(4)

ARL-STRUC-TM-490 AR-005-528

AD-A204 087

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Aircraft Structures Technical Memorandum 490

SMOOFF - A FORTRAN SMOOTHING PROGRAM (U)

by

I.H. Grundy

Approved for Public Release

JULY 1988

89 2 14 050

AR-005-528

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Aircraft Structures Technical Memorandum 490

SMOOFF - A FORTRAN SMOOTHING PROGRAM (U)

by

I.H. Grundy

SUMMARY

The FORTRAN program SMOOFF (SMOOthing by a Filtered Fourier transform) smooths a "noisy" function of two variables on a rectangular domain, by minimizing an appropriate functional subject to a least squares constraint.

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

CONTENTS

1. INTRODUCTION

Techniques for smoothing data over a two dimensional region are well known from the theory of image processing [1], [2]. They range from local methods e.g. using finite difference schemes such as five-point Laplacian smoothing, to global methods in which a smooth function is fitted in some fashion to the "noisy" input function over the whole region of interest.

One such global method is described in [2] pp. 53-76, in which the smooth output function is determined by minimising a given functional, namely the mean square value of the output function over the region, subject to the constraint that the output function should be sufficiently close to the input function in a least squares sense.

An alternative strategy is also given in [2] in which the functional to be minimised is taken to be the mean square value of the maximum directional derivative of the output function over the region. This article will describe a FORTRAN program SMOOFF, which implements this method on a rectangular grid.

In SMOOFF the constrained minimisation problem described above is approached by writing both the input function and the smooth output function as two-dimensional Fourier series. The problem then reduces to one of finding the Fourier coefficients which satisfy the least squares constraint. This constrained minimisation problem is solved using Lagrange multipliers.

The effect of the minimisation is to weight the high frequency coefficients of the new Fourier series relative to the old, such that these terms are attenuated. This process is governed by a single parameter which enters through the least squares constraint. By varying this parameter we can cause as little or as much smoothing as we desire to take place.

2. PROBLEM FORMULATION

The configuration is as depicted in Figure 1(a). The grid consists of N_x points (labelled $0,\dots,N_x-1$) evenly spaced a distance of Δ_x apart in the x-direction, and N_y points (labelled $0,\dots,N_y-1$) evenly spaced a distance of Δ_y apart in the y-direction. At each grid point we are given a value of the function $\hat{u}(x,y)$. This function may contain a significant random noise component. Our task is to find a smooth function $u(x,y)$ which approximates the noisy input function $\hat{u}(x,y)$ sufficiently closely.

Later we will seek to write u and $\hat{u}$ as Fourier series. A good fit of these Fourier series to the actual functions relies, however, on periodic continuations of u and $\hat{u}$ outside the grid being continuous. What this means is that the values of u and $\hat{u}$ at opposite edges of the grid must be identical, otherwise there will be a step discontinuity in the periodic continuation. In general there will be such a discontinuity, causing undesirable edge effects (the Gibbs phenomenon) to occur. This difficulty is avoided in the following manner:

We expand the grid to form the one shown in Figure 1(b). The points in this new grid are labelled

$$(x_i,y_j)=(i\Delta_x,j\Delta_y),\quad i=0,\dots,L_x-1,\quad j=0,\dots,L_y-1\ . \tag{2.1}$$

where $L_x=2N_x-1$ and $L_y=2N_y-1$. The functions $\hat{u}(x,y)$ and $u(x,y)$ are then extended by reflecting them in the lines $x=a/2$ and $y=b/2$, (where $a=L_x\Delta_x$ and $b=L_y\Delta_y$).

The newly constructed functions u and $\hat{u}$ have symmetry in both the x and y directions and thus can be represented in terms of two-dimensional Fourier cosine series. Further, by symmetry the values of u and $\hat{u}$ are equal at opposite edges of the expanded region so the Fourier representation of these functions is excellent. Any edge effects are now due to a *slope* discontinuity as opposed to a *function* discontinuity, so their severity is minimised. (Note that the last row and column of the grid are deleted in Figure 1(b) and also in (2.1). These points are now redundant from the assumed periodicity.)

The criterion for smoothing $\hat{u}$ is that given in [2], namely that u should minimise the functional

$$I = \int_0^b \int_0^a \Big[\big(\frac{\partial u}{\partial x}\big)^2 + \big(\frac{\partial u}{\partial y}\big)^2\Big] dx\, dy \, . \tag{2.2}$$

subject to the constraint that

$$\sum_{i=0}^{L_x-1} \sum_{j=0}^{L_y-1} [u(x_i, y_j) - \hat{u}(x_i, y_j)]^2 = L_x L_y \sigma^2 \, . \tag{2.3}$$

The quantity σ is the specified smoothing tolerance. If σ is zero, no smoothing occurs. If σ is taken to be too large the smoothing is effectively unconstrained. From (2.3) it is clear that the optimal value of σ is closely related to the standard deviation of the noise on the data.

The integrand in equation (2.2) is equal to the square of the maximum of the directional derivative at each point. Thus, a noisy function with an underlying harmonic nature will be particularly well treated, i.e. without much distortion, since the underlying function minimises I by definition. This makes the method particularly attractive for dealing with the output from SPATE, which has an underlying harmonic nature (see [3] and [4]). Even so, the method is quite general and can be applied as a "black-box" smoothing routine for anharmonic functions, provided the noise level is not too great.

To determine a suitable u, we proceed by writing u and $\hat{u}$ as Fourier cosine series in x and y, namely

$$u = \sum_{n=0}^{N_p} \sum_{m=0}^{N_q} D_{nm} \cos \frac{2n\pi x_i}{a} \cos \frac{2m\pi y_j}{b} \tag{2.4}$$

and

$$\hat{u} = \sum_{n=0}^{N_p} \sum_{m=0}^{N_q} \hat{D}_{nm} \cos \frac{2n\pi x_i}{a} \cos \frac{2m\pi y_j}{b} \, . \tag{2.5}$$

The upper summation bounds N_p and N_q are chosen so as to give a sufficient number of terms in the Fourier expansion to represent the smoothed function accurately. Ideally $N_p = N_x - 1$ and $N_q = N_y - 1$, i.e. the number of unknowns is equal to the number of original grid points.

The Fourier coefficients $\hat{D}_{nm}$ are known from $\hat{u}$ by the formulae

$$w_{nm} \hat{D}_{nm} = \frac{4}{L_x L_y} \sum_{i=0}^{L_x} \sum_{j=0}^{L_y} \hat{u}(x_i, y_j) \cos \frac{2n\pi x_i}{a} \cos \frac{2m\pi y_j}{b} \, , \tag{2.6}$$

where

$$w_{nm} = \begin{cases} 1, & \text{if } n \neq 0 \text{ and } m \neq 0 \\ 2, & \text{if one of } n \text{ or } m = 0 \\ 4, & \text{if } n = m = 0 \, . \end{cases} \tag{2.7}$$

The expression (2.4) for u is substituted into (2.2) and (2.3). Uniformity of the grid is important here as it allows us to make use of the discrete orthogonality property of sines and cosines, and this allows simplification of the constraint equation. The problem becomes one of finding the coefficients D_{nm} which minimise the sum

$$I^* = \sum_{n=0}^{N_p} \sum_{m=0}^{N_q} w_{nm} F_{nm} D_{nm}^2 , \tag{2.8}$$

where

$$F_{nm} = \frac{n^2}{a^2} + \frac{m^2}{b^2} , \tag{2.9}$$

subject to the constraint that

$$\sum_{n=0}^{N_p} \sum_{m=0}^{N_q} w_{nm} (D_{nm} - \hat{D}_{nm})^2 = 4\sigma^2 . \tag{2.10}$$

To solve this constrained optimisation problem we use the Lagrange multiplier λ to write

$$\sum_{n=0}^{N_p} \sum_{m=0}^{N_q} w_{nm} F_{nm} D_{nm}^2 + \lambda \Big\{ \sum_{n=0}^{N_p} \sum_{m=0}^{N_q} w_{nm} (D_{nm} - \hat{D}_{nm})^2 - 4\sigma^2 \Big\} = 0 . \tag{2.11}$$

Differentiation with respect to each of the coefficients in turn gives that coefficient in terms of its known hatted counterpart, i.e.

$$D_{nm} = \frac{\lambda \hat{D}_{nm}}{\lambda + F_{nm}} , \tag{2.12}$$

for all n and m.

The next step is to find the single unknown λ. This is accomplished by substituting (2.12) into the constraint equation (2.10), to give a single non-linear algebraic equation for λ, namely

$$E(\lambda) = \sum_{n=0}^{N_p} \sum_{m=0}^{N_q} \frac{w_{nm} F_{nm}^2}{(\lambda + F_{nm})^2} \hat{D}_{nm}^2 - 4\sigma^2 = 0 . \tag{2.13}$$

This equation is solved numerically by a Newton iteration scheme. We observe that E is a monotonically decreasing function of λ, hence (2.13) will have a unique solution for λ. A non-negative solution is guaranteed provided $E(0) \geq 0$. The critical case occurs when $E(0) = 0$. This case corresponds to all the Fourier coefficients of u being zero (see 2.12), with the exception of the constant term D_{00}. That is, it corresponds to unconstrained smoothing and total loss of the original information. Using (2.13) we can determine the critical σ value, namely

$$\sigma_{max} = \frac{1}{2} \sqrt{\sum_{n=0}^{N_p} \sum_{m=0}^{N_q} w_{nm} \hat{D}_{nm}^2 - w_{00} \hat{D}_{00}^2} , \tag{2.14}$$

which gives rise to this unconstrained smoothing.

With λ finally determined, it remains to calculate the Fourier coefficients D_{nm} from equation (2.12), and to substitute these back into the equation (2.4) for u. From this series u can be evaluated anywhere within the rectangular region, and in particular at the original grid points.

3. PROGRAM DESCRIPTION

The smoothing algorithm described above has been implemented in the double precision FORTRAN subroutine SMOOFF. This subroutine is divided into three distinct stages.

The first stage involves expanding the grid and determining the Fourier coefficients DH(N,M) of the expanded UDATA on this grid. Symmetry of UDATA about the lines $x = a/2$ and $y = b/2$ is used to offset the extra computational effort caused by the grid expansion.

The second stage of SMOOFF involves determining the Lagrange multiplier ALAM by Newton iteration. This is the fastest stage, being an order of magnitude quicker than stages one and three. Convergence always occurs within ten iterations. The critical SIG value SIGMAX is also determined here. If SIG is greater than SIGMAX then the subroutine exits immediately without going on to the third stage.

The third stage of SMOOFF involves back-substitution of ALAM to give the Fourier coefficients DS(N,M) of the output function UOUT. Further back-substitution gives the smooth output UOUT itself at the grid points. Again symmetry is used to reduce the computational effort by something approaching a factor of four.

A typical call to SMOOFF looks like

```
CALL SMOOFF(NX,NY,DX,DY,SIG,SIGMAX,NA,UDATA,UOUT,IOP,IER,IMESSG).
```

The input parameters are

NX the number of original grid points in the x-direction,
NY the number of original grid points in the y-direction,
DX the grid-spacing in the x-direction,
DY the grid-spacing in the y-direction,
SIG the smoothing tolerance,
NA the size of UDATA and UOUT in the calling program,
UDATA the array of raw function values, and
IOP the fast/slow run option

The input parameter IOP deserves a special mention here. This parameter controls the number of Fourier terms N_p and N_q which are considered. The default is for the ideal case: $N_p = N_x - 1$ and $N_q = N_y - 1$. For IOP equal to unity however, N_p and N_q are halved from their ideal values, in effect setting the remaining (high frequency) terms to zero and truncating the Fourier series. This is done in order to reduce the computation time by a factor of four. A side effect though, is that the smoothing parameter σ no longer relates directly to the noise level as in (2.3). A significantly smaller value of σ needs to be chosen, a fact which is clear from inspection of (2.11) and (2.12). Nevertheless, for cases with a lot of grid points, say greater than 40×40 the fast run option gives reasonable results in acceptable time. The option should not be used however, if a derivative of u is required, or if u itself is expected to contain high frequency components.

The output parameters of SMOOFF are

SIGMAX the critical value of SIG,
UOUT the array of smoothed function values,
IER the error number, and
IMESSG the error / normal-completion message.

To use SMOOFF the following declarations should be made in the calling program:

```
INTEGER NX,NY,IER,IOP
CHARACTER*50 IMESSG
REAL*8 DX,DY,SIG,SIGMAX
DIMENSION UDATA(0:NA,0:NA),UOUT(0:NA,0:NA)
PARAMETER (NA=199).
```

4. SAMPLE RESULTS

Sample results, consisting of surface and contour plots obtained using the NCAR plotting package, are presented here for the simple test function $\hat{u}(x,y) = xy$, both with and without added random (Gaussian) noise. The results are presented for a uniform 50×50 grid on $[0,1] \times [0,1]$. In each case the calculation of the smoothed approximation took around 110 CPU secs using the slow run option, of which about half was spent in stage 1 and about half in stage 3. (For a 120×120 grid the total CPU time is of the order of 2 hrs.)

Figure 2(a) shows a surface plot of the original function $\hat{u}$ without noise. Figure 2(b) shows the corresponding contour plot. The contours are (naturally) the hyperbolae $xy = c$.

Figure 3(a) shows $\hat{u}$ with added Gaussian noise with a standard deviation of 0.05. Figure 3(b) shows the corresponding contour plot. The original contours are now largely obscured by noise.

Figure 4(a) shows the effect of SMOOFF on the noisy $\hat{u}$ of Figure 3. The smoothing level is $\sigma = 0.05$ ($\sigma_{max} = 0.222$), which is precisely equal to the noise level. Clearly, this is very near the optimal value for σ.

Finally, Figures 5(a) and 5(b) show the effect of an excessively high choice of the smoothing tolerance ($\sigma = 0.2$). There is significant loss of information, demonstrated by gross distortion of the contours and flattening of the surface. (It should be pointed out here that the surface plot is not drawn to scale, and is rather flatter than it appears!)

5. CONCLUSION

The FORTRAN subroutine SMOOFF implements the method of smoothing noisy data on a rectangular grid, described in [2]. Numerical evidence presented for sample data in Section 4 shows that SMOOFF is very effective provided sufficient care is taken with the size of the smoothing parameter. Applications have already been found at ARL in smoothing output from SPATE (see [3], [4]).

6. ACKNOWLEDGEMENT

The author wishes to thank Dr. T. G. Ryall for numerous helpful discussions during the preparation of this work.

REFERENCES

[1] A. Rosenfeld, *Picture Processing By Computer*, Academic Press, New York, 1969.

[2] B. R. Hunt, "Image Restoration", *Digital Image Processing Tecniques*, Ed. M. P. Ekstrom, Academic Press, London, pp. 53–76, 1983.

[3] A. K. Wong, S. A. Dunn & J. G. Sparrow, "Residual Stress Measurement by Means of the Thermoelastic Effect", Nature, Vol. 332, No. 6165, pp. 613–615, 1988.

[4] M. Heller, S. A. Dunn, J. F. Williams & R. Jones, "Thermomechanical Analysis of Metallic and Composite Specimens", paper in preparation, 1988

APPENDIX - PROGRAM LISTING

```
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SNOOFF(NX,NY,DX,DY,SIG,SIGMAX,NA,
     *                                  UDATA,UOUT,IOP,IER,INESSG)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     AUTHOR: Ian Grundy 12/2/87.
C
C
C       This subroutine smooths noisy input data UDATA, defined on
C     a rectangular region containing an NX*NY distribution of
C     points a distance of DX and DY apart respectively.  (There
C     is no restriction on the oddness or evenness of NX and NY.)
C     The resulting smoothed data are written into UOUT.  The
C     degree of smoothing is controlled by the parameter SIG.
C
C       This process is implemented in three stages.  In Stage 1,
C     the grid and the raw input data UDATA are reflected in
C     vertical and horizontal lines of symmetry to form a grid of
C     size four times the original.  UDATA is then expressed as a
C     two-dimensional Fourier cosine series whose coefficients DH
C     are then computed.  Here the number of grid points in the x
C     direction is LX=2*NX-1 and the number in the y direction is
C     LY=2*NY-1 (since periodicity is assumed at the far
C     boundary).
C
C       The Fourier coefficients are input to Stage 2, where the
C     Lagrange multiplier ALAM is used to satisfy a variational
C     principle (minimise the integral of the squares of the first
C     derivatives of UOUT over the region), subject to the
C     constraint that the "distance" between UOUT and the known
C     Fourier series for UDATA is smaller the smoothing tolerance
C     LX*LY*SIG*SIG.
C
C       ALAM is input to Stage 3 where the Fourier coefficients DS
C     of UOUT, and finally UOUT are computed.
C
C     NOTES:
C     (1) SIG=0 implies that UOUT is exactly equal to the
C         Fourier series for UDATA, i.e. effective smoothing
C         will not take place.
C
C     (2) If SIG is taken too large, i.e.  larger than SIGMAX
C         which is computed within STAGE 2, the smoothing is
C         effectively unconstrained, ALAM is set to zero and the
C         smoothed function UOUT becomes constant over the region.
C
C     (3) Error conditions.
C         IER=0     No Error
C         IER=1     Fatal Error
C         IER=2     Exceptional Cases (1) and (2)
C         INESSG gives the error message.
```

```
C
C     (4) IOP=1.  Time saver mode.  The program runs for a reduced
C           number of Fourier coefficients, in effect setting the
C           remainder to zero.
C
C     DECLARATIONS:
C       In the calling program the variables should be declared in
C     the following manner:
C
C     INTEGER NX,NY,IER,IOP
C     CHARACTER*50 IMESSG
C     REAL*8 DX,DY,SIG,SIGMAX
C     DIMENSION UDATA(0:NA,0:NA),UOUT(0:NA,0:NA)
C     PARAMETER (NA=199)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      IMPLICIT  DOUBLE PRECISION (A-H,O-Z)
      INTEGER   LX,LY,NX,NY,NA,IER,IOP
      CHARACTER*50 IMESSG
      REAL*8    DX,DY,SIG,SIGMAX,ALAM
      DIMENSION DH(0:NU,0:NU),DS(0:NU,0:NU),
     *          UDATA(0:NU,0:NU),UOUT(0:NU,0:NU),
     *          DSNX(0:NU2),DCSX(0:NU2),DSNY(0:NU2),DCSY(0:NU2)
      PARAMETER (NU=199,NU2=399)
C
      PI=4.0D0*DATAN(1.0D0)
      LX=2*NX-1
      LY=2*NY-1
      A=LX*DX
      B=LY*DY
      TNN=16.0D0/DFLOAT(LX*LY)
      PNX=2.0D0*PI/DFLOAT(LX)
      PNY=2.0D0*PI/DFLOAT(LY)
C
C     If IOP=1 the program works with a reduced number of Fourier
C     coefficients, in effect setting the remainder to zero.  This
C     becomes attractive beyond NX=40,NY=40.
C
      IF (IOP.EQ.1) THEN
        NXH=(NX-1)/2
        NYH=(NY-1)/2
      ELSE
        NXH=NX-1
        NYH=NY-1
      ENDIF
C
C     Error Tests. If there is an error, IER=1.
C
      IER=0
      IF (DX.LE.0.OR.DY.LE.0) THEN
        IMESSG='DX or DY is negative or zero'
        IER=1
      ENDIF
      IF (NX.LE.0.OR.NY.LE.0) THEN
        IMESSG='NX or NY is negative or zero'
        IER=1
      ENDIF
      IF (SIG.LT.0.0D0) THEN
```

```
          IMESSG='Smoothing tolerance SIG must be non-negative'
          IER=1
        ENDIF
        IF (NA.NE.NU) THEN
          IMESSG='Array dimensions NA and NU must be equal'
          IER=1
        ENDIF
        IF (IER.EQ.1) RETURN
C
C       Test for zero sigma
C
        IF (SIG.LE.1E-10) THEN
          DO 2 I=0,NX-1
            DO 1 J=0,NY-1
              UOUT(I,J)=UDATA(I,J)
    1       CONTINUE
    2     CONTINUE
          IMESSG='SIG is zero. No smoothing done.'
          IER=2
          RETURN
        ENDIF
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       STAGE 1:
C
C       Takes raw data UDATA and computes its two-dimensional
C       Fourier series with coefficients DH(N,M).
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Set up array of sines and cosines
C
        DO 101 I=0,LX
          DSNX(I)=DSIN(PNX*I)
          DCSX(I)=DCOS(PNX*I)
  101   CONTINUE
        DO 102 J=0,LY
          DSNY(J)=DSIN(PNY*J)
          DCSY(J)=DCOS(PNY*J)
  102   CONTINUE
C
C       Find DH for all N and M.
C
        DO 110 N=0,NXH
          DO 109 M=0,NYH
            DH(N,M)=UDATA(0,0)
            DO 105 I=1,NX-1
              NI=MOD(N*I,LX)
              DH(N,M)=DH(N,M)+2.0D0*DCSX(NI)*UDATA(I,0)
  105       CONTINUE
            DO 106 J=1,NY-1
              MJ=MOD(M*J,LY)
              DH(N,M)=DH(N,M)+2.0D0*DCSY(MJ)*UDATA(0,J)
  106       CONTINUE
            DH(N,M)=DH(N,M)/4.0D0
C
C       Sum over space variables
C
```

```
      DO 108 I=1,NX-1
        TD=0.0D0
        DO 107 J=1,NY-1
          NJ=MOD(M*J,LY)
          TD=TD+UDATA(I,J)*DCSY(NJ)
107     CONTINUE
        NI=MOD(N*I,LX)
        DH(N,M)=DH(N,M)+DCSX(NI)*TD
108   CONTINUE
C
      WT=2.0D0
      IF (N.GT.0.AND.M.GT.0) WT=1.0D0
      IF (N.EQ.0.AND.M.EQ.0) WT=4.0D0
      DH(N,M)=DH(N,M)*TNN/WT
109   CONTINUE
110 CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     STAGE 2:
C
C     Finds Lagrange multiplier ALAN on which the smoothed
C     solution depends.  This involves solving a non-linear
C     algebraic equation for ALAN using a modified Newton
C     iteration scheme.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Find initial estimate for ALAN and critical sigma value
C     SIGMAX.
C
      ALAN=0.0D0
      SIGMAX=0.0D0
      DO 202 N=0,NXH
        DO 201 M=0,NYH
          IF (N.EQ.0.AND.M.EQ.0) GOTO 201
          FNM=N*N/A/A+M*M/B/B
          WT=2.0D0
          IF (N.GT.0.AND.M.GT.0) WT=1.0D0
          SNM=WT*DH(N,M)*DH(N,M)
          SIGMAX=SIGMAX+SNM
          HNM=SNM*FNM*FNM
          ALAN=ALAN+HNM
201     CONTINUE
202   CONTINUE
      SIGMAX=0.5D0*DSQRT(SIGMAX)
C
C     Test for unconstrained case
C
      IF (SIG.GE.SIGMAX) THEN
        DO 204 I=0,NX-1
          DO 203 J=0,NY-1
            UOUT(I,J)=DH(0,0)
203       CONTINUE
204     CONTINUE
        IMESSG='SIG is too large. Smoothing unconstrained.'
        IER=2
        RETURN
      ENDIF
C
```

```
C     Iterate to find ALAN
C
      IF (SIG.LT.0.25D0*SIGMAX) THEN
        ALAN=0.5D0*(DSQRT(ALAN)/2.0D0/SIG)
      ELSE
        ALAN=0.0D0
      ENDIF
 205  CONTINUE
      FLAN=0.0D0
      FDLAN=0.0D0
      DO 207 N=0,NXH
        DO 206 M=0,NYH
          IF (N.EQ.0.AND.M.EQ.0) GOTO 206
          FNM=N*N/A/A+M*M/B/B
          WT=2.0D0
          IF (N.GT.0.AND.M.GT.0) WT=1.0D0
          HNM=WT*FNM*FNM*DH(N,M)*DH(N,M)
          AF=(ALAN+FNM)
          FLAN=FLAN+HNM/AF/AF
          FDLAN=FDLAN+HNM/AF/AF/AF
 206    CONTINUE
 207  CONTINUE
      FDLAN=-2.0D0*FDLAN
      FLAN=FLAN-4.0D0*SIG*SIG
      ALANOLD=ALAN
      ALAN=ALAN-FLAN/FDLAN
C
C     If ALAN becomes negative, halve the initial guess for it.
C
      IF (ALAN.LT.0.0D0) ALAN = 0.5D0*ALANOLD
C
      IF (ABS((ALANOLD-ALAN)/ALAN).GT.1E-10) GOTO 205
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     STAGE 3:
C
C     Substitutes the calculated value for ALAN into the
C     expressions linking the Fourier coefficients DH and DS for
C     all N and M.  UOUT is then calculated at every grid point
C     using its now known Fourier series.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Calculate the smoothed Fourier coefficients DS.
C
      DO 302 N=0,NXH
        DO 301 M=0,NYH
          IF (N.EQ.0.AND.M.EQ.0) GOTO 301
          FNM=N*N/A/A+M*M/B/B
          AF=ALAN+FNM
          DS(N,M)=ALAN*DH(N,M)/AF
 301    CONTINUE
 302  CONTINUE
      DS(0,0)=DH(0,0)
C
C     Use the calculated coefficients to calculate UOUT at all
C     grid points.
C
      DO 306 I=0,NX-1
```

```
      DO 305 J=0,NY-1
C
        UOUT(I,J)=0.0D0
        DO 304 N=0,NXH
          TD=0.0D0
          DO 303 M=0,NYH
            MJ=MOD(M*J,LY)
            TD=TD+DS(N,M)*DCSY(MJ)
 303      CONTINUE
          NI=MOD(N*I,LX)
          UOUT(I,J)=UOUT(I,J)+DCSX(NI)*TD
 304    CONTINUE
 305  CONTINUE
 306 CONTINUE
C
     IMESSG='SMOOFF Normal Completion. Have a nice Day.'
     RETURN
     END
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

FIGURES

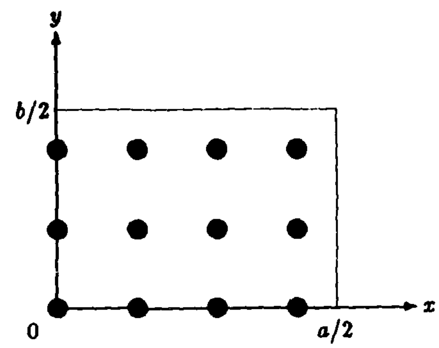

FIGURE 1(a). *Original grid.*

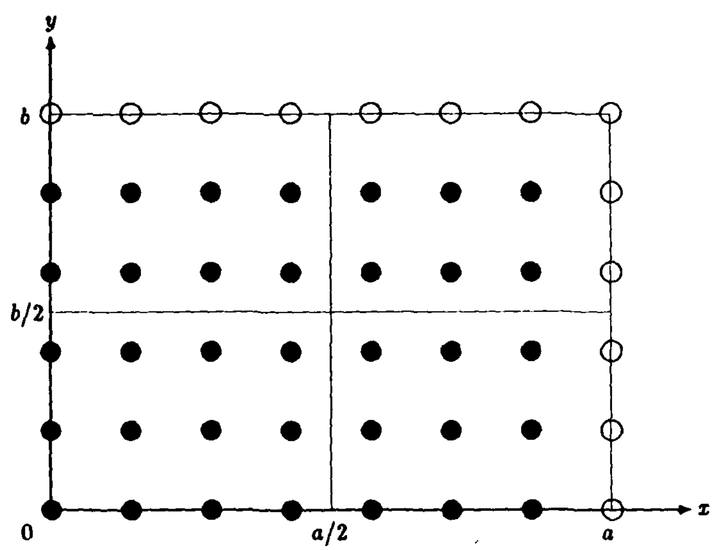

FIGURE 1(b). *Expanded grid.*

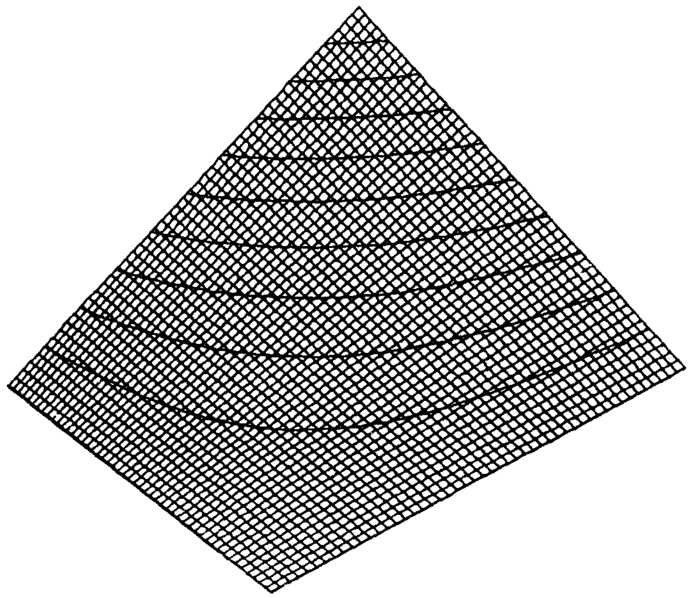

FIGURE 2(a). *Surface plot of original input function.*

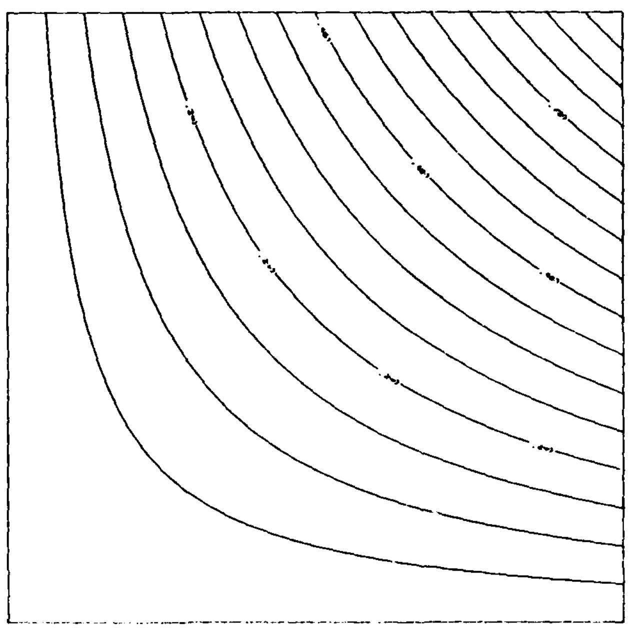

FIGURE 2(b). *Corresponding contour plot.*

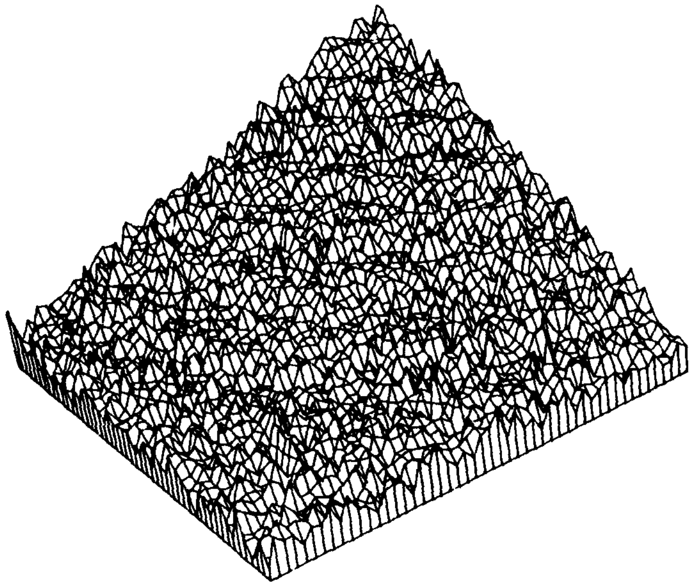

FIGURE 3(a). *Surface plot of noisy input function (original input with added Gaussian noise - standard deviation 0.05).*

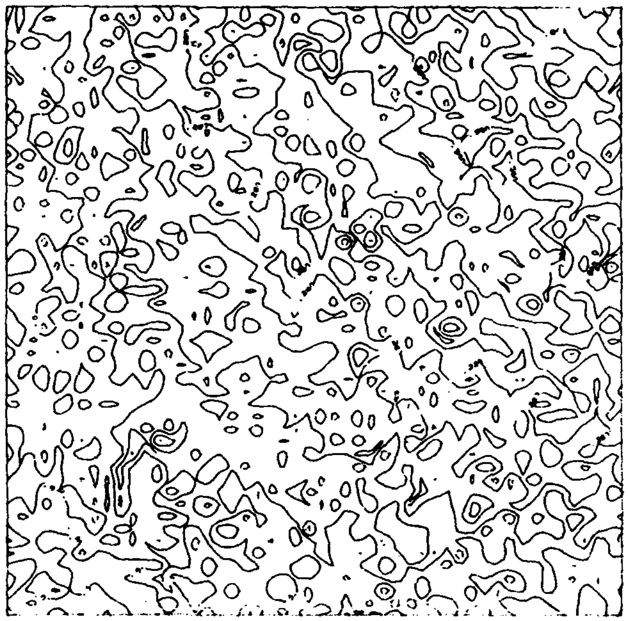

FIGURE 3(b). *Corresponding contour plot.*

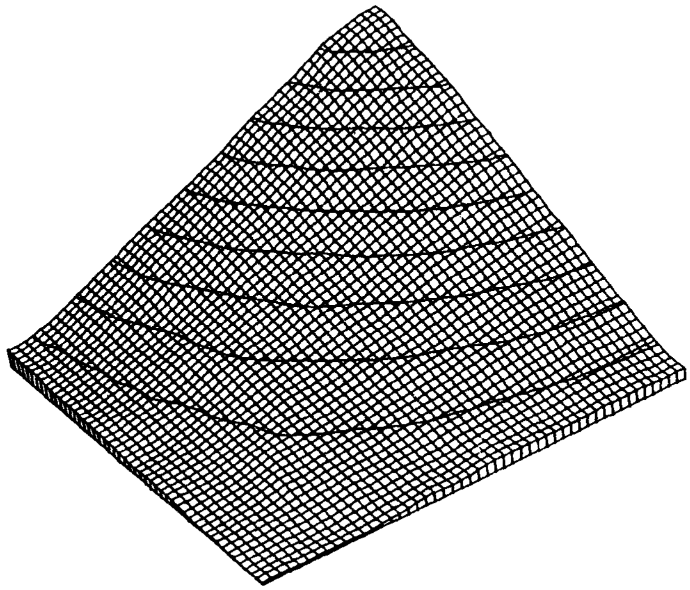

FIGURE 4(a). *Surface plot of noisy input function after application of SMOOFF at $\sigma = 0.05$.*

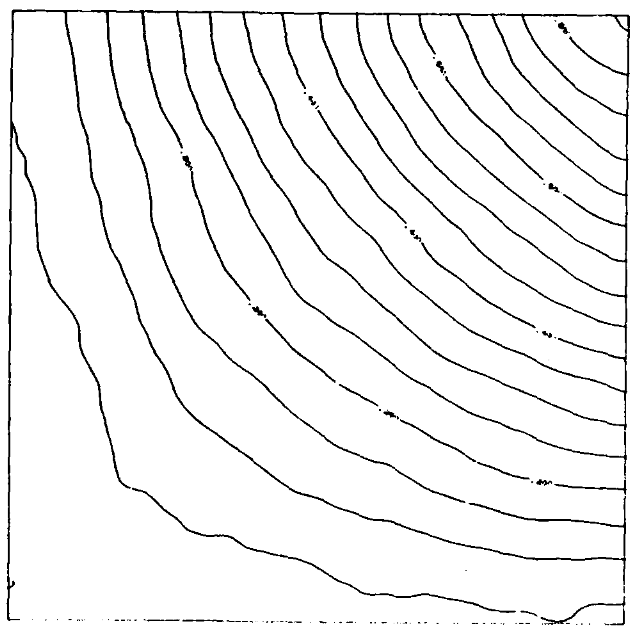

FIGURE 4(b). *Corresponding contour plot.*

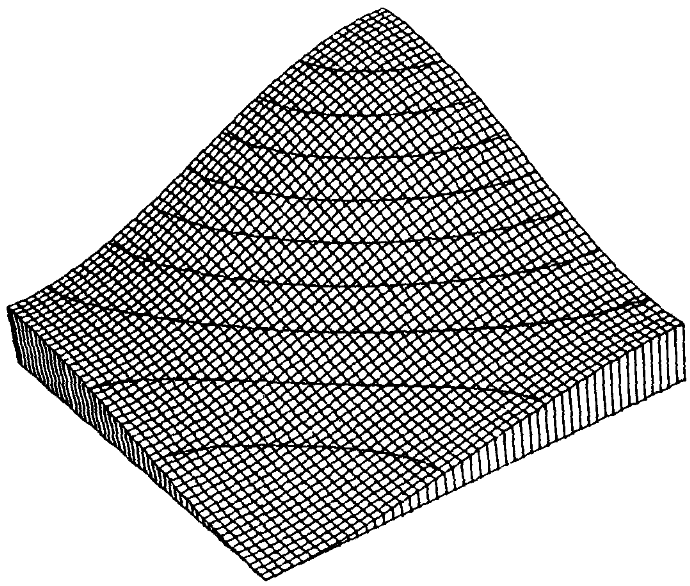

FIGURE 5(a). *Surface plot of noisy function after processing by* SMOOFF *showing the effect of an excessive* σ value ($\sigma = 0.2$).

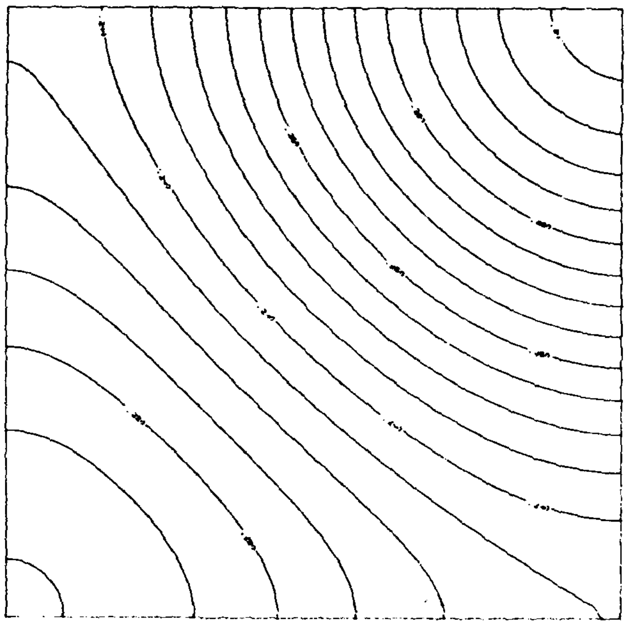

FIGURE 5(b). *Corresponding contour plot.*

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
Chief Defence Scientist
Assist Chief Defence Scientist, Operations (shared copy)
Assist Chief Defence Scientist, Policy (shared copy)
Director, Departmental Publications
Counsellor, Defence Science London (Doc Data Sheet Only)
Counsellor, Defence Science Washington (Doc Data Sheet Only)
S.A. to Thailand MRD (Doc Data Sheet Only)
S.A. to the DRC (Kuala Lumpur) (Doc Data Sheet Only)
OIC TRS, Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratory
Director
Library
Divisional File - Aircraft Structures
Author: I.H. Grundy
B.C. Hoskin
R. Jones
J.G. Sparrow
T.G. Ryall
A.K. Wong
S.A. Dunn
M. Heller
D. Lombardo

Materials Research Laboratory
Director/Library

Defence Science & Technology Organisation - Salisbury
Library

WSRL
Maritime Systems Division (Sydney)

Navy Office
Navy Scientific Adviser (3 copies Doc Data sheet)

Army Office
Scientific Adviser - Army (Doc Data sheet only)
Engineering Development Establishment, Library

Air Force Office
Air Force Scientific Adviser (Doc Data sheet only)

Engineering Division Library
Director General Aircraft Engineering - Air Force

Universities and Colleges

Adelaide
Barr Smith Library

NSW
Library, Australian Defence Force Academy

RMIT
Library

SPARES (10 copies)
TOTAL (58 copies)

AL 149 DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION	UNCLASSIFIED
PRIVACY MARKING	

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-005-528	ARL-STRUC-TM-490	JULY 1988	DST 86/019

4. TITLE
SMOOFF - A FORTRAN SMOOTHING PROGRAM

5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF.(C) RESTRICTED (R), UNCLASSIFIED (U)).

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES
17

7. NO. REFS.
4

8. AUTHOR(S)
I.H. Grundy

9. DOWNGRADING/DELIMITING INSTRUCTIONS
Not applicable

10. CORPORATE AUTHOR AND ADDRESS
AERONAUTICAL RESEARCH LABORATORY
P.O. BOX 4331, MELBOURNE VIC 3001

11. OFFICE/POSITION RESPONSIBLE FOR:
SPONSOR DSTO
SECURITY
DOWNGRADING
APPROVAL

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, CANBERRA, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO....
No limitations

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE [X] UNRESTRICTED OR [ ] AS FOR 13a.

14. DESCRIPTORS

Image processing
Data smoothing
Least squares method
Fourier transformation
SMOOFF computer program
Spate 8000 stress analyzer

15. DRDA SUBJECT CATEGORIES
0062F

16. ABSTRACT The FORTRAN program SMOOFF (SMOOthing by a Filtered Fourier transform) smooths a "noisy" function of two variables on a rectangular domain, by minimizing an appropriate functional subject to a least squares constraint.

PAGE CLASSIFICATION
UNCLASSIFIED
PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT.)		
17. IMPRINT AERONAUTICAL RESEARCH LABORATORY, MELBOURNE		
18. DOCUMENT SERIES AND NUMBER AIRCRAFT STRUCTURES TECHNICAL MEMORANDUM 490	19. COST CODE 231090	20. TYPE OF REPORT AND PERIOD COVERED
21. COMPUTER PROGRAMS USED NCAR		
22. ESTABLISHMENT FILE REF.(S)		
23. ADDITIONAL INFORMATION (AS REQUIRED)		