22
8-7-79
24 Ø NTIS

ORNL/CSD/TM-67

Fortran Subroutines for Bicubic Spline Interpolation

P. W. Gaffney

MASTER

OAK RIDGE NATIONAL LABORATORY

OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $4.50; Microfiche $3.00

ORNL/CSD/TM-67

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

FORTRAN SUBROUTINES FOR BICUBIC SPLINE INTERPOLATION

P. W. Gaffney

Sponsor: R. H. Fowler
Originator: P. W. Gaffney

Date Published - July 1979

NOTICE

This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

UNION CARBIDE CORPORATION, NUCLEAR DIVISION
Operating the
Oak Ridge Gaseous Diffusion Plant . Oak Ridge National Laboratory
Oak Ridge Y-12 Plant . Paducah Gaseous Diffusion Plant
for the
DEPARTMENT OF ENERGY

TABLE OF CONTENTS

ACKNOWLEDGMENTS

Support for this work was provided by the Fusion Energy Division and the Computing Applications Department, Computer Sciences Division.

FORTRAN SUBROUTINES FOR BICUBIC SPLINE INTERPOLATION

P. W. Gaffney

ABSTRACT

We present FORTRAN subroutines for computing a bicubic spline function which interpolates values of a function u on a rectangular grid.

1. INTRODUCTION

The purpose of this paper is to present FORTRAN subroutines for computing a bicubic spline function S which interpolates values of a function u on a rectangular grid. Specifically, given mn values

$$u_{v,w} = u(x_v, y_w) \qquad \begin{array}{l} v = 1,\dots,n \\ w = 1,\dots,m \end{array} \tag{1.1}$$

of a function u on the rectangular grid

$$R: = \begin{array}{l} x_1 < x_2 < \dots < x_{n-1} < x_n \\ y_1 < y_2 < \dots < y_{m-1} < y_m \end{array} , \tag{1.2}$$

the routines compute the linear parameters of a bicubic spline function S which satisfies the interpolation conditions

$$S(x_v, y_w) = u_{v,w} \qquad \begin{array}{l} v = 1,\dots,n \\ w = 1,\dots,m \end{array} . \tag{1.3}$$

The routines implement the algorithm of de Boor[1] for computing the coefficients of the spline S. FORTRAN subroutines for this calculation were published by Smith and Gaffney.[2] However, in order to increase the speed and efficiency of these earlier codes, we have completely rewritten the subroutines. In Section 2, we describe briefly the calculations performed by the new subroutines. In Section 3, we present a logical flowchart which describes the calling sequence of the subroutines DERIVS, COEFF and BICUBE. In Section 4, we give an example of a situation where the subroutines may be used. In Section 5, we describe the standards that these subroutines adhere to. In Section 6, we present FORTRAN listings of the single precision version of the codes.

2. SUBROUTINES DERIVS, COEFF AND BICUBE

In this section we describe the algorithm used by the new subroutines. We begin by stating the following theorem, which was proved by de Boor[1] and which forms the basis of the algorithm.

Theorem 1

Given the values

$$u_{v,w} = \hat{S}(x_v, y_w) \qquad v = 1,\ldots,n;\; w = 1,\ldots,m$$

$$p_{v,w} = \frac{\partial \hat{S}}{\partial x}\Big|_{x = x_v, y = y_w} \qquad v = 1,n;\; w = 1,\ldots,m$$

$$q_{v,w} = \frac{\partial \hat{S}}{\partial y}\Big|_{x = x_v, y = y_w} \qquad v = 1,\ldots,n;\; w = 1,m \tag{2.1}$$

and

$$s_{v,w} = \frac{\partial^2 \hat{S}}{\partial x \partial y}\Big|_{x = x_v, y = y_w} \qquad v = 1,n;\; w = 1,m\ ,$$

then there exists exactly one piecewise bicubic polynomial $\hat{S}$ which satisfies conditions (2.1).

In order to compute the bicubic spline S, which is the subject of this paper, we compute approximations to the derivatives in (2.1) using an appropriate four point forward (or backward) difference formula. This part of the calculation is performed in subroutine DERIVZ which is called by the driver routine DERIVS. Subroutine DERIVZ also uses cubic spline interpolation to compute approximations to the derivatives $p_{v,w}$, $q_{v,w}$, $s_{v,w}$ at the interior grid points (x_v,y_w), $v = 2,\ldots, n-1$, $w = 2,\ldots,m-1$, of the mesh (1.2). The derivatives are used by subroutine COEFF in order to compute the coefficients of the spline S. Specifically, it follows from Theorem 1 that within each subrectangle

$$R_{ij} := \begin{cases} x_{i-1} \leq x < x_i & 2 \leq i \leq n \\ y_{j-1} \leq y < y_j & 2 \leq j \leq m \end{cases} \tag{2.2}$$

the spline function S is a bicubic polynomial of the form

$$S_{ij}(x,y) = \sum_{r,t=0}^{3} C_{r,t}^{ij} (x-x_{i-1})^r (y-y_{j-1})^t , \tag{2.3}$$

whose coefficients $C_{r,t}^{ij}$ are uniquely determined from the 16 conditions

$$\left.\begin{aligned}
&S_{ij}(x_v, y_w) = u_{v,w} \\
&\left.\frac{\partial S_{ij}}{\partial x}\right|_{x=x_v, y=y_w} = p_{v,w} \\
&\left.\frac{\partial S_{ij}}{\partial y}\right|_{x=x_v, y=y_w} = q_{v,w} \\
&\left.\frac{\partial^2 S_{ij}}{\partial x \partial y}\right|_{x=x_v, y=y_w} = s_{v,w}
\end{aligned}\right\} \quad \begin{aligned} v &= i-1, i \\ w &= j-1, j \end{aligned} \quad . \tag{2.4}$$

It is straightforward to prove[1] that the matrix, C^{ij}, of coefficients in (2.3) is given by the matrix equation

$$C^{ij} = A(x_i - x_{i-1}) K_{ij} A^T (y_j - y_{j-1}) \tag{2.5}$$

where K_{ij} is the 4×4 matrix

$$K_{ij} = \begin{bmatrix} u_{i-1,j-1} & q_{i-1,j-1} & u_{i-1,j} & q_{i-1,j} \\ p_{i-1,j-1} & s_{i-1,j-1} & p_{i-1,j} & s_{i-1,j} \\ u_{i,j-1} & q_{i,j-1} & u_{i,j} & q_{i,j} \\ p_{i,j-1} & s_{i,j-1} & p_{i,j} & s_{i,j} \end{bmatrix} , \tag{2.6}$$

$A(h)$ is the matrix

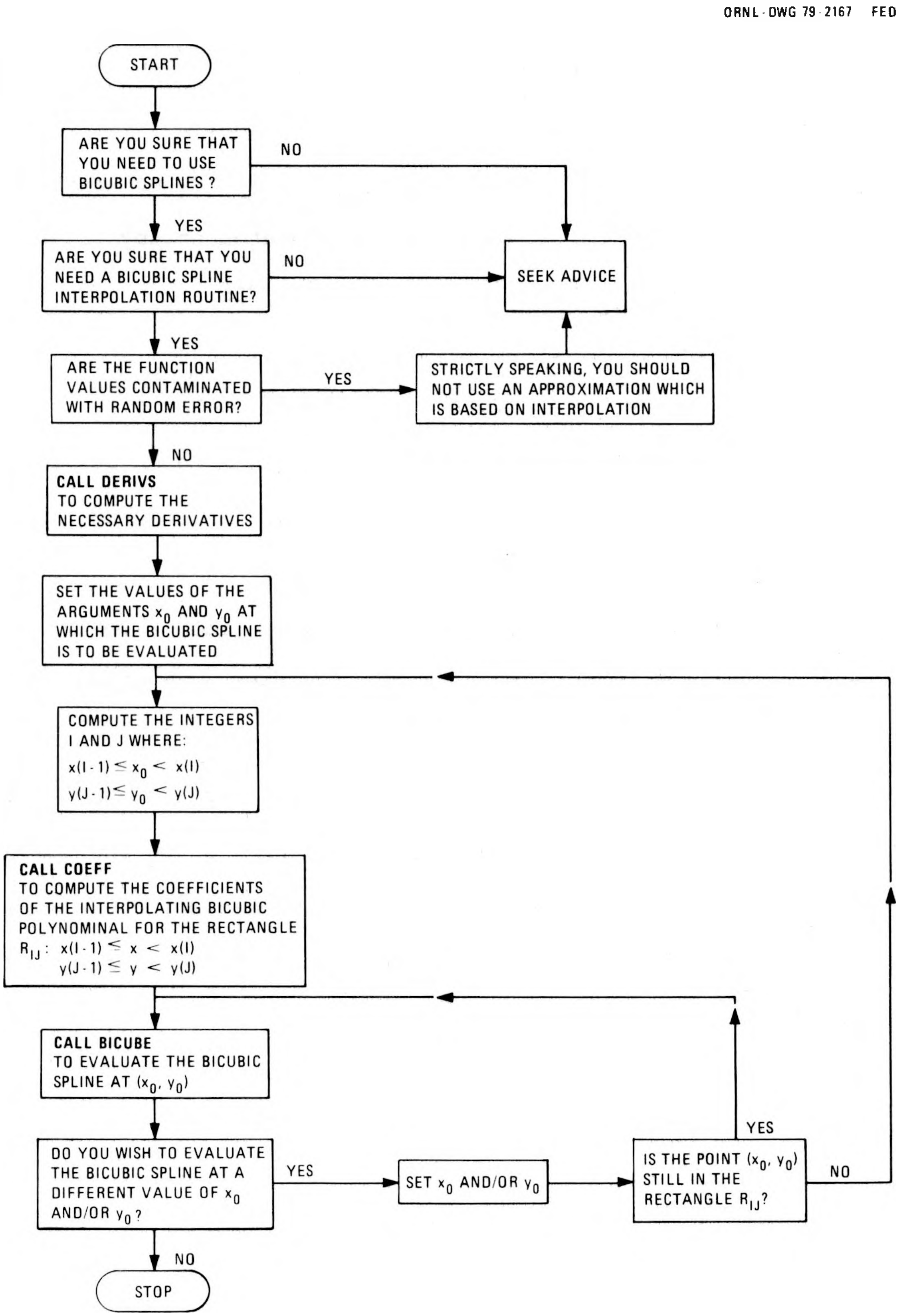
$$A(h) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/h^2 & -2/h & 3/h^2 & -1/h \\ 2/h^3 & 1/h^2 & -2/h^3 & 1/h^2 \end{bmatrix} , \tag{2.7}$$

and A^T denotes the transpose of A. Therefore, subroutine COEFF uses Eqs. (2.5)-(2.7) to compute the coefficients C^{ij} of the spline S for a given rectangle R_{ij}.

When the coefficients C^{ij} have been calculated, the value of S and its derivatives through order 2 may be computed at any value of x and y in the rectangle R_{ij} by using subroutine BICUBE.

3. LOGICAL FLOWCHART

In this section we present a flowchart which describes the calling sequence of the subroutines DERIVS, COEFF and BICUBE, for a given sequence of values $\{u_{vw}\}$. Prospective users are advised to consult the flowchart before incorporating the subroutines into a FORTRAN program. In particular we draw attention to the fact that subroutine DERIVS is only called once to compute the necessary derivatives. This is fortunate because calculating these derivatives is the most time-consuming part of the calculation. Therefore, since the computations performed by COEFF and BICUBE are usually accomplished rapidly, it follows that the computation of the bicubic spline S(x) for a number of values of x is correspondingly fast.



LOGICAL FLOWCHART

4. SAMPLE PROGRAM AND OUTPUT

In this section, we give an example of a situation where the subroutines DERIVS, COEFF and BICUBE may be used.

Suppose we are given values $u_{i,j}$, $i=1,\ldots,n$, $j=1,\ldots,m$ of a function $u(x,y)$ on the rectangular grid

$$x_1 < x_2 < \ldots < x_{n-1} < x_n$$

$$y_1 < y_2 < \ldots < y_{m-1} < y_m ,$$

and we wish to approximate $u(x,y)$ by a bicubic spline function $s(x,y)$ which satisfies the conditions

$$s(x_i,y_j) = u_{i,j} \qquad \begin{array}{l} i=1,\ldots,n \\ j=1,\ldots,m \end{array} .$$

Furthermore, suppose that the eventual aim of the approximation is to tabulate $s(x,y)$ and its derivatives ds/dx and d^2s/dy^2, at values of the arguments x and y. Then, the Fortran code which is required to tabulate these quantities might be as shown in Figure 1. In this case the values $u_{i,j}$ have been generated from the simple function $u(x,y) = yx^3$ and output is required at the points

$$y = 1.0$$

$$x = (i-1)/100 , \quad i=1,\ldots,31 .$$

The results from this program are given in Figure 2.

Figure 1. Sample Program

```
      REAL X(10), Y(20), U(30,35), UX(10,20), UY(10,20),
     *     UXY(10,20), WK(120), C(4,4)
C
C NOTE THAT IN THE ABOVE STATEMENT THE ARRAYS UX,UY, AND UXY HAVE
C TO BE DIMENSIONED N BY M
C
C SET THE NUMBER OF DATA POINTS IN THE X DIRECTION
C
      N = 10
C
C SET THE NUMBER OF DATA POINTS IN THE Y DIRECTION
C
      M = 20
C
C SET THE X DATA POINTS. (NOTE THAT THEY ARE NOT EQUALLY SPACED)
C
      DO 10 I=1,4
            X(I) = FLOAT(I-1)*0.2
            X(I+6) = 0.8 + FLOAT(I)*0.05
   10 CONTINUE
      X(5) = 0.7
      X(6) = 0.8
C
C SET THE Y DATA POINTS. (NOTE THAT THEY DO NOT HAVE TO BE EQUALLY
C                         SPACED. ONLY THE FIRST 4 AND THE LAST 4
C                         X AND Y DATA POINTS MUST BE EQUALLY SPACED)
C
      DO 20 I=1,M
            Y(I) = FLOAT(I-1)/19.0
   20 CONTINUE
C
C SET THE FUNCTION VALUES U(X(I),Y(J))  (IN THIS CASE
C U(X,Y)=Y*X**3)
C
      DO 40 J=1,M
            YJ = Y(J)
            DO 30 I=1,N
                  XI = X(I)
                  U(I,J) = YJ*XI**3
   30       CONTINUE
   40 CONTINUE
C
C SET THE LEADING DIMENSION OF U
C
      LDU = 30
C
C SET THE LENGTH OF THE WORKSPACE ARRAY WK
C
      IWK = 120
C
C COMPUTE THE DERIVATIVES USING DERIVS (NOTE THAT DERIVS IS ONLY
```

```
C                                         CALLED ONCE)
C
      CALL DERIVS(N, M, X, Y, LDU, U, IWK, WK, UX, UY, UXY,
     *     IFAIL)
      IF (IFAIL.GT.0) GO TO 110
C
C SET THE VALUES Y0,J,HY,AND YJM1
C
      Y0 = 1.0
      J = 2
   50 IF (Y0.LT.Y(J)) GO TO 60
      J = J + 1
      IF (J.LE.M) GO TO 50
      J = M
   60 HY = Y(J) - Y(J-1)
      YJM1 = Y(J-1)
C
C SET THE VALUES X0,I,HX,AND XIM1
C
      WRITE (6,99999)
      ILAST = 1
      DO 100 II=1,31
           X0 = FLOAT(II-1)*0.01
           I = ILAST
   70      IF (X0.LT.X(I)) GO TO 80
           I = I + 1
           IF (I.LE.N) GO TO 70
           I = N
C
C NOTE THAT THE COEFFICIENTS ARE NOT RECOMPUTED IF THE POINT
C (X0,Y0) IS STILL IN THE RECTANGLE R(I,J)
C
   80      IF (I.EQ.ILAST) GO TO 90
           HX = X(I) - X(I-1)
           XIM1 = X(I-1)
C
C COMPUTE THE COEFFICIENTS USING COEFF
C
           CALL COEFF(N, M, LDU, U, UX, UY, UXY, I, J, HX, HY,
     *          C, IFAIL)
           IF (IFAIL.GT.0) GO TO 110
C
C COMPUTE THE BICUBIC SPLINE AND ITS DERIVATIVES AT X0,Y0
C
   90      CALL BICUBE(C, XIM1, YJM1, X0, Y0, S, SX, SY, SXY,
     *          SXX, SYY, IFAIL)
           IF (IFAIL.GT.0) GO TO 110
           WRITE (6,99998) X0, S, SX, SYY
           ILAST = I
  100 CONTINUE
  110 STOP
99999 FORMAT (10X, 2HX0, 17X, 1HS, 16X, 2HSX, 16X, 3HSYY)
99998 FORMAT (4(4X, E14.6))
      END
```

Figure 2. Sample Output

XO	S	SX	SYY
0.000000E+00	0.000000E+00	0.310441E-08	0.000000E+00
0.100000E-01	0.100003E-05	0.300003E-03	-0.161977E-07
0.200000E-01	0.800005E-05	0.120000E-02	-0.242362E-07
0.300000E-01	0.270001E-04	0.270000E-02	-0.256003E-07
0.400000E-01	0.640001E-04	0.480000E-02	-0.217746E-07
0.500000E-01	0.125000E-03	0.750000E-02	-0.142438E-07
0.600000E-01	0.216000E-03	0.108000E-01	-0.449264E-08
0.700000E-01	0.343000E-03	0.147000E-01	0.599422E-08
0.800000E-01	0.512000E-03	0.192000E-01	0.157321E-07
0.900000E-01	0.729000E-03	0.243000E-01	0.232362E-07
0.100000E+00	0.100000E-02	0.300000E-01	0.270220E-07
0.110000E+00	0.133100E-02	0.363000E-01	0.256047E-07
0.120000E+00	0.172800E-02	0.432000E-01	0.174997E-07
0.130000E+00	0.219700E-02	0.507000E-01	0.122215E-08
0.140000E+00	0.274400E-02	0.588000E-01	-0.247125E-07
0.150000E+00	0.337500E-02	0.675000E-01	-0.617891E-07
0.160000E+00	0.409600E-02	0.768000E-01	-0.111492E-06
0.170000E+00	0.491300E-02	0.867000E-01	-0.175306E-06
0.180000E+00	0.583200E-02	0.972000E-01	-0.254717E-06
0.190000E+00	0.685900E-02	0.108300E+00	-0.351207E-06
0.200000E+00	0.800000E-02	0.120000E+00	0.220636E-06
0.210000E+00	0.926100E-02	0.132300E+00	0.185952E-06
0.220000E+00	0.106480E-01	0.145200E+00	0.152085E-06
0.230000E+00	0.121670E-01	0.158700E+00	0.120926E-06
0.240000E+00	0.138240E-01	0.172800E+00	0.943674E-07
0.250000E+00	0.156250E-01	0.187500E+00	0.743018E-07
0.260000E+00	0.175760E-01	0.202800E+00	0.626213E-07
0.270000E+00	0.196830E-01	0.218700E+00	0.612180E-07
0.280000E+00	0.219520E-01	0.235200E+00	0.719840E-07
0.290000E+00	0.243890E-01	0.252300E+00	0.968116E-07
0.300000E+00	0.270000E-01	0.270000E+00	0.137593E-06

5. QUALITY ASSURANCE AND SOFTWARE STANDARD

The subroutines were written to conform to the FORTRAN IV ANSI standard 1966, and they have been verified using the Bell Telephone Laboratories Fortran verifier, PFORT[3].

The subroutines have run successfully on a variety of test problems, and they have been analyzed for errors using the University of Colorado's DAVE[4] (a Validation Error Detection and Documentation) system.

To make the subroutines easier to read they have been reformatted using the University of Colorado's text editor POLISH[5].

6. THE FORTRAN LISTINGS OF SUBROUTINES DERIVS, COEFF AND BICUBE

```
      SUBROUTINE DERIVS(N, M, X, Y, LDU, U, IWK, WK, UX, UY,
     *     UXY, IFAIL)
      INTEGER N, M, LDU, IWK
      REAL X(N), Y(M), U(LDU,M), WK(IWK), UX(N,M), UY(N,M),
     *     UXY(N,M)
C ****************************************************************
C
C **SUBROUTINE DERIVS**
C
C   THE PURPOSE OF THIS SUBROUTINE IS TO COMPUTE APPROXIMATIONS
C   TO THE DERIVATIVES OF A FUNCTION U(X,Y) GIVEN VALUES OF THE
C   FUNCTION AT THE GRID POINTS OF A RECTANGULAR MESH R:
C
C                     X(1).LE.X.LT.X(N)
C                     Y(1).LE.Y.LT.Y(M).
C
C   THE SUBROUTINE MAY BE USED FOR COMPUTING A BICUBIC SPLINE
C   FUNCTION WHICH INTERPOLATES THE GIVEN DATA. IN THIS
C   CASE IT SHOULD ONLY BE CALLED ONCE FOR EACH MESH R.
C
C **METHOD**
C
C    THE APPROXIMATIONS ARE COMPUTED USING CUBIC SPLINE
C    INTERPOLATION IN THE X AND Y DIRECTIONS.
C
C **NOTE**
C
C   THIS IS A DRIVER SUBROUTINE WHICH CALLS DERIVZ IN ORDER
C   TO COMPUTE THE NECESSARY DERIVATIVES AT THE MESH POINTS,
C   OF THE RECTANGULAR GRID.
C
C **INPUT**
C
C   N   IS THE NUMBER OF DATA POINTS IN THE X-DIRECTION.
C       RESTRICTION: N.GE.4  N IS NOT ALTERED BY THE SUBROUTINE.
C   M   IS THE NUMBER OF DATA POINTS IN THE Y-DIRECTION.
C       RESTRICTION: M.GE.4  M IS NOT ALTERED BY THE SUBROUTINE.
C   X   IS A LINEAR ARRAY WHICH CONTAINS THE N DATA POINTS
C       X(1) TO X(N).
C       RESTRICTIONS:
C                    X(1).LT.X(2).,...,.LT.X(N),
C                    X(2)-X(1) = X(3)-X(2) = X(4)-X(3)
C                    X(N-2)-X(N-3) = X(N-1)-X(N-2) = X(N)-X(N-1)
C       X IS NOT ALTERED BY THE SUBROUTINE.
C   Y   IS A LINEAR ARRAY WHICH CONTAINS THE M DATA POINTS
C       Y(1) TO Y(M).
C       RESTRICTIONS:
C                    Y(1).LT.Y(2).,...,.LT.Y(M),
C                    Y(2)-Y(1) = Y(3)-Y(2) = Y(4)-Y(3)
C                    Y(M-2)-Y(M-3) = Y(M-1)-Y(M-2) = Y(M)-Y(M-1)
C       Y IS NOT ALTERED BY THE SUBROUTINE.
```

```
C   LDU IS THE LEADING DIMENSION OF THE ARRAY U. LDU IS NOT
C       ALTERED BY THE SUBROUTINE.
C   U   IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE GIVEN
C       FUNCTION VALUES  U(X(I),Y(J)) I=1 TO N, J=1 TO M.
C       U IS NOT ALTERED BY THE SUBROUTINE.
C   IWK IS THE LENGTH OF THE WORK ARRAY WK. THE VALUE OF IWK
C       SHOULD BE AT LEAST 6 * MAX(N,M). IWK IS NOT ALTERED
C       BY THE SUBROUTINE.
C   WK  IS A WORK ARRAY OF LENGTH IWK.
C
C **OUTPUT**
C
C   UX,UY, AND UXY ARE TWO-DIMENSIONAL ARRAYS WHICH MUST BE
C   DIMENSIONED N BY M.
C   UX  CONTAINS DU/DX AT THE MESH POINTS
C   UY  CONTAINS DU/DY AT THE MESH POINTS
C   UXY CONTAINS DU/DXDY AT THE MESH POINTS
C   IFAIL IS AN ERROR FLAG. ON OUTPUT FROM THE
C       SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:
C       IFAIL=0  SUCCESSFUL ENTRY
C       IFAIL=1  N OR M IS TOO SMALL
C       IFAIL=2  THE X DATA POINTS ARE NOT IN ASCENDING ORDER
C       IFAIL=3  THE FIRST FOUR X DATA POINTS ARE NOT EQUALLY
C                SPACED.
C       IFAIL=4  THE LAST FOUR X DATA POINTS ARE NOT EQUALLY
C                SPACED.
C       IFAIL=5  THE Y DATA POINTS ARE NOT IN ASCENDING ORDER
C       IFAIL=6  THE FIRST FOUR Y DATA POINTS ARE NOT EQUALLY
C                SPACED.
C       IFAIL=7  THE LAST FOUR Y DATA POINTS ARE NOT EQUALLY
C                SPACED.
C       IFAIL=8  IWK IS NOT .GE. 6*MAX(N,M)
C
C **QUALITY ASSURANCE AND SOFTWARE STANDARD**
C
C   THIS SUBROUTINE HAS BEEN WRITTEN TO CONFORM
C   TO THE FORTRAN IV ANSI STANDARD 1966, AND
C   IT HAS BEEN VERIFIED USING THE BELL
C   TELEPHONE LABORATORIES FORTRAN VERIFIER:
C   PFORT.
C   THE SUBROUTINE HAS RUN SUCCESSFULLY ON
C   A VARIETY OF TEST PROBLEMS, AND IT HAS
C   BEEN ANALYSED FOR ERRORS USING THE
C   DAVE SYSTEM FROM THE UNIVERSITY OF
C   COLORADO.
C
C
C  P.W.GAFFNEY  1ST. MARCH 1979
C
C ******************************************************************
C
C SET THE OUTPUT STREAM FOR DIAGNOSTIC PRINTING
C TO SUPPRESS PRINTING SET NOUT.LT.0
C
      DATA NOUT /6/
```

```
C
C SET SOME CONSTANTS
C
      DATA P5, ONE, FOUR /0.5,1.0,4.0/
C
C CHECK INPUT DATA
C
      IF (N.LT.4 .OR. M.LT.4) GO TO 40
      NM1 = N - 1
      DO 10 I=1,NM1
           IF (X(I).GE.X(I+1)) GO TO 50
   10 CONTINUE
      MM1 = M - 1
      DO 20 I=1,MM1
           IF (Y(I).GE.Y(I+1)) GO TO 80
   20 CONTINUE
C
C IN ORDER TO CHECK THAT THE FIRST FOUR AND THE LAST FOUR X AND Y
C DATA POINTS ARE EQUALLY SPACED WE REQUIRE THE MACHINE PRECISION EPS,
C WHICH WE NOW COMPUTE.
C
      EPS = ONE
   30 EPS = P5*EPS
      EPSP1 = ONE + EPS
      IF (EPSP1.GT.ONE) GO TO 30
      EPS = FOUR*EPS
C
C NOW CHECK FOR EQUALITY
C
      DIFF = X(2) - X(1)
      IF (ABS(X(3)-X(2)-DIFF).GT.EPS .OR. ABS(X(4)-X(3)-DIFF)
     *     .GT.EPS) GO TO 60
      DIFF = X(N) - X(N-1)
      IF (ABS(X(N-1)-X(N-2)-DIFF).GT.EPS .OR.
     *     ABS(X(N-2)-X(N-3)-DIFF).GT.EPS) GO TO 70
      DIFF = Y(2) - Y(1)
      IF (ABS(Y(3)-Y(2)-DIFF).GT.EPS .OR. ABS(Y(4)-Y(3)-DIFF)
     *     .GT.EPS) GO TO 90
      DIFF = Y(M) - Y(M-1)
      IF (ABS(Y(M-1)-Y(M-2)-DIFF).GT.EPS .OR.
     *     ABS(Y(M-2)-Y(M-3)-DIFF).GT.EPS) GO TO 100
C
      ITEMP = 6*MAX0(N,M)
      IF (IWK.LT.ITEMP) GO TO 110
      IFAIL = 0
C
C PARTITION WORKSPACE ARRAY
C N1=MAX(N,M)
C WK(I)=H(I)
C WK(N1+I)=TH(I)
C WK(2*N1+I)=T(I)
C WK(3*N1+I)=ALPHA(I)
C WK(4*N1+I)=BETAS(I)
C WK(5*N1+I)=BETA(I)
C
```

```
      N1 = MAXO(N,M)
      N2 = N1 + 1
      N3 = N2 + N1
      N4 = N3 + N1
      N5 = N4 + N1
      N6 = N5 + N1
      CALL DERIVZ(N, M, X, Y, LDU, U, N1, WK(1), WK(N2),
     *     WK(N3), WK(N4), WK(N5), WK(N6), UX, UY, UXY)
      GO TO 120
C
C DIAGNOSTIC PRINTING
C
   40 IF (NOUT.GE.0) WRITE (NOUT,99999)
      IFAIL = 1
      GO TO 120
   50 IF (NOUT.GE.0) WRITE (NOUT,99998)
      IFAIL = 2
      GO TO 120
   60 IF (NOUT.GE.0) WRITE (NOUT,99997)
      IFAIL = 3
      GO TO 120
   70 IF (NOUT.GE.0) WRITE (NOUT,99996)
      IFAIL = 4
      GO TO 120
   80 IF (NOUT.GE.0) WRITE (NOUT,99995)
      IFAIL = 5
      GO TO 120
   90 IF (NOUT.GE.0) WRITE (NOUT,99994)
      IFAIL = 6
      GO TO 120
  100 IF (NOUT.GE.0) WRITE (NOUT,99993)
      IFAIL = 7
      GO TO 120
  110 IF (NOUT.GE.0) WRITE (NOUT,99992)
      IFAIL = 8
  120 RETURN
C
C
C
C **END OF SUBROUTINE DERIVS**
C
99999 FORMAT (23H   MESSAGE FROM DERIVS,/19H   THE VALUE OF N O,
     *     16HR M IS TOO SMALL)
99998 FORMAT (23H   MESSAGE FROM DERIVS,/19H   THE DATA POINTS ,
     *     40HX(I),I=1 TO N ARE NOT IN ASCENDING ORDER)
99997 FORMAT (23H   MESSAGE FROM DERIVS,/12H   THE FIRST,
     *     42H FOUR X DATA POINTS ARE NOT EQUALLY SPACED)
99996 FORMAT (23H   MESSAGE FROM DERIVS,/18H   THE LAST FOUR X,
     *     35H DATA POINTS ARE NOT EQUALLY SPACED)
99995 FORMAT (23H   MESSAGE FROM DERIVS,/19H   THE DATA POINTS ,
     *     40HY(I),I=1 TO M ARE NOT IN ASCENDING ORDER)
99994 FORMAT (23H   MESSAGE FROM DERIVS,/19H   THE FIRST FOUR Y,
     *     35H DATA POINTS ARE NOT EQUALLY SPACED)
99993 FORMAT (23H   MESSAGE FROM DERIVS,/18H   THE LAST FOUR Y,
     *     35H DATA POINTS ARE NOT EQUALLY SPACED)
```

```
99992 FORMAT (23H   MESSAGE FROM DERIVS,/19H   THE VALUE OF IWK,
     *      23H IS NOT .GE. 6*MAX(N,M))
      END
      SUBROUTINE DERIVZ(N, M, X, Y, LDU, U, N1, H, TH, T,
     *     ALPHA, BETA, BETAS, P, Q, S)
      INTEGER N, M
      REAL X(N), Y(M), U(LDU,M), H(N1), TH(N1), T(N1),
     *     ALPHA(N1), BETA(N1), BETAS(N1), P(N,M), Q(N,M),
     *     S(N,M)
C
C FIRST COMPUTE THE BOUNDARY CONDITIONS:
C
C     P(V,W) = DU/DX AT X=X(V),Y=Y(W)  V=1 AND N, W=1 TO M
C
C     Q(V,W) = DU/DY AT X=X(V),Y=Y(W)  V=1 TO N, W=1 AND M
C
C     S(V,W) = D2U/DXDY AT X=X(V),Y=Y(W)  V=1 AND N, W=1 AND M
C
C BY APPROXIMATING DERIVATIVES WITH THE FOUR-POINT FORMULA:
C
C     (-11*F1 + 18*F2 - 9*F3 + 2*F4)/(6*H)
C
C WHICH IS OBTAINED BY DIFFERENTIATING THE LAGRANGE POLYNOMIAL
C OF DEGREE 3 THAT PASSES THROUGH THE VALUES F1,F2,F3,F4.
C NOTE THAT THIS FORMULA ASSUMES THAT THE FOUR DATA POINTS
C ASSOCIATED WITH F1,F2,F3,F4, ARE EQUALLY SPACED.
C IF IT IS INCONVENIENT TO USE A FORMULA THAT IS BASED ON
C FOUR EQUALLY SPACED DATA POINTS THEN THE ALTERNATIVE
C FORMULA
C              A*F1 + B*F2 + C*F3 + D*F4
C MAY BE USED, WHERE:
C
C     A = 1/(X1-X2) + 1/(X1-X3) + 1/(X1-X4)
C
C     B = (X1-X3)(X1-X4)/(X2-X1)(X2-X3)(X2-X4)
C
C     C = (X1-X2)(X1-X4)/((X3-X1)(X3-X2)(X3-X4))
C
C     D = (X1-X2)(X1-X3)/((X4-X1)(X4-X2)(X4-X3))
C
C AND X1,X2,X3,X4 ARE THE DATA POINTS ASSOCIATED
C WITH F1,F2,F3,F4.
C
C **NOTE THAT IF SOME OR ALL OF THE ABOVE BOUNDARY
C   DERIVATIVES ARE KNOWN IN ADVANCE THEN THEY NEED
C   NOT BE RECOMPUTED. IN THIS CASE THE USER SHOULD
C   INSERT APPROPRIATE STATEMENTS IN THE FOLLOWING
C   CODE TO SKIP THE RELEVANT DERIVATIVE COMPUTATIONS.
C
C SET SOME CONSTANTS
C
      DATA E2, E3, E6, E9, E11, E18 /2.,3.,6.,9.,11.,18./
      E6HX1 = E6*(X(2)-X(1))
      E6HXN = E6*(X(N)-X(N-1))
C
```

```
C COMPUTE THE APPROXIMATION TO DU(X,Y)/DX  AT
C X=X(1), Y=Y(J),J=1,...,M AND
C X=X(N), Y=Y(J),J=1,...,M.
C
      DO 10 J=1,M
           P(1,J) = (-E11*U(1,J)+E18*U(2,J)-E9*U(3,J)+E2*U(4,J))
     *          /E6HX1
           P(N,J) = (E11*U(N,J)-E18*U(N-1,J)+E9*U(N-2,J)-E2*
     *          U(N-3,J))/E6HXN
   10 CONTINUE
      E6HY1 = E6*(Y(2)-Y(1))
      E6HYM = E6*(Y(M)-Y(M-1))
C
C COMPUTE THE APPROXIMATION TO  DU(X,Y)/DY  AT
C X=X(I),I=1,...,N,  Y=Y(1)
C X=X(I),I=1,...,N  Y=Y(M)
C
      DO 20 I=1,N
           Q(I,1) = (-E11*U(I,1)+E18*U(I,2)-E9*U(I,3)+E2*U(I,4))
     *          /E6HY1
           Q(I,M) = (E11*U(I,M)-E18*U(I,M-1)+E9*U(I,M-2)-E2*U(I,
     *          M-3))/E6HYM
   20 CONTINUE
C
C COMPUTE THE APPROXIMATION TO  D2U(X,Y)/DXDY AT THE FOUR CORNERS
C OF THE RECTANGLE.
C
      S(1,1) = (-E11*Q(1,1)+E18*Q(2,1)-E9*Q(3,1)+E2*Q(4,1))/
     *     E6HX1
      S(N,1) = (E11*Q(N,1)-E18*Q(N-1,1)+E9*Q(N-2,1)-E2*Q(N-3,1))
     *     /E6HXN
      S(1,M) = (E11*P(1,M)-E18*P(1,M-1)+E9*P(1,M-2)-E2*P(1,M-3))
     *     /E6HYM
      S(N,M) = (E11*P(N,M)-E18*P(N,M-1)+E9*P(N,M-2)-E2*P(N,M-3))
     *     /E6HYM
C
C END OF COMPUTING BOUNDARY CONDITIONS
C
C
C COMPUTE APPROXIMATIONS TO THE DERIVATIVES OF U, AT THE INTERIOR
C GRID POINTS OF THE MESH, USING CUBIC SPLINE INTERPOLATION.
C
C SET UP THE BIDIAGONAL MATRIX, WHICH IS THE RESULT OF PERFORMING
C GAUSSIAN ELIMINATION ON THE TRIDIAGONAL MATRIX WHOSE ELEMENTS
C ARE FUNCTIONS OF X.
C ALPHA=DIAGONAL
C H=OFF-DIAGONAL
C AT THE SAME TIME STORE THE QUANTITIES H(I-1)/H(I),I=2,...,N-1
C WHICH ARE REQUIRED FOR SETTING UP THE RIGHT HAND SIDE, AND ALSO
C STORE THE QUANTITIES H(I)/ALPHA(I-1),I=3,...,N-1, WHICH
C APPEAR IN THE FORWARD AND BACKWARD SUBSTITUTION.
C
      H(1) = X(2) - X(1)
      H(2) = X(3) - X(2)
      TH(2) = H(1)/H(2)
```

```
      ALPHA(2) = E2*(X(3)-X(1))
C
      NM1 = N - 1
      DO 30 I=3,NM1
          IP1 = I + 1
          IM1 = I - 1
          H(I) = X(IP1) - X(I)
          TH(I) = H(IM1)/H(I)
          T(I) = H(I)/ALPHA(IM1)
          ALPHA(I) = E2*(X(IP1)-X(IM1)) - H(I-2)*T(I)
   30 CONTINUE
C
      DO 60 J=1,M
C
C SET UP THE TRANSFORMED RIGHT HAND SIDE IN PREPARATION
C FOR COMPUTING DU/DX IN P
C
C BETA=TRANSFORMED RIGHT HAND SIDE
C
          BETA(2) = E3*(TH(2)*(U(3,J)-U(2,J))+(U(2,J)-U(1,J))/
     *        TH(2)) - H(2)*P(1,J)
          DO 40 I=3,NM1
              BETA(I) = E3*(TH(I)*(U(I+1,J)-U(I,J))+(U(I,J)
     *            -U(I-1,J))/TH(I)) - BETA(I-1)*T(I)
   40     CONTINUE
          BETA(NM1) = BETA(NM1) - H(N-2)*P(N,J)
C
C BEGIN BACK SUBSTITUTION FOR P
C
          P(NM1,J) = BETA(NM1)/ALPHA(NM1)
          NM3 = N - 3
          DO 50 IB=1,NM3
              I = NM1 - IB
              P(I,J) = (BETA(I)-H(I-1)*P(I+1,J))/ALPHA(I)
   50     CONTINUE
C END OF COMPUTING DU/DX IN P
   60 CONTINUE
C
C SET UP THE TRANSFORMED RIGHT HAND SIDE IN PREPARATION
C FOR COMPUTING D2U/DXDY IN S ALONG THE LINES Y=Y(1) AND Y=Y(M)
C
      MM1 = M - 1
      DO 90 J=1,M,MM1
          BETA(2) = E3*(TH(2)*(Q(3,J)-Q(2,J))+(Q(2,J)-Q(1,J))/
     *        TH(2)) - H(2)*S(1,J)
          DO 70 I=3,NM1
              BETA(I) = E3*(TH(I)*(Q(I+1,J)-Q(I,J))+(Q(I,J)
     *            -Q(I-1,J))/TH(I)) - BETA(I-1)*T(I)
   70     CONTINUE
          BETA(NM1) = BETA(NM1) - H(N-2)*S(N,J)
C
C BEGIN BACK SUBSTITUTION FOR S
C
          S(NM1,J) = BETA(NM1)/ALPHA(NM1)
          NM3 = N - 3
```

```
          DO 80 IB=1,NM3
               I = NM1 - IB
               S(I,J) = (BETA(I)-H(I-1)*S(I+1,J))/ALPHA(I)
   80     CONTINUE
C END OF COMPUTING D2U/DXDY ALONG Y=Y(1) AND Y=Y(M)
   90 CONTINUE
C
C SIMILARLY SET UP THE BIDIAGONAL MATRIX, WHICH IS THE RESULT OF
C PERFORMING GAUSSIAN ELIMINATION ON THE TRIDIAGONAL MATRIX
C WHOSE ELEMENTS ARE FUNCTIONS OF Y.
      H(1) = Y(2) - Y(1)
      H(2) = Y(3) - Y(2)
      TH(2) = H(1)/H(2)
      ALPHA(2) = E2*(Y(3)-Y(1))
      DO 100 I=3,MM1
           IP1 = I + 1
           IM1 = I - 1
           H(I) = Y(IP1) - Y(I)
           TH(I) = H(IM1)/H(I)
           T(I) = H(I)/ALPHA(IM1)
           ALPHA(I) = E2*(Y(IP1)-Y(IM1)) - H(I-2)*T(I)
  100 CONTINUE
C
      DO 130 I=1,N
C
C SET UP THE TRANSFORMED RIGHT HAND SIDE IN PREPARATION FOR
C COMPUTING DU/DY IN Q
C
          BETA(2) = E3*(TH(2)*(U(I,3)-U(I,2))+(U(I,2)-U(I,1))/
     *         TH(2)) - H(2)*Q(I,1)
          BETAS(2) = E3*(TH(2)*(P(I,3)-P(I,2))+(P(I,2)-P(I,1))/
     *         TH(2)) - H(2)*S(I,1)
          DO 110 J=3,MM1
               BETA(J) = E3*(TH(J)*(U(I,J+1)-U(I,J))+(U(I,J)
     *              -U(I,J-1))/TH(J)) - BETA(J-1)*T(J)
               BETAS(J) = E3*(TH(J)*(P(I,J+1)-P(I,J))+(P(I,J)
     *              -P(I,J-1))/TH(J)) - BETAS(J-1)*T(J)
  110     CONTINUE
          BETA(MM1) = BETA(MM1) - H(M-2)*Q(I,M)
          BETAS(MM1) = BETAS(MM1) - H(M-2)*S(I,M)
C
C BEGIN BACK SUBSTITUTION FOR Q AND THE REMAINING S VALUES
C
          Q(I,MM1) = BETA(MM1)/ALPHA(MM1)
          S(I,MM1) = BETAS(MM1)/ALPHA(MM1)
          MM3 = M - 3
          DO 120 JB=1,MM3
               J = MM1 - JB
               Q(I,J) = (BETA(J)-H(J-1)*Q(I,J+1))/ALPHA(J)
               S(I,J) = (BETAS(J)-H(J-1)*S(I,J+1))/ALPHA(J)
  120     CONTINUE
C END OF COMPUTING DU/DY AND D2U/DXDY
  130 CONTINUE
C
C END OF COMPUTING DERIVATIVES
```

```
C                                                               DER 2180
      RETURN                                                    DER 2190
      END                                                       DER 2200
```

```
      SUBROUTINE COEFF(N, M, LDU, U, UX, UY, UXY, I, J, HX, HY,
     *     C, IFAIL)
      INTEGER N, M, LDU, I, J
      REAL U(LDU,M), UX(N,M), UY(N,M), UXY(N,M), HX, HY, C(4,4)
C ****************************************************************
C
C **SUBROUTINE COEFF**
C
C
C   THIS SUBROUTINE CALCULATES THE 16 COEFFICIENTS OF
C   THE INTERPOLATING BICUBIC POLYNOMIAL FOR THE
C   RECTANGLE R(I,J):
C                     X(I-1).LE.X.LT.X(I)
C                     Y(J-1).LE.Y.LT.Y(J)
C
C **METHOD**
C
C    THE MATRIX OF COEFFICIENTS IS OBTAINED FROM THE
C    EXPRESSION:
C
C                 C = A(HX)*K(I,J)*AT(HY)
C
C    WHERE AT IS THE TRANSPOSE OF THE MATRIX A WHICH
C    TOGETHER WITH K(I,J) IS DEFINED IN THE REPORT:
C    "FORTRAN SUBROUTINES FOR BICUBIC SPLINE INTERPOLATION"
C    BY P/W/GAFFNEY, ORNL/CSD/TM-67.
C
C **INPUT**
C
C   N   IS THE NUMBER OF DATA POINTS IN THE X-DIRECTION.
C       RESTRICTION: N.GE.4  N IS NOT ALTERED BY THE SUBROUTINE.
C   M   IS THE NUMBER OF DATA POINTS IN THE Y-DIRECTION.
C       RESTRICTION: M.GE.4  M IS NOT ALTERED BY THE SUBROUTINE.
C   LDU IS THE LEADING DIMENSION OF THE ARRAY U. LDU IS NOT
C       ALTERED BY THE SUBROUTINE.
C   U   IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE GIVEN
C       FUNCTION VALUES U(X(IR),Y(JR)), IR=1 TO N, JR=1 TO M.
C       U IS NOT ALTERED BY THE SUBROUTINE.
C   UX  IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE
C       DERIVATIVE OF U WITH RESPECT TO X EVALUATED AT
C       THE GRID POINTS X(IR),Y(JR). UX IS NOT ALTERED BY
C       THE SUBROUTINE.
C   UY  IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE
C       DERIVATIVE OF U WITH RESPECT TO Y EVALUATED AT
C       THE GRID POINTS X(IR),Y(JR). UY IS NOT ALTERED BY
C       THE SUBROUTINE.
C   UXY IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE
C       DERIVATIVE OF U WITH RESPECT TO X AND Y EVALUATED
C       AT THE GRID POINTS X(IR),Y(JR).  UXY IS NOT ALTERED BY
C       THE SUBROUTINE.
C   I,J ARE THE INTEGERS WHICH SPECIFY THE RECTANGLE R(I,J).
C       SPECIFICALLY IF   XO.NE.X(N)   THEN I IS THE UNIQUE
C       INTEGER WHICH SATISFIES THE INEQUALITIES
C                     X(I-1).LE.XO.LT.X(I).
C       OTHERWISE I=N.
```

```
C       SIMILARLY IF   YO.NE.Y(M)   THEN J IS THE UNIQUE
C       INTEGER WHICH SATISFIES THE INEQUALITIES
C                     Y(J-1).LE.YO.LT.Y(J).
C       OTHERWISE J=M.
C       RESTRICTION:  2.LE.I.LE.N  AND  2.LE.J.LE.M
C       NEITHER I NOR J IS ALTERED BY THE SUBROUTINE.
C   HX  IS THE VALUE OF  X(I)-X(I-1). IT IS NOT
C       ALTERED BY THE SUBROUTINE.
C   HY  IS THE VALUE OF  Y(J)-Y(J-1). IT IS NOT
C       ALTERED BY THE SUBROUTINE.
C
C **OUTPUT**
C
C   C   IS A 4 BY 4 ARRAY WHICH CONTAINS THE COEFFICIENTS OF THE
C       BICUBIC POLYNOMIAL:
C                       S(XBAR,YBAR) = XBAR*C*YBAR
C
C       WHERE XBAR IS A ROW VECTOR WITH COMPONENTS:
C
C                       XBAR(L) = (X-X(I-1))**(L-1) L=1,2,3,4
C
C       AND YBAR IS A COLUMN VECTOR WITH COMPONENTS:
C
C                       YBAR(L) = (Y-Y(J-1))**(L-1) L=1,2,3,4.
C
C   IFAIL IS AN ERROR FLAG. ON OUTPUT FROM THE
C       SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:
C       IFAIL=0  SUCCESSFUL ENTRY
C       IFAIL=1  N OR M IS TOO SMALL
C       IFAIL=2  I OR J IS EITHER TOO SMALL OR TOO LARGE
C
C **QUALITY ASSURANCE AND SOFTWARE STANDARD**
C
C    THIS SUBROUTINE HAS BEEN WRITTEN TO CONFORM
C    TO THE FORTRAN IV ANSI STANDARD 1966, AND
C    IT HAS BEEN VERIFIED USING THE BELL
C    TELEPHONE LABORATORIES FORTRAN VERIFIER:
C    PFORT.
C    THE SUBROUTINE HAS RUN SUCCESSFULLY ON
C    A VARIETY OF TEST PROBLEMS, AND IT HAS
C    BEEN ANALYSED FOR ERRORS USING THE
C    DAVE SYSTEM FROM THE UNIVERSITY OF
C    COLORADO.
C
C  P.W.GAFFNEY  1ST. MARCH 1979
C
C ****************************************************************
C
C
C  THE FOLLOWING ARRAY IS USED FOR WORKSPACE
      REAL WK(8)
C
C SET THE OUTPUT STREAM FOR DIAGNOSTIC PRINTING
C TO SUPPRESS PRINTING SET NOUT.LT.0
C
```

```
      DATA NOUT /6/
C
C SET SOME CONSTANTS
C
      DATA TWO, THREE /2.,3./
C
C CHECK THE INPUT DATA
C
      IF (N.LT.4 .OR. M.LT.4) GO TO 70
      IF (I.LT.2 .OR. I.GT.N .OR. J.LT.2 .OR. J.GT.M) GO TO 80
      IFAIL = 0
C
C COMPUTE THE MATRIX K(I,J) IN THE ARRAY C
C
      DO 20 IROW=1,3,2
           I1 = I - 1
           IF (IROW.EQ.3) I1 = I
           DO 10 JCOL=1,3,2
                J1 = J - 1
                IF (JCOL.EQ.3) J1 = J
                C(IROW,JCOL) = U(I1,J1)
                C(IROW,JCOL+1) = UY(I1,J1)
                C(IROW+1,JCOL) = UX(I1,J1)
                C(IROW+1,JCOL+1) = UXY(I1,J1)
   10      CONTINUE
   20 CONTINUE
C
C COMPUTE THE COEFFICIENTS IN THE ARRAY C
C
      DO 30 JJ=1,4
           WK(JJ) = (-THREE*C(1,JJ)-TWO*HX*C(2,JJ)+THREE*C(3,JJ)
     *          -HX*C(4,JJ))/(HX*HX)
           WK(JJ+4) = (TWO*C(1,JJ)+HX*C(2,JJ)-TWO*C(3,JJ)+HX*
     *          C(4,JJ))/(HX**3)
   30 CONTINUE
C
      DO 40 JJ=1,4
           C(3,JJ) = WK(JJ)
           C(4,JJ) = WK(JJ+4)
   40 CONTINUE
C
      DO 50 II=1,4
           WK(II) = (-THREE*C(II,1)-TWO*HY*C(II,2)+THREE*C(II,3)
     *          -HY*C(II,4))/(HY*HY)
           WK(II+4) = (TWO*C(II,1)+HY*C(II,2)-TWO*C(II,3)+HY*
     *          C(II,4))/(HY**3)
   50 CONTINUE
C
      DO 60 II=1,4
           C(II,3) = WK(II)
           C(II,4) = WK(II+4)
   60 CONTINUE
      GO TO 90
C
C DIAGNOSTIC PRINTING
```

```
C
   70 IF (NOUT.GE.0) WRITE (NOUT,99999)
      IFAIL = 1
      GO TO 90
   80 IF (NOUT.GE.0) WRITE (NOUT,99998)
      IFAIL = 2
   90 RETURN
C
C
C
C **END OF SUBROUTINE COEFF**
C
99999 FORMAT (22H   MESSAGE FROM COEFF,/20H   THE VALUE OF N OR,
     *     15H M IS TOO SMALL)
99998 FORMAT (22H   MESSAGE FROM COEFF,/20H   THE VALUE OF I OR,
     *     35H J IS EITHER TOO SMALL OR TOO LARGE)
      END
```

```
      SUBROUTINE BICUBE(C, XIM1, YJM1, X0, Y0, S, SX, SY, SXY,
     *     SXX, SYY, IFAIL)
      REAL C(4,4), XIM1, YJM1, X0, Y0, S, SX, SY, SXY, SXX, SYY
C ********************************************************************
C
C **SUBROUTINE BICUBE**
C
C
C    THIS SUBROUTINE EVALUATES A BICUBIC POLYNOMIAL S, AND ITS
C    DERIVATIVES THROUGH ORDER 2. SPECIFICALLY, GIVEN VALUES
C    X=X0 AND Y=Y0 IN THE SUBRECTANGLE:
C            R(I,J): X(I-1).LE.X.LT.X(I)
C                    Y(J-1).LE.Y.LT.Y(J)
C    OF THE GRID WHICH IS DEFINED BY THE VERTICAL LINES X=X(1)
C    TO X=X(N), AND THE HORIZONTAL LINES Y=Y(1) TO Y=Y(M),
C    AND GIVEN THE 16 COEFFICIENTS OF THE BICUBIC INTERPOLATING
C    POLYNOMIAL ASSOCIATED WITH R(I,J), THIS SUBROUTINE COMPUTES
C    THE VALUE:
C                S = S(X0,Y0)
C    AND THE DERIVATIVES:
C               SX = DS/DX
C               SY = DS/DY
C              SXY = DS/DXDY
C              SXX = DDS/DXDX
C              SYY = DDS/DYDY
C    EVALUATED AT THE POINT X0,Y0.
C
C **METHOD**
C
C    THE BICUBIC POLYNOMIAL IS EXPRESSED IN THE FORM:
C                 S(XBAR,YBAR) = XBAR*C*YBAR
C
C    WHERE XBAR IS A ROW VECTOR WITH COMPONENTS:
C
C                 XBAR(L) = (X-X(I-1))**(L-1) L=1 TO 4
C
C    AND YBAR IS A COLUMN VECTOR WITH COMPONENTS:
C
C                 YBAR(L) = (Y-Y(J-1))**(L-1) L=1 TO 4.
C
C    THE RESPECTIVE DERIVATIVES ARE OBTAINED BY DIFFERENTIATING
C    THE COMPONENTS OF THE APPROPRIATE VECTORS XBAR AND YBAR.
C
C    NOTE THAT THIS ROUTINE DOES NOT EXPLICITLY REQUIRE THE
C    VALUE OF I AND J. IT IS ONLY NECESSARY TO SUPPLY THE
C    VALUE OF X(I-1) AND Y(J-1).
C
C **INPUT**
C
C    C      IS A 4 BY 4 ARRAY WHICH CONTAINS THE COEFFICIENTS OF
C           THE BICUBIC POLYNOMIAL S. C IS NOT ALTERED BY THE
C           SUBROUTINE.
C    XIM1   IS THE VALUE OF X(I-1) WHERE I IS DEFINED ABOVE.
C           IF X0=X(N) THEN XIM1=X(N-1). THE VALUE OF XIM1 IS NOT
C           ALTERED BY THE SUBROUTINE.
```

```
C    YJM1  IS THE VALUE OF Y(J-1) WHERE J IS DEFINED ABOVE.
C          IF Y0=Y(M) THEN YJM1=Y(M-1). THE VALUE OF YJM1 IS NOT
C          ALTERED BY THE SUBROUTINE.
C    X0    IS THE VALUE OF THE X-ARGUMENT AT WHICH S IS TO BE
C          EVALUATED. RESTRICTION: THE VALUE OF X0 MUST BE
C          CONSISTENT WITH THE VALUE OF I AS DEFINED ABOVE.
C          X0 IS NOT ALTERED BY THE SUBROUTINE.
C    Y0    IS THE VALUE OF THE Y-ARGUMENT AT WHICH S IS TO BE
C          EVALUATED. RESTRICTION: THE VALUE OF Y0 MUST BE
C          CONSISTENT WITH THE VALUE OF J AS DEFINED ABOVE.
C          Y0 IS NOT ALTERED BY THE SUBROUTINE.
C **OUTPUT**
C
C    S,SX,SY,SXY,SXX,SYY   AS DEFINED ABOVE.
C    IFAIL IS AN ERROR FLAG. ON OUTPUT FROM THE
C          SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:
C          IFAIL=0  SUCCESSFUL ENTRY
C          IFAIL=1  XIM1 OR YJM1 IS NOT CORRECT
C
C **QUALITY ASSURANCE AND SOFTWARE STANDARD**
C
C    THIS SUBROUTINE HAS BEEN WRITTEN TO CONFORM
C    TO THE FORTRAN IV ANSI STANDARD 1966, AND
C    IT HAS BEEN VERIFIED USING THE BELL
C    TELEPHONE LABORATORIES FORTRAN VERIFIER:
C    PFORT.
C    THE SUBROUTINE HAS RUN SUCCESSFULLY ON
C    A VARIETY OF TEST PROBLEMS, AND IT HAS
C    BEEN ANALYSED FOR ERRORS USING THE
C    DAVE SYSTEM FROM THE UNIVERSITY OF
C    COLORADO.
C
C  P.W.GAFFNEY  1ST. MARCH 1979
C
C ****************************************************************
C
C
C THE FOLLOWING ARRAYS ARE USED FOR WORKSPACE
      REAL XX(4), YY(4), DX(4), DY(4), D2X(4), D2Y(4)
C
C SET THE OUTPUT STREAM FOR DIAGNOSTIC PRINTING
C TO SUPPRESS PRINTING SET NOUT.LT.0
C
      DATA NOUT /6/
C
C SET SOME CONSTANTS
C
      DATA ZERO, ONE, TWO, THREE, SIX /0.,1.,2.,3.,6./
C
C CHECK THE INPUT DATA. NOTE THAT, BECAUSE THIS SUBROUTINE
C DOES NOT COMPUTE THE COEFFICIENTS OF THE BICUBIC, IT IS
C LEFT TO THE USER TO CHECK THAT THE ARRAY C CONTAINS THE
C COEFFICIENTS THAT ARE ASSOCIATED WITH R(I,J). MOREOVER
C SINCE THIS SUBROUTINE DOES NOT HAVE ACCESS TO THE
C DATA POINTS X(I),Y(J) OR TO THE QUANTITIES N,M,I,OR J
```

```
C IT IS NOT POSSIBLE TO CHECK THAT X0,Y0,I AND J
C HAVE BEEN DEFINED CORRECTLY BY THE USER.
C
      IF (X0.LT.XIM1 .OR. Y0.LT.YJM1) GO TO 30
      IFAIL = 0
C
C SET THE COMPONENTS OF XBAR AND YBAR AND THEIR DERIVATIVES
C
      XBAR = X0 - XIM1
      YBAR = Y0 - YJM1
C
      XX(1) = ONE
      XX(2) = XBAR
      XX(3) = XBAR*XBAR
      XX(4) = XBAR**3
C
      YY(1) = ONE
      YY(2) = YBAR
      YY(3) = YBAR*YBAR
      YY(4) = YBAR**3
C
      DX(1) = ZERO
      DX(2) = ONE
      DX(3) = TWO*XBAR
      DX(4) = THREE*XBAR*XBAR
      DY(1) = ZERO
      DY(2) = ONE
      DY(3) = TWO*YBAR
      DY(4) = THREE*YBAR*YBAR
C
      D2X(1) = ZERO
      D2X(2) = ZERO
      D2X(3) = TWO
      D2X(4) = SIX*XBAR
      D2Y(1) = ZERO
      D2Y(2) = ZERO
      D2Y(3) = TWO
      D2Y(4) = SIX*YBAR
C
C COMPUTE S AND ITS DERIVATIVES
C
      S = ZERO
      SX = ZERO
      SY = ZERO
      SXY = ZERO
      SXX = ZERO
      SYY = ZERO
      DO 20 JJ=1,4
           YJ1 = YY(JJ)
           DO 10 II=1,4
                X1I = XX(II)
                CIJ = C(II,JJ)
                S = S + X1I*CIJ*YJ1
                SX = SX + DX(II)*CIJ*YJ1
                SY = SY + X1I*CIJ*DY(JJ)
```

```
               SXY = SXY + DX(II)*CIJ*DY(JJ)
               SXX = SXX + D2X(II)*CIJ*YJ1
               SYY = SYY + X1I*CIJ*D2Y(JJ)
   10      CONTINUE
   20 CONTINUE
      GO TO 40
C
C DIAGNOSTIC PRINTING
C
   30 IF (NOUT.GE.0) WRITE (NOUT,99999)
      IFAIL = 1
   40 RETURN
C
C
C
C **END OF SUBROUTINE BICUBE**
C
99999 FORMAT (23H   MESSAGE FROM BICUBE,/19H   XIM1 OR YJM1 HAS,
     *     28H NOT BEEN DEFINED CORRECTLY.)
      END
```

REFERENCES

1. C. de Boor, J. Math and Physics, Vol. 41, pp. 212-218 (1962).

2. P. D. Smith and P. W. Gaffney, "Approximation of the Surface Metric Tensor by Means of Bicubic Spline Interpolation," Royal Aircraft Establishment Technical Report 72185 (1972).

3. B. G. Ryder, "The PFORT Verifier," Software Practice and Experience 4, pp. 359-377 (1974).

4. L. J. Osterweil and L. D. Fosdick, "DAVE - A Validation Error Detection and Documentation System for Fortran Programs," Software Practice and Experience 6, pp. 473-486 (1976).

5. J. Dorrenbacher, D. Paddock, D. Wisneski, and L. D. Fosdick, "POLISH, A Fortran Program to Edit Fortran Programs," University of Colorado, Department of Computer Science Technical Report #CU-CS-050-76 (Revised) May 1976.

ORNL/CSD/TM-67

INTERNAL DISTRIBUTION

1. S. E. Attenberger
2. R. G. Bateman
3. J. D. Callen
4. J. A. Carpenter
5. K. H. Carpenter
6. B. Carreras
7. H. P. Carter/A. A. Brooks/CSD Library
8-9. Central Research Library
10. L. A. Charlton
11. J. S. Crowell
12. L. E. Deleanu
13. Document Reference Section - Y-12
14. R. A. Dory
15. S. K. Fischer
16. R. H. Fowler
17-40. P. W. Gaffney
41. G. E. Giles
42. R. C. Goldfinger
43. W. H. Gray
44. M. T. Heath
45. J. T. Hogan
46. J. T. Holdeman
47. S. K. Iskander
48-50. Laboratory Records
51. Laboratory Records - RC
52. L. B. Maddox
53. D. H. McCollough
54. R. W. McGaffey
55. G. S. McNeilly
56. J. K. Munro
57. M. Murakami
58. B. D. Murphy
59. A. P. Navarro
60. F. W. Nehring
61. C. W. Nestor
62. D. Noid
63. ORNL Patent Office
64. L. W. Owen
65. J. E. Park
66. M. R. Patterson
67. R. O. Sayer
68. D. J. Strickler
69. R. E. Textor
70. W. I. van Rij
71. R. C. Ward
72. G. W. Westley
73. J. C. Whitson
74. R. M. Wieland
75. D. G. Wilson

EXTERNAL DISTRIBUTION

76. Chief, Mathematics and Geoscience Branch, DOE, Washington, DC 20545

77. Office of Assistant Manager for Energy Research and Development, DOE/ORO, P.O. Box E, Oak Ridge, TN 37830

78. J. N. Rogers, Division 8324, Sandia Laboratories, Livermore, CA 94550

79-105. Technical Information Center, P.O. Box 62, Oak Ridge, TN 37830