DECUS
PROGRAM LIBRARY

DECUS NO.	8-483
TITLE	GRFIT, A SIMPLE LEAST SQUARES ROUTINE
AUTHOR	R. C. Gross
COMPANY	Eastman Kodak Company Rochester, New York
DATE	August 25, 1971
SOURCE LANGUAGE	PS/8 FORTRAN

GRFIT, A Simple Least Squares Routine

by R. C. Gross

In the environment in which many small computers are found, there is often apparent a need for a simple statistical routine such as found in GRFIT but not the facilities to write or debug the program. While GRFIT is essentially similar to many other available least squares type programs, this is the first time, to my knowledge, that a similar program has been made available for PDP-8 machines in both Fortran and Focal.* In addition, the Fortran program has been written both as a main program and as a subroutine.

In FORTRAN the program requires a PDP-8 with 8K of memory and one large storage device, either the single DECTAPE or a DF-32 disk so that the facility supports the PS-8 operating system.

In FOCAL the program requires only the minimum configuration PDP-8 with 4K of memory.

The FORTRAN program as dimensioned allows an additional 16 pages of memory free for other material. It can, of course, be considerably reduced in its dimension statement. For users really short of space, the loop following the blank comment line ending at statement 12 can be removed; this removes the actual calculation of both the calculated values of Y and the differences between the calculated and the entered values of Y as well as cutting the dimension requirements in half. The calculated slopes and intercepts as well as most of the error information are still available

* See FOCAL8-209

to the user. The forceoff statement immediately preceding statement 30 in the main program can be changed to any value not necessary as a potential X value if 999 is not satisfactory.

The FOCAL program accepts a much more limited array size, but it will easily accept the 8 to 12 pairs of points that are generally used with such a program. If 8-K FOCAL were available the array size would, of course, be considerably expanded.

I emphasize that the programs are simple, the sort that are available at every sophisticated installation, but might be of use in the somewhat unusual environment in which many PDP-8 computers are installed.

Variables:

X - Array of input x values

Y - Array of input y values

YC - Array of calculated y values

DIFF - Array of calculated differences between Y and YC

NV - Number of points

B - Calculated slope

A - Calculated intercept

R - Correlation coefficient

SUM - Sum of the squares of the residuals

EA - Standard error in intercept

EB - Standard error in slope

Note that Arrays of X and Y, as well as a value for NV, must be passed to the subroutine, while the remainder of the variables

are calculated and returned to the main program by the subroutine. The dimensioning of the main program must account for the returned arrays YC and DIFF.

Cautionary Notes:

In any statistical program the values of the calculated errors must be examined carefully. In the case of the limitations of 12 bit words the accuracy of the results must always be examined carefully, especially when small differences are involved.

Values of YC and DIFF can be produced in the main program without the accompanying dimensioning of them as arrays if they are printed immediately after their calculation and never stored.

```
      DIMENSION X(135),Y(135),YC(135),DIFF(135)
      IN=1
      IOUT=1
      WRITE(IOUT,105)
      WRITE(IOUT,107)
      SX = 0.
      SY = 0.
      SY2 = 0.
      NV = 0
      SXY = 0.
      SX2 = 0.
      DO 30 J=1,200
      NV = NV + 1
      READ(IN,100)X(J),Y(J)
      IF (X(J)-999.)30,40,30
30    CONTINUE
40    NV=NV-1
      TV=FLOAT(NV)
      DO 11 I=1,NV
      SX = SX + X(I)
      SY = SY + Y(I)
      SXY = SXY + X(I) * Y(I)
      SY2 = SY2 + Y(I) * Y(I)
11    SX2 = SX2 + X(I) * X(I)
      SSX = SX * SX
      B=(TV*SXY-SX*SY)/(TV*SX2-SSX)
      A = (SY-B*SX)/TV
      R=(TV*SXY-SX*SY)/(SQRT(TV*SX2-SX*SX)*SQRT(TV*SY2-SY*SY))
     RR=SQRT((SY2-A*SY-B*SXY)/(TV-2.))
     EA=RR*SQRT(SX2/(TV*SX2-SX*SX))
     EB=RR*SQRT(TV/(TV*SX2-SX*SX))
      SUM = 0.
C
      DO 12 I=1,NV
      YC(I) = A+ B*X(I)
      DIFF(I) = YC(I) - Y(I)
      SUM = SUM + DIFF(I) * DIFF(I)
12    CONTINUE
      WRITE(IOUT,106)
      DO 50 I=1,NV
      WRITE(IOUT,210)X(I),Y(I),YC(I),DIFF(I)
50     CONTINUE
105   FORMAT(44H INPUT FORMAT IS 2F10.4,  X=999. TERMINATES    )
      WRITE(IOUT,215)B,A
      WRITE(IOUT,216)R,SUM
     WRITE(IOUT,218)EB,EA
218   FORMAT(' SLOPE ST ERROR=',E10.3,' INTERCEPT ST ERROR=',E10.3)
215   FORMAT(7H SLOPE=F10.4,11H INTERCEPT=F10.4)
216   FORMAT(13H CORR.COEFF =F10.4,18H SUM SQ.RESID.ERR=F10.4)
106   FORMAT(43H       X(I)    Y(I)      YC(I)    DIFF(I)     )
100   FORMAT(2F10.4)
210   FORMAT(4F10.4)
107   FORMAT('    X          Y')
      CALL EXIT
      END
```

Fortran Subroutine Listing

```
      SUBROUTINE GRFIT(X,Y,NV,YC,DIFF,B,A,R,SUM,EA,EB)
      DIMENSION X(1),Y(1),YC(1),DIFF(1)
       SX = 0.
       SY = 0.
       SY2 = 0.
       NV = 0
       SXY = 0.
       SX2 = 0.
       TV = FLOAT(NV)
       DO 11 I=1,NV
       SX = SX + X(I)
       SY = SY + Y(I)
       SXY = SXY + X(I) * Y(I)
       SY2 = SY2 + Y(I) * Y(I)
11     SX2 = SX2 + X(I) * X(I)
       SSX = SX * SX
       B=(TV*SXY-SX*SY)/(TV*SX2-SSX)
       A = (SY-B*SX)/TV
       R=(TV*SXY-SX*SY)/(SQRT(TV*SX2-SX*SX)*SQRT(TV*SY2-SY*SY))
       SUM = 0.
      RR=SQRT((SY2-A*SY-B*SXY)/(TV-2.))
      EA=RR*SQRT(SX2/(TV*SX2-SX*SX))
      EB=RR*SQRT(TV/(TV*SX2-SX*SX))
       DO 12 I=1,NV
       YC(I) = A+ B*X(I)
       DIFF(I) = YC(I) - Y(I)
       SUM = SUM + DIFF(I) * DIFF(I)
12     CONTINUE
      RETURN
       CALL EXIT
       END
```

Typical Data Set - Fortran

```
 INPUT FORMAT IS 2F10.4,  X=999. TERMINATES
    X           Y
0.             0.
1.              1.
2.              2.08
3.              2.95
4.              4.
999.
       X(I)      Y(I)        YC(I)      DIFF(I)
    0.0000     0.0000     0.0160     0.0160
    1.0000     1.0000     1.0110     0.0110
    2.0000     2.0800     2.0060    -0.0740
    3.0000     2.9500     3.0010     0.0510
    4.0000     4.0000     3.9960    -0.0040
SLOPE=     0.9950 INTERCEPT=     0.0160
CORR.COEFF =     0.9996 SUM SQ.RESID.ERR=     0.0085
SLOPE ST ERROR= 0.168E-01 INTERCEPT ST ERROR= 0.411E-01
```