SEE JOE GOOD (7/15/74)
OR RALPH NELSON

Rotraut Weiss
SLAC Computation Group
Stanford, California

CGTM No. 147
March 1973
Revised: April 1973

P R A X I S

A Minimizer for Multidimensional Functions - Without Derivatives

PRAXIS is a FORTRAN routine by R. Brent[(1)] that minimizes a function of several variables without requiring user coded derivatives. It is based on Powell's method of minimizing a function along conjugate axes.

PRAXIS is used widely because

- it is reliable,
- it is efficient,
- it does not require derivatives (this means less coding for the user), and
- it is very easy to use.

We performed a systematic comparison of PRAXIS and two other minimizers that do not require user supplied derivatives: STREAM (Rosenbrock method)[(2)] and MIGRAD (Davidon method)[(3)]. For the considered test functions, PRAXIS proved to be superior. For detailed description of the comparison study, see CGTM (Computation Group Technical Memo) No. 148.

To make PRAXIS available to a job, the following JCL cards should be included:

```
//LKED.X DD DSN=WYL.CG.PUB.LOADMODS,DISP=SHR
//LKED.SYSIN DD *
 INCLUDE X(PRAXIS)
```

PRAXIS is called by the statement

```
Y = PRAXIS(TO,MACHEP,HO,N,PRIN,X,F,FMIN).
```

The calling program must contain the statements

```
REAL*8 TO,MACHEP,HO,X,F,FMIN,PRAXIS
EXTERNAL F
INTEGER PRIN
```

What follows are the instructions to the user as given by Brent:

> PRAXIS returns the minimum of the function F(X,N) of N variables using the principal axis method. The gradient of the function is not required.

The parameters are:

TO — is a tolerance. PRAXIS attempts to return PRAXIS=F(X) such that if XO is the true local minimum near X, then norm(X-XO) < to + square root(MACHEP)*norm(X).

MACHEP — is the machine precision, the smallest number such that 1 + MACHEP > 1. MACHEP should be 16. **-13 (about 2.22D-16) for REAL*8 arithmetic on the IBM 360.

HO — is the maximum step size. HO should be set to about the maximum distance from the initial guess to the minimum. (If HO is set too large or too small, the initial rate of convergence may be slow).

N — (at least two) is the number of variables upon which the function depends.

PRIN — controls the printing of intermediate results. If PRIN=0, nothing is printed. If PRIN=1, F is printed after every N+1 or N+2 linear minimizations. Final X is printed but intermediate X is printed only if N is at most 4.
If PRIN=2, the scale factors and the principal values of the approximating quadratic form are also printed. If PRIN=3, X is also printed after every few linear minimizations.
If PRIN=4, the principal vectors of the approximating quadratic form are also printed.

X — is an array containing, on entry, a guess of the point of minimum; on return, the estimated point of minimum.

F(X,N) — is the function to be minimized. F should be a REAL*8 function declared external in the calling program.

FMIN — is an estimate of the minimum, used only in printing intermediate results.

The approximating quadratic form is

Z(X') = F(X,N) + (1/2) * (X'-X)-transpose * A * (X'-X)

Where X is the best estimate of the minimum and A is

inverse(V-transpose) * D * inverse(V)

(V(*,*) is the matrix of search directions; D(*) is the array of second differences). If F has continuous second derivatives near XO, A will tend to the Hessian of F at XO as X approaches XO.

It is assumed that on floating-point underflow, the result is set to zero.

The user should observe the comment on heuristic numbers after the initialization of machine dependent numbers.

P R A X C G

PRAXCG is the SLAC Computation Group version of PRAXIS, with the following features not in PRAXIS:

- Number of variable parameters can be 1.
- Parameters can be fixed.
- Stepsizes in the different directions need not be equal.
- The covariance matrix and the correlation coefficients are printed.
- The covariance matrix can be made available to the calling program.

To make PRAXCG available to a job, the following JCL cards should be included:

```
//LKED.X DD DSN=WYL.CG.PUB.LOADMODS,DISP=SHR
//LKED.SYSIN DD *
 INCLUDE X(PRAXCG)
```

PRAXCG is called by the statement

```
CALL PRAXCG(N,X,DX,EPSI,ISW,FX,F)
```

The calling program must contain the statements

```
REAL*8 X,DX,EPSI,FX,F
EXTERNAL F
```

The parameters are:

X — is an array containing on entry, guess of the point of minimum,
On return, the estimated point of minimum.

N — is the number of parameters upon which the function F depends.
N must not exceed 30. (See PRAXC 018)

DX — is an array containing the stepsizes;
DX(I) the stepsize in the direction X(I).
If DX(I) .LE.0.DO, X(I) is considered a fixed parameter.

EPSI — is a convergence constant:
If EPSI.GT.0.DO, it is assumed to be user set
(TO = EPSI, see PRAXIS);
If the user does not want to choose EPSI, EPSI must be set .LE.0.DO
(TO = 1.0D-04).

ISW — controls the printing:
If ISE=0, nothing is printed,
If ISW=1, normal printout,
If 2.LE.ISW.LE.5, normal printout and PRIN=ISW-1
(see PRAXIS).

F(X,N) — is the function to be minimized.
F should be a REAL*8 function declared external in the calling program.

FX — is upon return, the minimum of the function F(X,N).

WARNING: The number of variable parameters must not exceed 20. (See PRAXC 024)

To make the covariance matrix available to the calling program, the following common statement should be included:

```
COMMON /VARIAN/ V(20,20)
```

For further description, see comments in function subroutine PRAXIS.

REFERENCES

(1) Richard P. Brent, Algorithms for Minimization without Derivatives, Prentice-Hall, 1973

(2) MINF68 - A General Minimizer Routine, Lawrence Radiation Laboratory, Group A Programming Note No. P-190, 1969

(3) MINUIT - A Program to minimize a function of n variables, compute the covariance matrix, and find the true errors, CERN Computer 6000 Series Program Library, 1969