,'THE ORIGINAL PROBLEMS ARE UNCONSTR',
     3'AINED DUMMY EXPLICIT AND IMPLICIT'/1X,'CONSTRAINTS HAVE BEEN',
     4' INTRODUCED FOR COMPATIBILITY WITH SUBROUTINE'/1X,'"EVOP". IN',
     5' ORDER TO EXPEDITE CONVERGENCE MOVING UPPER AND LOWER'/1X,
     6'BOUNDARIES CONSTITUTE THE EXPLICIT CONSTRAINTS.'//)
      END
C
C
C
C
```

```
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION AXT,DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-1.D30
      DO 100 I=1, N
        AXT=DABS(XT(I))
        IF (AXT .GT. F) THEN
          F=AXT
        ENDIF
  100 CONTINUE
      F=F+1.D0
      RETURN
      END
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
C                    ABOVE TO STOP THE COMPILER FROM COMPILING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=1.01D0
          XMIN(I)=-1.01D0
  109   CONTINUE
      ELSE
        DO 100 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.01D0) THEN
              XMIN(I)=-1.01D0
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.01D0) THEN
              XMAX(I)=1.01D0
            ENDIF
          ENDIF
  100   CONTINUE
      ENDIF
```

```
      RETURN
      END
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=-21.D0
      RETURN
      END
```

c5_v1:

```
C
C
C
C                MAIN PROGRAM FOR PROBLEM "C5_V1.FOR"
C
C  SCHWEFEL, H: 'NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY
1981,
C                 PAGE 327, PROBLEM 3.6. MODIFIED.  AN IMPLICIT
CONSTRAINT
C                 IS INTRODUCED WHICH IS ACTIVE AT ITS LOWER BOUND.
C
C
      DOUBLE PRECISION C(20),FF(21),H(420),XDN(20),XG(20),XMAX(20),
     1XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1, N
```

```
        XT(I)=1.D0
  100 CONTINUE
C
C     CONTROL PARAMETERS FOR EVOP
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: "C5_V1.FOR"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMIZATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.22 AND 3.6 PAGES ',
     2'306 AND 327 RESPCTVLY.'/1X,'THE ORIGINAL PROBLEMS ARE UNCONS',
     3'TRAINED. AN ACTIVE IMPLICIT'/1X,'CONSTRAINT HAS BEEN INTRODU',
     4'CED BY DR. S. N. GHANI TO MAKE THE PROBLEM EVEN MORE PATHOLO',
     5'GICAL.'//)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=DABS(XT(1))
      DO 100 I=2, N
        F=F*DABS(XT(I))
  100 CONTINUE
      DO 110 I=1, N
        F=F+DABS(XT(I))
  110 CONTINUE
      RETURN
      END
C
C
C
C
C
```

```
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      XT(1)=XT(1)
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 100 I=1, N
        XMAX(I)=1.01D0
        XMIN(I)=-1.01D0
  100 CONTINUE
      RETURN
      END
C
C
C                SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=5.D0
      RETURN
      END
```

c5m_v1:

```
C
C                MAIN PROGRAM FOR PROBLEM "C5M_V1.FOR"
C
C  SCHWEFEL, H: 'NUMERICAL OPTIMISATION OF COMPUTER MODELS", WILEY
1981,
C                 PAGE 327, PROBLEM 3.6. MODIFIED.  AN IMPLICIT
CONSTRAINT
C                 IS INTRODUCED WHICH IS ACTIVE AT ITS LOWER BOUND.
C
C
      DOUBLE PRECISION C(20),FF(21),H(420),XDN(20),XG(20),XMAX(20),
     1XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1, N
        XT(I)=1.D0
  100 CONTINUE
C
C     CONTROL PARAMETERS FOR EVOP
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: "C5M_V1.S"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMIZATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.22 AND 3.6 PAGES ',
     2'306 AND 327 RESPCTVLY.'/1X,'THE ORIGINAL PROBLEMS ARE UNCONS',
     3'TRAINED. AN ACTIVE IMPLICIT'/1X,'CONSTRAINT HAS BEEN INTRODU',
     4'CED BY DR. S. N. GHANI TO MAKE THE PROBLEM EVEN MORE PATHOLO',
     5'GICAL.'//)
      END
```

```
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=DABS(XT(1))
      DO 100 I=2, N
        F=F*DABS(XT(I))
  100 CONTINUE
      DO 110 I=1, N
        F=F+DABS(XT(I))
  110 CONTINUE
      RETURN
      END
C
C
C
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
C                      ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=1.01D0
          XMIN(I)=-1.01D0
  109   CONTINUE
      ELSE
        DO 120 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.01D0) THEN
              XMIN(I)=-1.01D0
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
```

```
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.01D0) THEN
              XMAX(I)=1.01D0
            ENDIF
          ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                    ABOVE TO STOP COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=21.D0
      XXMIN(1)=5.D0
      RETURN
      END
```

c6_v1:

```
C
C
C
C                MAIN PROGRAM FOR PROBLEM "C6_V1.FOR"
C
C   C - CONSTRAINED
C
C   H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER MODELS",
C                    JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES
C                    306 - 326 RESPECTIVELY.
C   THE ORIGINAL PROBLEMS ARE UNCONSTRAINED. ACTIVE EXPLICIT AND
C   IMPLICIT CONSTRAINTS HAVE BEEN INTRODUCED BY DR. S. N. GHANI.
C   IN ORDER TO EXPEDITE CONVERGENCE MOVING UPPER BOUNDARY FORMS A
C   PART OF THE EXPLICIT CONSTRAINTS.
C
      DOUBLE PRECISION C(20),FF(21),H(420),XDN(20),XG(20),XMAX(20),
     1XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1, N
        XT(I)=10.D0
  100 CONTINUE
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-8
      ICON=5
      LIMIT=120000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "C6_V1.FOR"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES ',
```

```
     2'306 AND 326 RSPCTVLY.'/1X,'THE ORIGINAL PROBLEMS ARE UNCONST',
     3'RAINED. ACTIVE EXPLICIT AND'/1X,'IMPLICIT CONSTRAINTS HAVE ',
     4'BEEN INTRODUCED BY DR S. N. GHANI. IN ORDER'/1X,'TO EXPEDITE',
     5'CONVERGENCE MOVING BOUNDARIES FORM PART OF EXPLICIT CONSTRAI',
     6'NTS.'//)

      END
C
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION AXT,DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-1.D30
      DO 100 I=1, N
        AXT=DABS(XT(I))
        IF (AXT .GT. F) THEN
          F=AXT
        ENDIF
  100 CONTINUE
      RETURN
      END
C
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      XT(1)=XT(1)
C                  ABOVE TO STOP THE COMPLIER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 100 I=1, N
        XMIN(I)=.25D0
        XMAX(I)=10.1D0
  100 CONTINUE
      RETURN
      END
C
C
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS.
```

```
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=201.D0
      XXMIN(1)=5.D0
      RETURN
      END
```

c6m_v1:

```
C
C
C
C                MAIN PROGRAM FOR PROBLEM "C6M_V1.FOR"
C
C   C - CONSTRAINED
C
C   H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER MODELS",
C                     JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES
C                     306 - 326 RESPECTIVELY.
C   THE ORIGINAL PROBLEMS ARE UNCONSTRAINED. ACTIVE EXPLICIT AND
C   IMPLICIT CONSTRAINTS HAVE BEEN INTRODUCED BY DR. S. N. GHANI.
C   IN ORDER TO EXPEDITE CONVERGENCE MOVING UPPER BOUNDARY FORMS A
C   PART OF THE EXPLICIT CONSTRAINTS.
C
      DOUBLE PRECISION C(20),FF(21),H(420),XDN(20),XG(20),XMAX(20),
     1XMIN(20),XT(20),XUP(20),XX(1),XXMAX(1),XXMIN(1),OLDCC(20)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      N=20
      DO 100 I=1, N
        XT(I)=10.D0
  100 CONTINUE
```

```
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-8
      ICON=5
      LIMIT=60000
      KNT=25
      NIC=1
      K=21
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  110 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 110
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "C6M_V1.FOR"'//)
  998 FORMAT(5X,'H. P. SCHWEFEL: "NUMERICAL OPTIMISATION OF COMPUTER ',
     1'MODELS",'/1X,'JOHN WILEY, 1981, PROBLEMS 2.21 AND 3.5 PAGES ',
     2'306 AND 326 RSPCTVLY.'/1X,'THE ORIGINAL PROBLEMS ARE UNCONST',
     3'RAINED. ACTIVE EXPLICIT AND'/1X,'IMPLICIT CONSTRAINTS HAVE ',
     4'BEEN INTRODUCED BY DR S. N. GHANI. IN ORDER'/1X,'TO EXPEDITE',
     5'CONVERGENCE MOVING BOUNDARIES FORM PART OF EXPLICIT CONSTRAI',
     6'NTS.'//)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER I,KOUNT,KUT,N
      DOUBLE PRECISION AXT,DABS,F
      DOUBLE PRECISION XT(1)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-1.D30
      DO 100 I=1, N
        AXT=DABS(XT(I))
        IF (AXT .GT. F) THEN
          F=AXT
        ENDIF
  100 CONTINUE
      RETURN
      END
C
C
C
C
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
```

```
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I,IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(1),XMIN(1),XT(1)
C
      IFLG=IFLG
      ISKP=ISKP
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=10.1D0
          XMIN(I)=0.25D0
  109   CONTINUE
      ELSE
        DO 110 I=1, N
          XMIN(I)=.25D0
          IF (XT(I) .GT. 0.25D0) THEN
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 10.1D0) THEN
              XMAX(I)=10.1D0
            ENDIF
          ELSE
            XMAX(I)=.2502D0
          ENDIF
  110   CONTINUE
      ENDIF
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS.
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER I,KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(1)
C
      NIC=NIC
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 100 I=1, N
        XX(1)=XX(1)+XT(I)
  100 CONTINUE
      XXMAX(1)=201.D0
      XXMIN(1)=5.D0
      RETURN
      END
```

EQUALITY CONSTRAINED

ec1_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "EC1_V1.FOR"
C
C       EC - EQUALITY CONSTRAINED
C
C       K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.166 (A BOOK).
C       THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C       COMPATIBLITY WITH SUBROUTINE "EVOP".
C
      DOUBLE PRECISION C(4),FF(8),H(32),OLDCC(4),XDN(4),XG(4),XMAX(4),
     1XMIN(4),XT(4),XUP(4),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=0.D0
      XT(2)=0.D0
      XT(3)=2.D0
      XT(4)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=0.3D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-9
      ICON=5
      LIMIT=6000
      KNT=25
      N=4
      NIC=1
      K=8
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
```

```
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC1_V1.FOR".'//)
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NON',
     1'LINEAR PROGRAMMING,'/1X,'CODES", SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS,',
     3' VOL.282, P.166 (A BOOK).'/1X,'THE PROBLEM HAS BEEN SLIG',
     4'HTLY MODIFIED BY DR. S. N. GHANI FOR COMPATIBILITY ',
     5'WITH SUBROUTINE "EVOP".'//)
      END
C
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DSQRT,F
      DOUBLE PRECISION XT(4)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=(XT(1)-1.D0)*(XT(1)-1.D0)+(XT(1)-XT(2))*(XT(1)-XT(2))
      F=F+(XT(2)-XT(3))*(XT(2)-XT(3))*(XT(2)-XT(3))*(XT(2)-XT(3))
      F=F+XT(4)*(XT(1)*(1.D0+XT(2)*XT(2))+XT(3)*XT(3)*XT(3)*XT(3))
      F=F-XT(4)*(4.D0+3.D0*DSQRT(2.D0))
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(4),XMIN(4),XT(4)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      N=N
      XT(1)=XT(1)
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=0.D0
      XMIN(2)=0.D0
      XMIN(3)=0.D0
      XMIN(4)=0.D0
      XMAX(1)=2.D0
      XMAX(2)=2.D0
      XMAX(3)=2.D0
      XMAX(4)=4.D0
      RETURN
      END
```

```
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION DSQRT
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(4)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)*(1.D0+XT(2)*XT(2))
      XX(1)=XX(1)+XT(3)*XT(3)*XT(3)*XT(3)-4.D0-3.D0*DSQRT(2.D0)
      XXMAX(1)=10.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

C ec1m_v1:

```
C
C
C              MAIN PROGRAM FOR PROBLEM "EC1M_V1.FOR"
C
C       EC - EQUALITY CONSTRAINED
C
C       K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.166 (A BOOK).
C       THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C       COMPATIBLITY WITH SUBROUTINE "EVOP".
C
      DOUBLE PRECISION C(4),FF(8),H(32),OLDCC(4),XDN(4),XG(4),XMAX(4),
     1XMIN(4),XT(4),XUP(4),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
```

```
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=0.D0
      XT(2)=0.D0
      XT(3)=2.D0
      XT(4)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=0.3D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-9
      ICON=5
      LIMIT=6000
      KNT=25
      N=4
      NIC=1
      K=8
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC1M_V1.FOR".'//)
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NON',
     1'LINEAR PROGRAMMING,'/1X,'CODES", SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND MATHEMATICAL SYSTEMS,',
     3' VOL.282, P.166 (A BOOK).'/1X,'THE PROBLEM HAS BEEN SLIG',
     4'HTLY MODIFIED BY DR. S. N. GHANI FOR COMPATIBILITY ',
     5'WITH SUBROUTINE "EVOP".'//)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DSQRT,F
      DOUBLE PRECISION XT(4)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=(XT(1)-1.D0)*(XT(1)-1.D0)+(XT(1)-XT(2))*(XT(1)-XT(2))
      F=F+(XT(2)-XT(3))*(XT(2)-XT(3))*(XT(2)-XT(3))*(XT(2)-XT(3))
      F=F+XT(4)*(XT(1)*(1.D0+XT(2)*XT(2))+XT(3)*XT(3)*XT(3)*XT(3))
      F=F-XT(4)*(4.D0+3.D0*DSQRT(2.D0))
      RETURN
      END
```

```
C
C
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(4),XMIN(4),XT(4)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                       ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, 3
          XMAX(I)=2.D0
          XMIN(I)=0.D0
  109   CONTINUE
        XMAX(4)=4.D0
        XMIN(4)=0.D0
      ELSE
        DO 100 I=1, 3
          XMIN(I)=0.0D0
          IF (XT(I) .GT. 0.0D0) THEN
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 2.D0) THEN
              XMAX(I)=2.D0
            ENDIF
          ELSE
            XMAX(I)=1.D-1
          ENDIF
  100   CONTINUE
        XMIN(4)=0.D0
        IF (XT(4) .GT. 0.0D0) THEN
          XMAX(4)=1.001D0*XT(4)
          IF (XMAX(4) .GT. 4.D0) THEN
            XMAX(4)=4.D0
          ENDIF
        ELSE
          XMAX(4)=1.D-1
        ENDIF
      ENDIF
      RETURN
      END
C
C
C
C
C
C
```

```
C                SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION DSQRT
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(4)
C
      N=N
      NIC=NIC
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)*(1.D0+XT(2)*XT(2))
      XX(1)=XX(1)+XT(3)*XT(3)*XT(3)*XT(3)-4.D0-3.D0*DSQRT(2.D0)
      XXMAX(1)=10.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

c ec2_v1:

```
c
C
C              MAIN PROGRAM FOR PROBLEM "EC2_V1.FOR"
C
C       EC - EQUALITY CONSTRAINED
C
C       K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.173 (A BOOK).
C       THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C       COMPATIBILITY WITH SUBROUTINE "EVOP".
C
C
      DOUBLE PRECISION C(5),FF(10),H(50),OLDCC(5),XDN(5),XG(5),XMAX(5),
     1XMIN(5),XT(5),XUP(5),XX(3),XXMAX(3),XXMIN(3)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C          STARTING POINT FOR OPTIMISATION
      XT(1)=0.D0
      XT(2)=0.D0
```

```
      XT(3)=.397519652D0
      XT(4)=.600317377D0
      XT(5)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-8
      PHICPX=1.D-9
      ICON=5
      LIMIT=50000
      KNT=25
      N=5
      NIC=3
      K=10
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC2_V1.FOR".'//)
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NO',
     1'NLINEAR PROGRAMMING'/1X,'CODES, SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND'/1X,'MATHEMATICAL SYSTEMS,',
     3' VOL.282, P.173 (A BOOK).'/1X,'THE PROBLEM HAS BEEN SLIGH',
     4'TLY MODIFIED BY DR. S. N. GHANI FOR'/1X,
     5'COMPATIBILITY WITH SUBROUTINE "EVOP".'//)
      END
C
C
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(5)
C
      N=N
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=24.55D0*XT(1)+26.75D0*XT(2)+39.D0*XT(3)+40.5D0*XT(4)
      F=-(F)+XT(5)*(XT(1)+XT(2)+XT(3)+XT(4)-1.D0)
      RETURN
      END
C
C
C
```

```
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(5),XMIN(5),XT(5)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      N=N
      XT(1)=XT(1)
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=0.D0
      XMIN(2)=0.D0
      XMIN(3)=0.D0
      XMIN(4)=0.D0
      XMIN(5)=0.D0
      XMAX(1)=1.D0
      XMAX(2)=1.D0
      XMAX(3)=1.D0
      XMAX(4)=1.D0
      XMAX(5)=4.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION DSQRT
      DOUBLE PRECISION XX(3),XXMAX(3),XXMIN(3),XT(5)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=2.3D0*XT(1)+5.6D0*XT(2)+11.1D0*XT(3)+1.3D0*XT(4)
      XX(3)=.53D0*XT(1)*.53D0*XT(1)+.44D0*XT(2)*.44D0*XT(2)+4.5D0*XT
     1  (3)*4.5D0*XT(3)
      XX(3)=XX(3)+.79D0*XT(4)*.79D0*XT(4)
      XX(2)=12.D0*XT(1)+11.9D0*XT(2)+41.8D0*XT(3)+52.1D0*XT(4)-1.645D0
     1  *DSQRT(XX(3))
      XX(3)=XT(1)+XT(2)+XT(3)+XT(4)-1.D0
      XXMAX(1)=5.5D0
      XXMIN(1)=5.D0
      XXMAX(2)=45.300864D0
      XXMIN(2)=12.D0
      XXMAX(3)=0.D0
      XXMIN(3)=-.5D0
      RETURN
      END
```

ec2m_v1:

```
C
C              MAIN PROGRAM FOR PROBLEM "EC2M_V1.FOR"
C
C       EC - EQUALITY CONSTRAINED
C
C       K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.173 (A BOOK).
C       THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C       COMPATIBILITY WITH SUBROUTINE "EVOP".
C
C
      DOUBLE PRECISION C(5),FF(10),H(50),OLDCC(5),XDN(5),XG(5),XMAX(5),
     1XMIN(5),XT(5),XUP(5),XX(3),XXMAX(3),XXMIN(3)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C         STARTING POINT FOR OPTIMISATION
      XT(1)=0.D0
      XT(2)=0.D0
      XT(3)=.397519652D0
      XT(4)=.600317377D0
      XT(5)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=0.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-8
      PHICPX=1.D-12
      ICON=5
      LIMIT=600000
      KNT=25
      N=5
      NIC=3
      K=10
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC2M_V1.FOR".'//)
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NO',
```

```
     1'NLINEAR PROGRAMMING'/1X,'CODES, SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND'/1X,'MATHEMATICAL SYSTEMS,',
     3' VOL.282, P.173 (A BOOK).'/1X,'THE PROBLEM HAS BEEN SLIGH',
     4'TLY MODIFIED BY DR. S. N. GHANI FOR'/1X,
     5'COMPATIBILITY WITH SUBROUTINE "EVOP".'//)
      END
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(5)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=24.55D0*XT(1)+26.75D0*XT(2)+39.D0*XT(3)+40.5D0*XT(4)
      F=-(F)+XT(5)*(XT(1)+XT(2)+XT(3)+XT(4)-1.D0)
      RETURN
      END
C
C
C
C
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(5),XMIN(5),XT(5)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, 4
          XMAX(I)=1.D0
          XMIN(I)=0.D0
  109   CONTINUE
        XMAX(5)=4.D0
        XMIN(5)=0.D0
      ELSE
        DO 100 I=1, 4
          XMIN(I)=0.D0
          IF (XT(I) .GT. 0.D0) THEN
            XMAX(I)=1.01D0*XT(I)
```

```
            IF (XMAX(I) .GT. 1.D0) THEN
              XMAX(I)=1.D0
            ENDIF
          ELSE
            XMAX(I)=1.D-1
          ENDIF
  100   CONTINUE
        XMIN(5)=0.D0
        IF (XT(5) .GT. 0.D0) THEN
          XMAX(5)=1.01D0*XT(5)
          IF (XMAX(5) .GT. 4.D0) THEN
            XMAX(5)=4.D0
          ENDIF
        ELSE
          XMAX(5)=1.D-1
        ENDIF
      ENDIF
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION DSQRT
      DOUBLE PRECISION XX(3),XXMAX(3),XXMIN(3),XT(5)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=2.3D0*XT(1)+5.6D0*XT(2)+11.1D0*XT(3)+1.3D0*XT(4)
      XX(3)=.53D0*XT(1)*.53D0*XT(1)+.44D0*XT(2)*.44D0*XT(2)+4.5D0*XT
     1  (3)*4.5D0*XT(3)
      XX(3)=XX(3)+.79D0*XT(4)*.79D0*XT(4)
      XX(2)=12.D0*XT(1)+11.9D0*XT(2)+41.8D0*XT(3)+52.1D0*XT(4)-1.645D0
     1  *DSQRT(XX(3))
      XX(3)=XT(1)+XT(2)+XT(3)+XT(4)-1.D0
      XXMAX(1)=5.5D0
      XXMIN(1)=5.D0
      XXMAX(2)=45.300864D0
      XXMIN(2)=12.D0
      XXMAX(3)=0.D0
      XXMIN(3)=-.5D0
      RETURN
      END
```

ec3_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "EC3_V1.FOR"
C
C      EC - EQUALITY CONSTRAINED
C
C      K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.168 AND 220 (A BOOK).
C       THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C       COMPATIBILITY WITH SUBROUTINE "EVOP".
C
C
      DOUBLE PRECISION C(4),FF(8),H(32),OLDCC(4),XDN(4),XG(4),XMAX(4),
     1XMIN(4),XT(4),XUP(4),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)= 0.7000001D0
      XT(2)= 0.2D0
      XT(3)= 0.1D0
      XT(4)=1.D0
C
C
C      CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=12000
      KNT=25
      N=4
      NIC=1
      K=8
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC3_V1.FOR".'//)
```

```
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NON',
     1'LINEAR PROGRAMMING'/1X,'CODES, SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND'/1X,'MATHEMATICAL SYSTEMS,',
     3' VOL.282, PP.168 AND 220 (A BOOK).'/1X,'THE PROBLEM HAS',
     4' BEEN SLIGHTLY MODIFIED BY DR.S. N. GHANI FOR'/1X,'COMP',
     5'ATIBILITY WITH SUBROUTINE "EVOP".'//)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DLOG,F
      DOUBLE PRECISION XT(4)

C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=8204.37D0*DLOG((XT(1)+XT(2)+XT(3)+3.D-2)/(9.D-2*XT(1)+XT
     1 (2)+XT(3)+3.D-2))
      F=F+9008.72D0*DLOG((XT(2)+XT(3)+3.D-2)/(7.D-2*XT(2)+XT(3)+3.D-2))
      F=F+9330.46D0*DLOG((XT(3)+3.D-2)/(.13D0*XT(3)+3.D-2))
      F=F+XT(4)*(XT(1)+XT(2)+XT(3)-1.D0)
      RETURN
      END
C
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(4),XMIN(4),XT(4)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      N=N
      XT(1)=XT(1)
C                        ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=0.D0
      XMIN(2)=0.D0
      XMIN(3)=0.D0
      XMIN(4)=0.D0
      XMAX(1)=1.D0
      XMAX(2)=1.D0
      XMAX(3)=1.D0
      XMAX(4)=10.D0
      RETURN
      END
```

```
C
C
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(4)
C
      N=N
      NIC=NIC
C                 ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(3)-1.D0
      XXMAX(1)=10.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

ec3m_v1:

```
C
C
C
C              MAIN PROGRAM FOR PROBLEM "EC3M_V1.FOR"
C
C       EC - EQUALITY CONSTRAINED
C
C       K.SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NONLINEAR PROGRAMMING
C                        CODES", SPRINGER-VERLAG, 1987, LECTURE NOTES
C                        IN ECONOMICS AND MATHEMATICAL SYSTEMS, VOL.
C                        282, P.168 AND 220 (A BOOK).
C        THE PROBLEM HAS BEEN SLIGHTLY MODIFIED BY DR. S. N. GHANI FOR
C        COMPATIBILITY WITH SUBROUTINE "EVOP".
C
C
      DOUBLE PRECISION C(4),FF(8),H(32),OLDCC(4),XDN(4),XG(4),XMAX(4),
     1XMIN(4),XT(4),XUP(4),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
```

```
C
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=.70000001D0
      XT(2)=.2D0
      XT(3)=.1D0
      XT(4)=1.D0
C
C
C       CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=12000
      KNT=25
      N=4
      NIC=1
      K=8
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "EC3M_V1.FOR".'//)
  998 FORMAT(1X,'K. SCHITTKOWSKI: "MORE TEST EXAMPLES FOR NON',
     1'LINEAR PROGRAMMING'/1X,'CODES, SPRINGER-VERLAG, 1987,',
     2' LECTURE NOTES IN ECONOMICS AND'/1X,'MATHEMATICAL SYSTEMS,',
     3' VOL.282, PP.168 AND 220 (A BOOK).'/1X,'THE PROBLEM HAS',
     4' BEEN SLIGHTLY MODIFIED BY DR.S. N. GHANI FOR'/1X,'COMP',
     5'ATIBILITY WITH SUBROUTINE "EVOP".'//)
      END
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DLOG,F
      DOUBLE PRECISION XT(4)

C
      N=N
C                        ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=8204.37D0*DLOG((XT(1)+XT(2)+XT(3)+3.D-2)/(9.D-2*XT(1)+XT
     1 (2)+XT(3)+3.D-2))
      F=F+9008.72D0*DLOG((XT(2)+XT(3)+3.D-2)/(7.D-2*XT(2)+XT(3)+3.D-2))
      F=F+9330.46D0*DLOG((XT(3)+3.D-2)/(.13D0*XT(3)+3.D-2))
```

```
      F=F+XT(4)*(XT(1)+XT(2)+XT(3)-1.D0)
      RETURN
      END
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(4),XMIN(4),XT(4)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, 3
          XMAX(I)=1.D0
          XMIN(I)=0.D0
  109   CONTINUE
        XMAX(4)=10.D0
        XMIN(4)=0.D0
      ELSE
        DO 100 I=1, 3
          XMIN(I)=0.D0
          IF (XT(I) .GT. 0.D0) THEN
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.D0) THEN
              XMAX(I)=1.D0
            ENDIF
          ELSE
            XMAX(I)=1.D-1
          ENDIF
  100   CONTINUE
        XMIN(4)=0.D0
        IF (XT(4) .GT. 0.D0) THEN
          XMAX(4)=1.001D0*XT(4)
          IF (XMAX(4) .GT. 10.D0) THEN
            XMAX(4)=10.D0
          ENDIF
        ELSE
          XT(4)=1.D-1
        ENDIF
      ENDIF
      RETURN
      END
C
C
C
C
C
```

```
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(4)
C
      N=N
      NIC=NIC
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(3)-1.D0
      XXMAX(1)=10.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

MULTIPLE MINIMA

mm1_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM1_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA",
C            MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115),
C            PP.569-574.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2), XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C        STARTING POINT FOR OPTIMISATION
      XT(1)=4.3D0
```

```
      XT(2)=2.5D0
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT= 2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM1_V1.FOR".'//)
  998 FORMAT('A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT ',
     1'FROM LOCAL MINIMA",'/1X,'MATHEMATICS OF COMPUTATION,',
     2' 1971, VOL.25, (115), PP.569-574.'//)
      END
C
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION A,B,CF,DEXP,DSIN,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
C           FOR 'A' GREATER THAN '88'  DEXP(A) BELOW CRASHES FOR VAX
COMPUTERS
      IF (A .GT. 88.0D0) A=88.0D0
      B=XT(1)+XT(1)+XT(1)+XT(1)-XT(2)-XT(2)-XT(2)
      CF=XT(1)+XT(1)+XT(2)-10.D0
      CF=CF*CF
      F=DSIN(B)
      F=F*F*F*F
      F=F+DEXP(A)+.5D0*CF
      RETURN
      END
C
```

```
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
C              ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=-1.D0
      XMIN(2)=-1.D0
      XMAX(1)=5.D0
      XMAX(2)=5.D0
      RETURN
      END
C
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION A
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
      XX(1)=A
      XXMAX(1)=174.673D0
      XXMIN(1)=-180.2182D0
      RETURN
      END
```

mm1m_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM1M_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA",
C            MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115),
C            PP.569-574.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2), XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=4.3D0
      XT(2)=2.5D0
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM1M_V1.FOR".'//)
  998 FORMAT('A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT ',
     1'FROM LOCAL MINIMA",'/1X,'MATHEMATICS OF COMPUTATION,',
     2' 1971, VOL.25, (115), PP.569-574.'//)
      END
C
C
```

```
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION A,B,CF,DEXP,DSIN,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                       ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
C           FOR 'A' GREATER THAN '88'  DEXP(A) BELOW CRASHES FOR VAX
C                                                    COMPUTERS
      IF (A .GT. 88.0D0) A=88.0D0
      B=XT(1)+XT(1)+XT(1)+XT(1)-XT(2)-XT(2)-XT(2)
      CF=XT(1)+XT(1)+XT(2)-10.D0
      CF=CF*CF
      F=DSIN(B)
      F=F*F*F*F
      F=F+DEXP(A)+.5D0*CF
      RETURN
      END
C
C
C
C
C
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=5.D0
          XMIN(I)=-1.D0
  109   CONTINUE
      ELSE
        DO 120 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
```

```
          IF (XMIN(I) .LT. -1.D0) THEN
            XMIN(I)=-1.D0
          ENDIF
        ENDIF
        IF (XT(I) .GT. 0.D0) THEN
          XMIN(I)=0.D0
          XMAX(I)=1.001D0*XT(I)
          IF (XMAX(I) .GT. 5.D0) THEN
            XMAX(I)=5.D0
          ENDIF
        ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION A
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
      XX(1)=A
      XXMAX(1)=174.673D0
      XXMIN(1)=-180.2182D0
      RETURN
      END
```

mm2_v1:

```
C
C
C
C              MAIN PROGRAM FOR PROBLEM "MM2_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA",
C              MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115),
C              PP.569-574.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2), XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9, 999)
      WRITE(9, 998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
C          CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.7D0
      GAMA=1.5D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=10
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM2_V1.FOR".'//)
  998 FORMAT(1X,'A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT',
     1' FROM LOCAL MINIMA",'/1X,'MATHEMATICS OF COMPUTATION, ',
     2'1971, VOL.25, (115), PP. 569-574.'//)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
```

```
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F,FA
      DOUBLE PRECISION XT(2)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=18.D0-32.D0*XT(1)+12.D0*XT(1)*XT(1)+48.D0*XT(2)
      F=F-36.D0*XT(1)*XT(2)+27.D0*XT(2)*XT(2)
      F=F*(XT(1)+XT(1)-XT(2)-XT(2)-XT(2))
      F=F*(XT(1)+XT(1)-XT(2)-XT(2)-XT(2))+30.D0
      FA=-14.D0*XT(1)+3.D0*XT(1)*XT(1)-14.*XT(2)
      FA=FA+6.D0*XT(1)*XT(2)+3.D0*XT(2)*XT(2)+19.D0
      FA=FA*(XT(1)+XT(2)+1.D0)*(XT(1)+XT(2)+1.D0)+1.D0
C             VAX COMPUTERS CANNOT COPE WITH NUMBERS MUCH BIGGER THAN
C                                                  1.0E38
      IF (F .GT. 1.0D19) F=1.0D19
      IF (FA .GT. 1.0D19) FA=1.0D19
      F=F*FA
      RETURN
      END
C
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
C                   ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=-1.D5
      XMIN(2)=-1.D5
      XMAX(1)=1.D5
      XMAX(2)=1.D5
      RETURN
      END
C
C
C
C
C
C
C
C
C
```

```
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=1.D5
      XXMIN(1)=-1.D5
      RETURN
      END
```

mm2m_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM2M_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT FROM LOCAL MINIMA",
C               MATHEMATICS OF COMPUTATION, 1971, VOL.25, (115),
C               PP.569-574.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2), XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C

      WRITE(9, 999)
      WRITE(9, 998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
```

```
C          CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.7D0
      GAMA=1.5D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM2M_V1.FOR".'//)
  998 FORMAT(1X,'A. A. GOLDSTEIN AND J. F. PRICE: "ON DESCENT',
     1' FROM LOCAL MINIMA",'/1X,'MATHEMATICS OF COMPUTATION, ',
     2'1971, VOL.25, (115), PP. 569-574.'//)
      END
C
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F,FA
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=18.D0-32.D0*XT(1)+12.D0*XT(1)*XT(1)+48.D0*XT(2)
      F=F-36.D0*XT(1)*XT(2)+27.D0*XT(2)*XT(2)
      F=F*(XT(1)+XT(1)-XT(2)-XT(2)-XT(2))
      F=F*(XT(1)+XT(1)-XT(2)-XT(2)-XT(2))+30.D0
      FA=-14.D0*XT(1)+3.D0*XT(1)*XT(1)-14.*XT(2)
      FA=FA+6.D0*XT(1)*XT(2)+3.D0*XT(2)*XT(2)+19.D0
      FA=FA*(XT(1)+XT(2)+1.D0)*(XT(1)+XT(2)+1.D0)+1.D0
C            VAX COMPUTERS CANNOT COPE WITH NUMBERS MUCH BIGGER THAN
1.0E38
      IF (F .GT. 1.0D19) F=1.0D19
      IF (FA .GT. 1.0D19) FA=1.0D19
      F=F*FA
      RETURN
      END
C
C
```

```
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER I, IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
C                       ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 109 I=1, N
          XMAX(I)=1.D5
          XMIN(I)=-1.D5
  109   CONTINUE
      ELSE
        DO 100 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.D5) THEN
              XMIN(I)=-1.D5
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.D5) THEN
              XMAX(I)=1.D5
            ENDIF
          ENDIF
  100   CONTINUE
      ENDIF
      RETURN
      END
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=1.D5
      XXMIN(1)=-1.D5
      RETURN
      END
```

mm3_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM3_V1.FOR"
C
C
C  MM - MULTIPLE MINIMA
C
C  L. C. W. DIXON AND G. P. SZEGO (EDS): "TOWARDS GLOBAL MINIMISATION",
C           NORTH HOLLAND PUBLISHING CO., 1975 (TRECCANI).
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM3_V1.FOR".'//)
  998 FORMAT(1X,'L.C. W. DIXON AND G. P. SZEGO (EDS):',
     1' "TOWARDS GLOBAL MINIMISATION",'/1X,'NORTH HOL',
     2'LAND PUBLISHING CO., 1975  (TRECCANI).'//)
      END
C
C
C
```

```
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=XT(1)*XT(1)*XT(1)*XT(1)
      F=F+4.D0*XT(1)*XT(1)*XT(1)
      F=F+4.D0*XT(1)*XT(1)
      F=F+XT(2)*XT(2)+1.D0
      RETURN
      END
C
C
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=-1.D5
      XMIN(2)=-1.D5
      XMAX(1)=1.D5
      XMAX(2)=1.D5
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
```

```
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=1.D5
      XXMIN(1)=-1.D5
      RETURN
      END
```

mm3m_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM3M_V1.FOR"
C
C
C  MM - MULTIPLE MINIMA
C
C  L. C. W. DIXON AND G. P. SZEGO (EDS): "TOWARDS GLOBAL MINIMISATION",
C           NORTH HOLLAND PUBLISHING CO., 1975 (TRECCANI).
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
```

```
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM3_V1.FOR".'//)
  998 FORMAT(1X,'L.C. W. DIXON AND G. P. SZEGO (EDS):',
     1' "TOWARDS GLOBAL MINIMISATION",'/1X,'NORTH HOL',
     2'LAND PUBLISHING CO., 1975  (TRECCANI).'//)
      END
C
C
C                SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=XT(1)*XT(1)*XT(1)*XT(1)
      F=F+4.D0*XT(1)*XT(1)*XT(1)
      F=F+4.D0*XT(1)*XT(1)
      F=F+XT(2)*XT(2)+1.D0
      RETURN
      END
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMAX(1)=1.D5
        XMAX(2)=1.D5
        XMIN(1)=-1.D5
        XMIN(2)=-1.D5
      ELSE
        DO 120 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.D5) THEN
```

```
              XMIN(I)=-1.D5
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.D5) THEN
              XMAX(I)=1.D5
            ENDIF
          ENDIF
  120   CONTINUE
      ENDIF
      RETURN
      END
C
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=1.D5
      XXMIN(1)=-1.D5
      RETURN
      END
```

mm4_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM4_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  HE - HESSE [HESSE 1973], (N=6), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(6),FF(7),H(42),OLDCC(6),XDN(6),XG(6),
     1XMAX(6),XMIN(6),XT(6),XUP(6),XX(5),XXMAX(5),XXMIN(5)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=1.875D0
      XT(2)=0.375D0
      XT(3)=5.D0
      XT(4)=0.95D0
      XT(5)=5.D0
      XT(6)=8.D0
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-8
      PHICPX=1.D-6
      ICON=5
      LIMIT=60000
      KNT=25
      N=6
      NIC=5
      K=7
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM4_V1.FOR".'//)
```

```
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, HE - HESSE, VOLUME 350, SPRINGER VERLAG, 1987.'///)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(6)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=25.D0*(XT(1)-2.D0)*(XT(1)-2.D0)+(XT(2)-2.D0)*(XT(2)-2.D0)
      F=F+(XT(3)-1.D0)*(XT(3)-1.D0)+(XT(4)-4.D0)*(XT(4)-4.D0)
      F=F+(XT(5)-1.D0)*(XT(5)-1.D0)+(XT(6)-4.D0)*(XT(6)-4.D0)
      F=-F
      RETURN
      END
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(6),XMIN(6),XT(6)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
      M=M
      XT(1)=XT(1)
C                         ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=0.D0
      XMIN(2)=0.D0
      XMIN(3)=1.D0
      XMIN(4)=0.D0
      XMIN(5)=1.D0
      XMIN(6)=0.D0
      XMAX(1)=1000.D0
      XMAX(2)=1000.D0
      XMAX(3)=5.D0
      XMAX(4)=6.D0
      XMAX(5)=5.D0
      XMAX(6)=10.D0
      RETURN
      END
C
C
```

```
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(5),XXMAX(5),XXMIN(5),XT(6)
C
      N=N
      NIC=NIC
C                 ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XX(2)=-XT(1)+XT(2)
      XX(3)=XT(1)-XT(2)-XT(2)-XT(2)
      XX(4)=(XT(3)-3.D0)*(XT(3)-3.D0)+XT(4)
      XX(5)=(XT(5)-3.D0)*(XT(5)-3.D0)+XT(6)
      XXMIN(1)=2.D0
      XXMIN(2)=-1000.D0
      XXMIN(3)=-1000.D0
      XXMIN(4)=4.D0
      XXMIN(5)=4.D0
      XXMAX(1)=6.D0
      XXMAX(2)=2.D0
      XXMAX(3)=2.D0
      XXMAX(4)=1000.D0
      XXMAX(5)=1000.D0
      RETURN
      END
```

mm4m_v1:

```
C
C
C
C               MAIN PROGRAM FOR PROBLEM "MM4M_V1.FOR"
C
C  MM - MULTIPLE MINIMA WITH MOVING BOUNDARY.
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  HE - HESSE [HESSE 1973], (N=6), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C  MOVING BOUNDARY HAS BEEN INTRODUCED BY DR. S. N. GHANI.
C
C
C
      DOUBLE PRECISION C(6),FF(7),H(42),OLDCC(6),XDN(6),XG(6),
     1XMAX(6),XMIN(6),XT(6),XUP(6),XX(5),XXMAX(5),XXMIN(5)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=1.875D0
      XT(2)=0.375D0
      XT(3)=5.D0
      XT(4)=0.95D0
      XT(5)=5.D0
      XT(6)=8.D0
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=0.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-8
      PHICPX=1.D-7
      ICON=5
      LIMIT=60000
      KNT=25
      N=6
      NIC=5
      K=7
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
```

```
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM4M_V1.FOR."'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, HE - HESSE, VOLUME 350, SPRINGER VERLAG, 1987.'///)
      END
C
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(6)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=25.D0*(XT(1)-2.D0)*(XT(1)-2.D0)+(XT(2)-2.D0)*(XT(2)-2.D0)
      F=F+(XT(3)-1.D0)*(XT(3)-1.D0)+(XT(4)-4.D0)*(XT(4)-4.D0)
      F=F+(XT(5)-1.D0)*(XT(5)-1.D0)+(XT(6)-4.D0)*(XT(6)-4.D0)
      F=-F
      RETURN
      END
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(6),XMIN(6),XT(6)
C
      IFLG=IFLG
      ISKP=ISKP
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=0.D0
        XMIN(2)=0.D0
        XMIN(3)=1.D0
        XMIN(4)=0.D0
        XMIN(5)=1.D0
        XMIN(6)=0.D0
        XMAX(1)=1000.D0
        XMAX(2)=1000.D0
        XMAX(3)=5.D0
        XMAX(4)=6.D0
        XMAX(5)=5.D0
        XMAX(6)=10.D0
      ELSE
        XMIN(1)=0.D0
C       IF (XT(1) .GT. 0.D0) THEN
```

```
C           XMAX(1)=1.001D0*XT(1)
C           IF (XMAX(1) .GT. 1000.D0) THEN
              XMAX(1)=1000.D0
C           ENDIF
C         ELSE
C           XMAX(1)=1.D-1
C         ENDIF
          XMIN(2)=0.D0
C         IF (XT(2) .GT. 0.D0) THEN
C           XMAX(2)=1.001D0*XT(2)
C           IF (XMAX(2) .GT. 1000.D0) THEN
              XMAX(2)=1000.D0
C           ENDIF
C         ELSE
C           XMAX(2)=1.D-1
C         ENDIF
          XMIN(3)=1.D0
C         IF (XMAX(3) .GT. 1.D0) THEN
C           XMAX(3)=1.001D0*XT(3)
C           IF (XMAX(3) .GT. 5.D0) THEN
              XMAX(3)=5.D0
C           ENDIF
C         ELSE
C           XMAX(3)=1.1D0
C         ENDIF
          XMIN(4)=0.D0
C         IF (XT(4) .GT. 0.D0) THEN
C           XMAX(4)=1.001D0*XT(4)
C           IF (XMAX(4) .GT. 6.D0) THEN
              XMAX(4)=6.D0
C           ENDIF
C         ELSE
C           XMAX(4)=1.D-1
C         ENDIF
          XMIN(5)=1.D0
          IF (XT(5) .GT. 1.D0) THEN
            XMAX(5)=1.001D0*XT(5)
            IF (XMAX(5) .GT. 5.D0) THEN
              XMAX(5)=5.D0
            ENDIF
          ELSE
            XMAX(5)=1.1D0
          ENDIF
          XMIN(6)=0.D0
          IF (XT(6) .GT. 0.D0) THEN
            XMAX(6)=1.001D0*XT(6)
            IF (XMAX(6) .GT. 10.D0) THEN
              XMAX(6)=10.D0
            ENDIF
          ELSE
            XMAX(6)=1.D-1
          ENDIF
        ENDIF
        RETURN
        END
C
C
```

```
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(5),XXMAX(5),XXMIN(5),XT(6)
C
      N=N
      NIC=NIC
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XX(2)=-XT(1)+XT(2)
      XX(3)=XT(1)-XT(2)-XT(2)-XT(2)
      XX(4)=(XT(3)-3.D0)*(XT(3)-3.D0)+XT(4)
      XX(5)=(XT(5)-3.D0)*(XT(5)-3.D0)+XT(6)
      XXMIN(1)=2.D0
      XXMIN(2)=-1000.D0
      XXMIN(3)=-1000.D0
      XXMIN(4)=4.D0
      XXMIN(5)=4.D0
      XXMAX(1)=6.D0
      XXMAX(2)=2.D0
      XXMAX(3)=2.D0
      XXMAX(4)=1000.D0
      XXMAX(5)=1000.D0
      RETURN
      END
```

mm5_v1:

```
C
C
C              MAIN PROGRAM FOR PROBLEM "MM5_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=2), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),
     1XMAX(2),XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      N=2
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-12
      ICON=5
      LIMIT=6000
      KNT=25
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM5_V1.FOR."'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
```

```
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(2)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/2.D2
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      M=M
      IFLG=IFLG
      ISKP=ISKP
      XT(1)=XT(2)
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 10 I=1,N
        XMIN(I)=-1.D2
        XMAX(I)=1.D2
   10 CONTINUE
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
```

```
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMIN(1)=-2.D2
      XXMAX(1)=2.D2
      RETURN
      END
```

mm5m_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "MM5M_V1.FOR"
C
C  MM - MULTIPLE MINIMA AND MOVING BOUNDARIES.
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=2), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),
     1XMAX(2),XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      N=2
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
```

```
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM5M_V1.FOR."'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(2)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/2.D2
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      IFLG=IFLG
      ISKP=ISKP
C                         ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
```

```
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 100 I=1,N
          XMIN(I)=-1.D2
          XMAX(I)=1.D2
  100   CONTINUE
      ELSE
        DO 110 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.D2) THEN
                XMIN(I)=-1.D2
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.D2) THEN
              XMAX(I)=1.D2
             ENDIF
          ENDIF
  110   CONTINUE
      ENDIF
      RETURN
      END
C
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 10 I=1,N
        XX(1)=XX(1)+XT(I)
   10 CONTINUE
      XXMIN(1)=-2.D2
      XXMAX(1)=2.D2
      RETURN
      END
```

mm6_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "MM6_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(10),FF(11),H(110),OLDCC(10),XDN(10),XG(10),
     1XMAX(10),XMIN(10),XT(10),XUP(10),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C       STARTING POINT FOR OPTIMISATION
      N=10
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=60000
      KNT=25
      NIC=1
      K=11
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM6_V1.FOR".'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
```

```
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(10)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/2.D2
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(10),XMIN(10),XT(10)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      XT(1)=XT(1)
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 10 I=1,N
        XMIN(I)=-1.D2
        XMAX(I)=1.D2
   10 CONTINUE
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
```

```
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(10)
C
      NIC=NIC
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 10 I=1,N
        XX(1)=XX(1)+XT(I)
   10 CONTINUE
      XXMIN(1)=-1.D3
      XXMAX(1)=1.D3
      RETURN
      END
```

mm6m_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "MM6M_V1.FOR"
C
C  MM - MULTIPLE MINIMA AND MOVING BOUNDARIES.
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(10),FF(11),H(110),OLDCC(10),XDN(10),XG(10),
     1XMAX(10),XMIN(10),XT(10),XUP(10),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C        STARTING POINT FOR OPTIMISATION
      N=10
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
```

```
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-16
      ICON=5
      LIMIT=60000
      KNT=25
      NIC=1
      K=11
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM6M_V1.FOR".'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(10)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/2.D2
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
```

```
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(10),XMIN(10),XT(10)
C
      IFLG=IFLG
      ISKP=ISKP
C                      ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 100 I=1,N
          XMIN(I)=-1.D2
          XMAX(I)=1.D2
  100   CONTINUE
      ELSE
        DO 110 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.001D0*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -1.D2) THEN
                XMIN(I)=-1.D2
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.001D0*XT(I)
            IF (XMAX(I) .GT. 1.D2) THEN
              XMAX(I)=1.D2
             ENDIF
          ENDIF
  110   CONTINUE
      ENDIF
      RETURN
      END
C
C
C              SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(10)
C
      NIC=NIC
C                         ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 10 I=1,N
        XX(1)=XX(1)+XT(I)
   10 CONTINUE
      XXMIN(1)=-1.D3
      XXMAX(1)=1.D3
      RETURN
      END
C
C
```

```
C
C
C
C
C
C
C
C
```

mm7_v1:

```
C
C
C              MAIN PROGRAM FOR PROBLEM "MM7_V1.FOR"
C
C  MM - MULTIPLE MINIMA
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(10),FF(11),H(110),OLDCC(10),XDN(10),XG(10),
     1XMAX(10),XMIN(10),XT(10),XUP(10),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      N=10
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-12
      ICON=5
      LIMIT=60000
      KNT=25
      NIC=1
      K=11
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
```

```
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM7_V1.FOR".'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
C
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(10)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/4.D3
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(10),XMIN(10),XT(10)
C
      IFLG=IFLG
      ISKP=ISKP
      M=M
      XT(1)=XT(1)
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      DO 10 I=1,N
        XMIN(I)=-6.D2
        XMAX(I)=6.D2
   10 CONTINUE
      RETURN
      END
C
C
C
```

```
C
C
C
C
C
C
C
C
C    SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(10)
C
      NIC=NIC
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 10 I=1,N
        XX(1)=XX(1)+XT(I)
   10 CONTINUE
      XXMIN(1)=-6.D3
      XXMAX(1)=6.D3
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
C
```

mm7m_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "MM7M_V1.FOR"
C
C  MM - MULTIPLE MINIMA AND MOVING BOUNDARIES.
C
C  AIMO TORN AND ANTANAS ZILINSKAS: `GLOBAL OPTIMIZATION', PAGE 186,
C  Gn - GRIEWANK [GRIEWANK 1981], (N=10), VOLUME 350, SPRINGER-VERLAG,
C  1987, ISBN 3-540-50871-6 AND ISBN 0-387-50871-6.
C
C
C
      DOUBLE PRECISION C(10),FF(11),H(110),OLDCC(10),XDN(10),XG(10),
     1XMAX(10),XMIN(10),XT(10),XUP(10),XX(1),XXMAX(1),XXMIN(1)
C
```

```
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C
C      STARTING POINT FOR OPTIMISATION
      N=10
      DO 10 I=1,N
        XT(I)=10.D0
   10 CONTINUE
C
C
C    CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.1D0
      BETA=.6D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-9
      PHICPX=1.D-10
      ICON=5
      LIMIT=60000
      KNT=25
      NIC=1
      K=11
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MM7M_V1.FOR".'//)
  998 FORMAT(1X,'A. TORN AND A. ZILINSKAS: "GLOBAL OPTIMIZATION"',/,
     1' PAGE 186, Gn - GRIEWANK, VOLUME 350, SPRINGER VERLAG,
1987.'///)
      END
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DCOS,DFLOAT,DSQRT,F,F2
      DOUBLE PRECISION XT(10)
C
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=0.D0
      DO 10 I=1,N
        F=F+XT(I)*XT(I)
   10 CONTINUE
      F=F/4.D3
```

```
      F2=1.D0
      DO 20 I=1,N
        F2=F2*DCOS(XT(I)/DSQRT(DFLOAT(I)))
   20 CONTINUE
      F=F-F2+2.D0
      RETURN
      END
C
C
C
C
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION XMAX(10),XMIN(10),XT(10)
C
      IFLG=IFLG
      ISKP=ISKP
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        DO 100 I=1,N
          XMIN(I)=-6.D2
          XMAX(I)=6.D2
  100   CONTINUE
      ELSE
        DO 110 I=1, N
          IF (XT(I) .LT. 0.D0) THEN
            XMIN(I)=1.D4*XT(I)
            XMAX(I)=0.D0
            IF (XMIN(I) .LT. -6.D2) THEN
                XMIN(I)=-6.D2
            ENDIF
          ENDIF
          IF (XT(I) .GT. 0.D0) THEN
            XMIN(I)=0.D0
            XMAX(I)=1.D4*XT(I)
            IF (XMAX(I) .GT. 6.D2) THEN
              XMAX(I)=6.D2
             ENDIF
          ENDIF
  110   CONTINUE
      ENDIF
      RETURN
      END
C
C
C
C
```

```
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(10)
C
      NIC=NIC
C                         ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=0.D0
      DO 10 I=1,N
        XX(1)=XX(1)+XT(I)
   10 CONTINUE
      XXMIN(1)=-6.D3
      XXMAX(1)=6.D3
      RETURN
      END



C
C
C
C
C
```

MIXED VARIABLES

```
C
C
C
```

mv1_v1:

```
C
C
C
C                MAIN PROGRAM FOR PROBLEM "MV1_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "UC1.V1" BUT VARIABLE "XT(1)" ALLOWED TO TAKE
C       VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF
C       INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN
C       PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
C
      WRITE(9,999)
      WRITE(9,998)
```

```
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=60000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MV6_V1.FOR"'//)
  998 FORMAT(1X,'SAME AS "MV1_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED',
     1' TO TAKE ONLY'/1X,'NEAR INTEGER VALUES. MODIFICATIONS INT',
     2'RODUCED BY DR. S. N. GHANI.'//)
      END
C
C
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=100.D0*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
      F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
      RETURN
      END
C
C
C
C
C
C
```

```
C
C
C
C

C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      M=M
      N=N
C              ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=-1000.001D0
      XMIN(2)=-1000.D0
      XMAX(1)=1000.001D0
      XMAX(2)=1000.D0
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
      ENDIF
      RETURN
      END
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)
      XXMAX(1)=1.D5
      XXMIN(1)=-1.D5
      RETURN
      END

C
C
C
C
C
C
C
C
```

mv1m_v1:

```
C
C
C
C                MAIN PROGRAM FOR PROBLEM "MV1M_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "MV1_V1.FOR" BUT MOVING EXPLICIT CONSTRAINTS
C       TO EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY DR. S. N.
C       GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
C
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=-1.2D0
      XT(2)=1.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MV6M_V1.FOR"'//)
  998 FORMAT(1X,'SAME AS "MV1_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED',
     1' TO TAKE ONLY'/1X,'NEAR INTEGER VALUES. MODIFICATIONS INT',
     2'RODUCED BY DR. S. N. GHANI.'//)
      END
C
C
```

```
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(2)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=100.D0*((XT(1)*XT(1)-XT(2))*(XT(1)*XT(1)-XT(2)))
      F=F+(1.D0-XT(1))*(1.D0-XT(1))+1.D0
      RETURN
      END
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=-1000.001D0
        XMIN(2)=-1000.D0
        XMAX(1)=1000.001D0
        XMAX(2)=1000.D0
      ELSE
        XMIN(1)=-1000.001D0
        XMAX(1)=1000.001D0
        IF (XT(2) .LT. 0.D0) THEN
          XMAX(2)=0.D0
          XMIN(2)=1.001D0*XT(2)
          IF (XMIN(2) .LT. -1000.D0) THEN
            XMIN(2)=-1000.D0
          ENDIF
        ENDIF
        IF (XT(2) .EQ. 0.D0) THEN
           XMIN(2)=-0.01D0
           XMAX(2)=0.01D0
        ENDIF
        IF (XT(2) .GT. 0.D0) THEN
          XMIN(2)=0.D0
          XMAX(2)=1.001D0*XT(2)
          IF (XMAX(2) .GT. 1000.D0) THEN
             XMAX(2)=1000.D0
          ENDIF
```

```
          ENDIF
        ENDIF
        IF (IFLG .EQ. 0) THEN
          STRIP=1.D-4
          CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
        ENDIF
        RETURN
        END
C
C
C                SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
        SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
        INTEGER KOUNT,M,N,NIC
        DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
        N=N
        NIC=NIC
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
        KOUNT=KOUNT+1
        M=M+1
        XX(1)=XT(1)+XT(2)
        XXMAX(1)=1.D5
        XXMIN(1)=-1.D5
        RETURN
        END
```

mv2_v1:

```
C
C

C               MAIN PROGRAM FOR PROBLEM "MV2_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C
C       SAME AS PROBLEM "C2.V1" BUT VARIABLES "XT(1)" AND "XT(2)"
C       ALLOWED TO TAKE VALUES FROM A USER DEFINED NARROW STRIP AROUND
C       THE ROUNDED OFF INTEGER VALUE AND PREASSIGNED DISCRETE
C       VALUE RESPECTIVELY. SET CONTROL PARAMETER "IMV" TO 0 IN THE
C       MAIN PROGRAM. MODIFICATIONS INTRODUCED BY DR S. N. GHANI.

C
      DOUBLE PRECISION C(3),FF(6),H(18),OLDCC(3),XDN(3),XG(3),XMAX(3),
     1XMIN(3),XT(3),XUP(3),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
```

```
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
C
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=10.D0
      XT(2)=10.D0
      XT(3)=10.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-9
      ICON=5
      LIMIT=60000
      KNT=25
      N=3
      NIC=1
      K=6
      IPRINT=2
      IJK=1
      NRSTRT=2
      IMV=0
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM:  MV7_V1.FOR'//)
  998 FORMAT('SAME AS "C2_V1.FOR" BUT VARIABLES "XT(1)" AND',
     1' "XT(2)" ALLOWED TO TAKE ONLY INTEGER AND PREASSIGNED',
     2' DISCRETE VARIABLES RESPECTIVELY.'/1X,'MODIFICATIONS',
     3' INTRODUCED BY DR. S. N. GHANI.'//)
      END
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(3)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)*XT(2)*XT(3))
      RETURN
      END
C
```

```
C
C
C            SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION DX(10),XMAX(3),XMIN(3),XT(3)
      DOUBLE PRECISION STRIP
C
      N=N
      M=M
C                  ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=.0000001D0
      XMIN(2) =.00000015D0
      XMIN(3) =0.D0
      XMAX(1) =20.01D0
      XMAX(2) =11.01D0
      XMAX(3) =42.D0
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
        DX(1)=.1D0
        DX(2)=.8D0
        DX(3)=1.5D0
        DX(4)=1.9D0
        DX(5)=3.5D0
        DX(6)=6.2D0
        DX(7)=8.1D0
        DX(8)=9.2D0
        DX(9)=10.5D0
        DX(10)=11.D0
        CALL DISCR2(DX,ISKP,10,STRIP,XT(2),XMAX(2),XMIN(2))
      ENDIF
      RETURN
      END
C
C
C
C
C            SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(3)
C
      N=N
      NIC=NIC
C                 ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(2)+XT(3)+XT(3)
      XXMAX(1)=72.D0
      XXMIN(1)=0.D0
```

```
      RETURN
      END
```

mv2m_v1:

```
C
C
C
C              MAIN PROGRAM FOR PROBLEM "MV7M_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "MV2_V1.FOR" BUT MOVING EXPLICIT CONSTRAINTS
C       TO EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY DR. S. N.
C       GHANI.
C
      DOUBLE PRECISION C(3),FF(6),H(18),OLDCC(3),XDN(3),XG(3),XMAX(3),
     1XMIN(3),XT(3),XUP(3),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
C
      WRITE(9,999)
      WRITE(9,998)
C
C      STARTING POINT FOR OPTIMISATION
      XT(1)=10.D0
      XT(2)=10.D0
      XT(3)=10.D0
C
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.3D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-8
      ICON=5
      LIMIT=60000
      KNT=25
      N=3
      NIC=1
      K=6
      IPRINT=2
      IJK=1
      NRSTRT=2
      IMV=0
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
```

```
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MV7M_V1.FOR"'//)
  998 FORMAT(1X,'SAME AS "MV2_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED',
     1' TO TAKE ONLY'/1X,'NEAR INTEGER VALUES. MODIFICATIONS INT',
     2'RODUCED BY DR. S. N. GHANI.'//)
      END
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION F
      DOUBLE PRECISION XT(3)
C
      N=N
C                        ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)*XT(2)*XT(3))
      RETURN
      END
C
C             SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION DX(10),XMAX(3),XMIN(3),XT(3)
      DOUBLE PRECISION STRIP
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=.0000001D0
        XMIN(2) =.00000015D0
        XMIN(3) =0.D0
        XMAX(1) =20.01D0
        XMAX(2) =11.01D0
        XMAX(3) =42.D0
      ELSE
        XMIN(1)=0.0000001D0
        XMIN(2)=0.00000015D0
        XMAX(1)=20.01D0
        XMAX(2)=11.01D0
        XMIN(3)=0.D0
        IF (XT(3) .GT. 0.D0) THEN
          XMAX(3)=1.001D0*XT(3)
          IF (XMAX(3) .GT. 42.D0) THEN
            XMAX(3)=42.D0
          ENDIF
        ELSE
```

```
          XMAX(3)=1.D-1
        ENDIF
      ENDIF
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
        DX(1)=.1D0
        DX(2)=.8D0
        DX(3)=1.5D0
        DX(4)=1.9D0
        DX(5)=3.5D0
        DX(6)=6.2D0
        DX(7)=8.1D0
        DX(8)=9.2D0
        DX(9)=10.5D0
        DX(10)=11.D0
        CALL DISCR2(DX,ISKP,10,STRIP,XT(2),XMAX(2),XMIN(2))
      ENDIF
      RETURN
      END
C
C
C
C
C
C            SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(3)
C
      N=N
      NIC=NIC
C                       ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)+XT(2)+XT(2)+XT(3)+XT(3)
      XXMAX(1)=72.D0
      XXMIN(1)=0.D0
      RETURN
      END
```

mv3_v1:

```
C
C
C                MAIN PROGRAM FOR PROBLEM "MV3_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "MM1.V1" BUT VARIABLE "XT(1)" ALLOWED TO TAKE
C       VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF
C       INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN
C       PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.
C
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C        STARTING POINT FOR OPTIMISATION
      XT(1)=4.3D0
      XT(2)=2.5D0
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-10
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: MV8_V1.FOR.'//)
  998 FORMAT(5X,'SAME AS "MM3_V1.FOR" BUT VARIABLE "XT(1)"',
     1' ALLOWED TO TAKE ONLY',/1X,'INTEGER VALUES.'//)
      END
C
C
C
C
```

```
C
C               SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION A,B,CF,DEXP,DSIN,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                       ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
C           FOR 'A' GREATER THAN '88'  DEXP(A) BELOW CRASHES FOR VAX
C COMPUTERS
      IF (A .GT. 88.0D0) A=88.0D0
      B=XT(1)+XT(1)+XT(1)+XT(1)-XT(2)-XT(2)-XT(2)
      CF=XT(1)+XT(1)+XT(2)-10.D0
      CF=CF*CF
      F=DSIN(B)
      F=F*F*F*F
      F=F+DEXP(A)+.5D0*CF
      RETURN
      END
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      N=N
      M=M
C                        ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=.000001D0
      XMIN(2)=0.D0
      XMAX(1)=5.000001D0
      XMAX(2)=5.D0
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
      ENDIF
      RETURN
      END
C
C
C
C
```

```
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION A
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
      XX(1)=A
      XXMAX(1)=174.673D0
      XXMIN(1)=-180.2182D0
      RETURN
      END
```

mv3m_v1:

```
C
C
C                MAIN PROGRAM FOR PROBLEM "MV3M_V1.FOR"
C
C  MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "MV3_V1.FOR" BUT MOVING EXPLICIT
C       CONSTRAINTS TO EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED
C       BY DR. S. N. GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C         STARTING POINT FOR OPTIMISATION
      XT(1)=4.3D0
      XT(2)=2.5D0
```

```
C
C     CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.5D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-9
      PHICPX=1.D-9
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(1X,'OPTIMIZATION OF TEST PROBLEM: MV8M_V1.FOR.'//)
  998 FORMAT(5X,'SAME AS "MV3M_V1.FOR" BUT VARIABLE "XT(1)" ',
     1'ALLOWED TO TAKE ONLY',/1X,'INTEGER VALUES.'//)
      END
C
C
C
C
C
C              SUBROUTINE FOR FUNCTION VALUE
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION A,B,CF,DEXP,DSIN,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
      A=.25D0*A*A
      B=XT(1)+XT(1)+XT(1)+XT(1)-XT(2)-XT(2)-XT(2)
      CF=XT(1)+XT(1)+XT(2)-10.D0
      CF=CF*CF
      F=DSIN(B)
      F=F*F*F*F
      F=F+DEXP(A)+.5D0*CF
      RETURN
      END
C
C
C
C
```

```
C
C
C
C
C                SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      N=N
C                 ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=.000001D0
        XMIN(2)=0.D0
        XMAX(1)=5.000001D0
        XMAX(2)=5.D0
      ELSE
        XMIN(1)=.000001D0
        XMIN(2)=0.D0
        XMAX(1)=5.000001D0
        IF (XT(2) .GT. 0.D0) THEN
          XMAX(2)=1.001D0*XT(2)
          IF (XMAX(2) .GT. 5.D0) THEN
            XMAX(2)=5.D0
          ENDIF
        ELSE
          XMAX(2)=1.D-1
        ENDIF
      ENDIF
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
      ENDIF
      RETURN
      END
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION A
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      A=XT(1)*XT(1)+XT(2)*XT(2)-25.D0
```

```
      A=.25D0*A*A
      XX(1)=A
      XXMAX(1)=174.673D0
      XXMIN(1)=-180.2182D0
      RETURN
      END
```

mv4_v1:

```
C
C
C                MAIN PROGRAM FOR PROBLEM "MV4_V1.FOR"
C
C   MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "C1.V1" BUT VARIABLE "XT(1)" ALLOWED TO TAKE
C       VALUES FROM A USER DEFINED NARROW STRIP AROUND THE ROUNDED OFF
C       INTEGER VALUE. SET CONTROL PARAMETER "IMV" TO 0 IN THE MAIN
C       PROGRAM. MODIFICATIONS INTRODUCED BY DR. S. N. GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=1.D0
      XT(2)=1.D0
C
C
C       CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
```

```
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MV9_V1.FOR".'//)
  998 FORMAT(1X,'SAME AS "MV4_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED',
     1' TO TAKE ONLY'/1X,'INTEGER VALUES. MODIFICATIONS INTRODUCED',
     2' BY DR. S. N. GHANI.'//)
      END
C
C                SUBROUTINE FOR FUNCTION EVALUATION
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DEXP,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                     ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)-1.D0)*(XT(1)-1.D0)
      F=F-(XT(2)*XT(2)-.5D0)*(XT(2)*XT(2)-.5D0)/.132D0
      F=-(DEXP(F))
      RETURN
      END
C
C
C
C               SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
C
      M=M
      N=N
C                        ABOVE TO STOP THE COPMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      XMIN(1)=.2D0
      XMIN(2)=.2D0
      XMAX(1)=2.00001D0
      XMAX(2)=2.D0
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
      ENDIF
      RETURN
      END
C
```

```
C
C
C
C
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                        ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)*XT(1)+XT(2)*XT(2)
      XXMAX(1)=4.D0
      XXMIN(1)=-99999.D0
      RETURN
      END

C
C
C
C
```

mv4m_v1:

```
C
C
C               MAIN PROGRAM FOR PROBLEM "MV4M_V1.FOR"
C
C   MV - MIXED VARIABLE TYPES
C
C       SAME AS PROBLEM "MV4_V1.FOR" BUT MOVING EXPLICIT CONSTRAINTS
C       TO EXPEDITE CONVERGENCE. MODIFICATIONS INTRODUCED BY DR. S. N.
C       GHANI.
C
      DOUBLE PRECISION C(2),FF(4),H(8),OLDCC(2),XDN(2),XG(2),XMAX(2),
     1XMIN(2),XT(2),XUP(2),XX(1),XXMAX(1),XXMIN(1)
C
      DOUBLE PRECISION ALPHA,BETA,DEL,GAMA,PHI,PHICPX
C
      INTEGER ICON,IJK,IMV,IPRINT,K,KNT,LIMIT,N,NRSTRT,NIC
C
      WRITE(9,999)
      WRITE(9,998)
C
C       STARTING POINT FOR OPTIMISATION
      XT(1)=1.D0
      XT(2)=1.D0
C
C
C       CONTROL PARAMETERS FOR "EVOP"
      ALPHA=1.2D0
```

```
      BETA=.4D0
      GAMA=2.D0
      DEL=1.D-12
      PHI=1.D-10
      PHICPX=1.D-11
      ICON=5
      LIMIT=6000
      KNT=25
      N=2
      NIC=1
      K=4
      IPRINT=2
      NRSTRT=2
      IMV=0
      IJK=1
  100 CALL EVOP(ALPHA,BETA,C,DEL,FF,GAMA,H,ICON,IJK,IMV,IPRINT,K,
     1KNT,LIMIT,N,NRSTRT,NIC,OLDCC,PHI,PHICPX,XDN,XG,XMAX,XMIN,XT,XUP,
     2XX,XXMAX,XXMIN)
      IF (IJK .LT. 9) GOTO 100
  999 FORMAT(5X,'OPTIMIZATION OF TEST PROBLEM: "MV9M_V1.FOR".'//)
  998 FORMAT(1X,'SAME AS "MV4M_V1.FOR" BUT VARIABLE "XT(1)" ALLOWED',
     1' TO TAKE ONLY'/1X,'INTEGER VALUES. MODIFICATIONS INTRODUCED',
     2' BY DR. S. N. GHANI.'//)

      END
C
C
C               SUBROUTINE FOR FUNCTION EVALUATION
C
      SUBROUTINE FUNC(F,KOUNT,KUT,N,XT)
C
      INTEGER KOUNT,KUT,N
      DOUBLE PRECISION DEXP,F
      DOUBLE PRECISION XT(2)
C
      N=N
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KUT=KUT+1
      F=-(XT(1)-1.D0)*(XT(1)-1.D0)
      F=F-(XT(2)*XT(2)-.5D0)*(XT(2)*XT(2)-.5D0)/.132D0
      F=-(DEXP(F))
      RETURN
      END
C
C
C
C
C
C
C              SUBROUTINE FOR EXPLICIT CONSTRAINTS
C
      SUBROUTINE EXPCON(IFLG,ISKP,KKT,KOUNT,M,N,XMAX,XMIN,XT)
C
      INTEGER IFLG,ISKP,KKT,KOUNT,M,N,NFUNC
      DOUBLE PRECISION STRIP
      DOUBLE PRECISION XMAX(2),XMIN(2),XT(2)
```

```
C
      N=N
C                          ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      KKT=KKT+1
      NFUNC=KOUNT-KKT-M
      IF (NFUNC .LT. 2) THEN
        XMIN(1)=.2D0
        XMIN(2)=.2D0
        XMAX(1)=2.00001D0
        XMAX(2)=2.D0
      ELSE
        XMIN(1)=.2D0
        XMIN(2)=.2D0
        XMAX(1)=2.00001D0
        IF (XT(2) .GT. .2D0) THEN
          XMAX(2)=1.001D0*XT(2)
          IF (XMAX(2) .GT. 2.D0) THEN
            XMAX(2)=2.D0
          ENDIF
        ELSE
          XMAX(2)=.201D0
        ENDIF
      ENDIF
      IF (IFLG .EQ. 0) THEN
        STRIP=1.D-4
        CALL DINTG2(ISKP,STRIP,XT(1),XMAX(1),XMIN(1))
      ENDIF
      RETURN
      ENDC
C
C
C               SUBROUTINE FOR IMPLICIT CONSTRAINTS
C
      SUBROUTINE IMPCON(KOUNT,M,N,NIC,XT,XX,XXMAX,XXMIN)
C
      INTEGER KOUNT,M,N,NIC
      DOUBLE PRECISION XX(1),XXMAX(1),XXMIN(1),XT(2)
C
      N=N
      NIC=NIC
C                    ABOVE TO STOP THE COMPILER FROM COMPLAINING
      KOUNT=KOUNT+1
      M=M+1
      XX(1)=XT(1)*XT(1)+XT(2)*XT(2)
      XXMAX(1)=4.D0
      XXMIN(1)=-99999.D0
      RETURN
      END
```