AN N-DIMENSIONAL FUNCTION - ONLY CODE FOR NON-LINEAR UNCONSTRAINED OPTIMIZATION

C. Sutti (*) e *R. Voltini* (**)

1. *Introduction.* -

The present report documents a code, compiled in the two versions OTLSSS and OTLSSD, for minimizing n-dimensional functions.

This routine is to be inserted in a library which will be provided from the CNR, SOFMAT Project, to solve a wide range of mathematical and statistical problems arising in a variety of fields such as applied mathematics, physics, chemistry, engineering, biology, economics, manageral science, market research, governement, agricultural and medical research.

Such library will be available in FORTRAN language for minicomputers, namely for PDP 11/40. It will cater for both the novice and the experienced programmer, therefore the documentation of all routines must be comprehensive, detailed anc clear. Moreover the selection and the implementation of the algorithms and the choice of the test problems must reflect the aim of the library which tends to possess efficiency, usefulness, accuracy and reliability.

2. *Routine document.* -

The two codes OTLSSS and OTLSSD, written in FORTRAN language for the PDP 11/40 computer, are two versions of the same program respectively compiled in single and in double precision. This program has been developed to solve the problem of non-linear uncostrained optimisation having the following mathematical description

$$\min_{x \in R^n} F(x)$$

(*) Mathematical Institute - University - LECCE - ITALY

(**)Mathematical Institute - University - PARMA - ITALY

The first three characters OTL refer to the field of unconstrained optimization, the fourth character S mentions the used Sutti's method, the fifth S indicates that this one is the second implementation of Sutti's method, and the final S and D distinguish the version in single precision from the version in double precision. OTLSSS and OTLSSD and the related subroutines differ only for same declarative stetements and for same library functions.

OTLSSS and OTLSSD read and print the following input parameters: dimension of the variable space, initial approximation of the minimizer, stopping tolerances, initial step lenght of the line search, maximum allowed number of function evaluations. Moreover these routines read the index of printing, then they call respectively the subroutines CNS and CNSD.

CNS and CNSD search for a minimum of a n-dimensional function by the Sutti's method, using function values only (1). This method is intended for quadratic, strictly convex and non-convex functions (1,2,3). It computes a sequence of points of descent by moving along sets of n linearly normalized independent directions. The initial set, consisting of the n coordinate axes, is modified in order to build mutually coniugate directions with respect to the hessian matrix of a quadratic objective function. CNS calls the subroutines SEARCH and CALFUN and CNSD calls SEARD and CALFUD. SEARCH and SEARD search for a minimum of an one-dimensional function by a method using function values only, which is based on quadratic interpolation (3). The method computes a point set bracketing the minimum of the objective function along the search direction and sets the position of the minimum in the vertex of the interpolating parabole. Safeguards to avoid spurious stationary points are provided. SEARCH and SEARD call respectively CALFUN and CALFUD.

CALFUN and CALFUD compute the function value in a required point. These subroutines must be supplied from the user.

The argument lists are the following:

```
SUBROUTINE CNS (XA,N,F,DIR,EPS,EPS1,EPS3,EPS4,IFMAX,XMU,IPRINT)
SUBROUTINE CNSD(XA,N,F,DIR,EPS,EPS1,EPS3,EPS4,IFMAX,XMU,IPRINT)
```

with

XA,real — n-dimensional vector containing, on entry, the user's estimate of the minimizer and, on exit, the computed minimizer;

N,integer — variable specifying the number n of independent variables: N must be assigned before entry;

F,real — variable containing function value in the current point, on exit F contains the estimated value of the minimum;

DIR,real — matrix of the search vectors: DIR is built in OTLSSS and in OTLSSD;

EPS,EPS1,EPS3,
EPS4,real — variables containing the accuracies, to be assigned before entry: EPS and EPS1 must be to the relative accuracies to which the minimizer and the minimum are required, EPS4 and EPS3 scale EPS to the different accurancies EPS2 required in the line searches respetively along the 1-st, 2-nd,..., (n-k)-th direction and along the (n-k+1)-th,...,n-th direction. To make consistent these accuracies, EPS4 should be not smaller than 1 and not biger than 10, while EPS3 should be not smaller than 10^2 and not biger than 10^3, whenever EPS and EPS1 are set to 10^{-5};

IFMAX, integer — variable containing the maximum allowed number of function evaluations: IFMAX must be assigned before entry. It depends from the behaviour and from the dimension of the objective function and from the required accuracies: in the performed proofs IFMAX is set to 10^4;

XMU,real — variable containing the initial step lenght for the line search, to be assigned less or equal to 1 before entry;

IPRINT,integer — parameter controlling print as follows: for IPRINT=1 the current values and the final ones of the cycle index, of the iteration index, and of the minimizer and minimum approximations are printed; for IPRINT=0 only the final values are printed.

SUBROUTINE SEARCH (D,IFMAX1,EPS2,XO,N,FO,MU,X,IFUN)

SUBROUTINE SEARD (D,IFMAX1,EPS2,XO,N,FO,MU,X,IFUN)

with

D,real	n-dimensional vector to be computed before entry;
IFMAX1,integer	variable containing the difference between IFMAX and IFUN to be computed before entry;
EPS2,real	variable containing the accuracy to which the position of the one-dimensional minimum is required:EPS2 must be calculated before entry;
XO,real	variable containing the actual approximation of the minimizer;
FO,real	variable containing the function value in XO;
MU,real	variable containing the step lenght on entry;
X,real	variable containing the step lenght on exit;
IFUN,integer	variable containing the total number of function evaluations;

SUBROUTINE CALFUN (X,N,F,IFUN)

SUBROUTINE CALFUD (X,N,F,IFN)

with

X, real	variable containing the point at which the function value is required;
N,integer	variable specifying the number of independent variables;
F,real	variable containing the function value in X;
IFUN,integer	variable containing the total number of function evaluations.

The lenght of the codes, i.e. the total number of statements in OTLSSS and in OTLSSD are respectively 309 and 313. The size of the problems for which the codes has been designed is $n \leq 50$. The related required storage is of 9.132 words (9.132x16 bits) for OTLSSS and 14.986 words (14.986x16 bits) for OTLSSD. In the above sums none care is taken or of the subroutine CALFUN or of CALFUD.

The test problems solved by OTLSSS and OTLSSD on the PDP 11/40 of 32K words, at the Mathematical Institute, University of PARMA (ITALY), were the minimizations of the following functions:

1 - Extended Rosenbrock
2 - Extended Powell
3 - Oren's Quartic
4 - Penalty I
5 - More first function
6 - Trigonometric
7 - More second function
8 - Brown almost linear
9 - Mancino
10 - Watson
11 - Penalty II
12 - Chebyquad

For the mathematical description of the above functions with the related starting points $X_o = (X_{oi}), i = 1,\ldots,n$, see ref.(4).

The proofs have been performed for $n = 4,10$ and for $n = 4,8$ for the Extended Powell function. Moreover the following initial approximations of the minimizer were assumed: $X_o^1 = (X_{oi}), X_o^2 = (X_{oi} + \Delta_i)$ with $\Delta_i = 10^{-3}(1+|X_o|)$ and $X_o^3 = (10\ X_{oi})$, $i = 1,\ldots,n$. The other input parameters were assigned as above described.

In the annexed listing 1 and 2, we present the executions of the programs OTLSSS and OTLSSD, with IPRINT = 0, for the sample problem

$$\min_{x_1,x_2} 100(x_2-x_1^2)^2 + (1-x_1)^2 \qquad x_o = (-1.2,1)$$

having analytical solution $x_{min} = (1,1)$, $F_{min} = 0$.

The annexed numerical tables 3 and 4 visualize the results obtained by OTLSSS and OTLSSD. The parameter NPROB is the number of the objective function in the above sequence, N is the size of the problem, XZERO indicates which starting vector is tested, CYC is the total number of the performed cycles, ITER the number of the iterations in the final cycle, IFUN the total number of function evaluations and F the computed minimum.