

**Fortran Subroutines for Bicubic
Spline Interpolation**

P. W. Gaffney

MASTER

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION · FOR THE DEPARTMENT OF ENERGY

22
8-7-79
240715

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

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

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, contractors, subcontractors, or their employees, makes any warranty, express or implied, nor assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, nor represents that its use by such third party would not infringe privately owned rights.

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

FORTTRAN 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

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

...

...

...

...

...

...

TABLE OF CONTENTS

ACKNOWLEDGMENTS	v
ABSTRACT	1
1. INTRODUCTION	1
2. SUBROUTINES DERIVS, COEFF AND BICUBE	2
3. LOGICAL FLOWCHART	5
4. SAMPLE PROGRAM AND OUTPUT	7
5. QUALITY ASSURANCE AND SOFTWARE STANDARD	11
6. THE FORTRAN LISTINGS OF SUBROUTINES DERIVS, COEFF AND BICUBE	12
REFERENCES	29

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

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

of a function u on the rectangular grid

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

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

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

The routines implement the algorithm of de Boor¹ 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 Boor¹ and which forms the basis of the algorithm.

Theorem 1

Given the values

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

$$p_{v,w} = \left. \frac{\partial \hat{S}}{\partial x} \right|_{x = x_v, y = y_w} \quad v = 1, n; w = 1, \dots, m$$

$$q_{v,w} = \left. \frac{\partial \hat{S}}{\partial y} \right|_{x = x_v, y = y_w} \quad v = 1, \dots, n; w = 1, m$$

(2.1)

and

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

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) , $v = 2, \dots, n-1$, $w = 2, \dots, m-1$, 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

$$R_{ij}: = \begin{array}{ll} x_{i-1} \leq x < x_i & 2 \leq i \leq n \\ y_{j-1} \leq y < y_j & 2 \leq j \leq m \end{array} \quad (2.2)$$

the spline function S is a bicubic polynomial of the form

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

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

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

It is straightforward to prove¹ that the matrix, C^{ij} , of coefficients in (2.3) is given by the matrix equation

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

where K_{ij} is the 4×4 matrix

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

$A(h)$ is the matrix

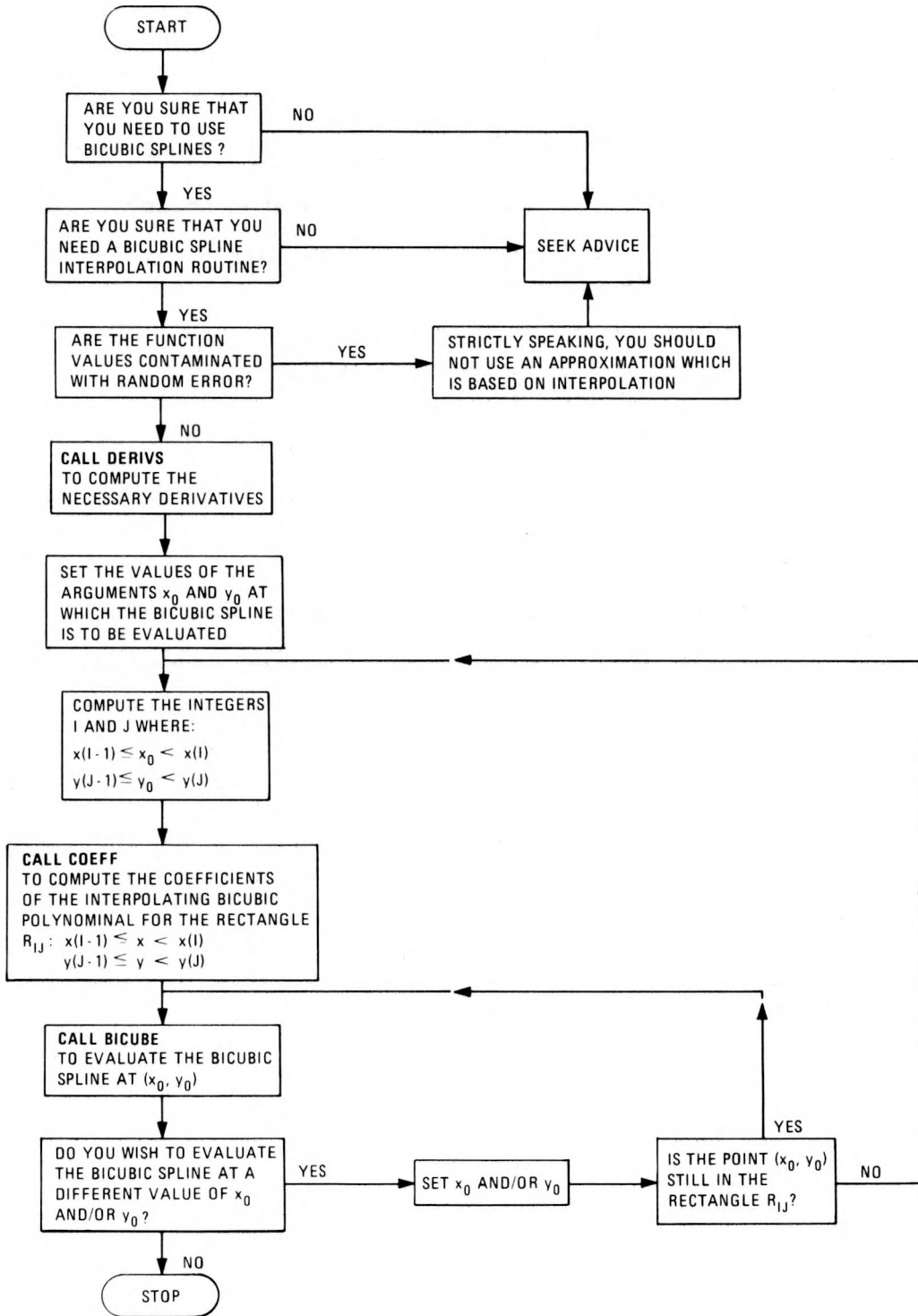
$$A(h) = \begin{bmatrix}
 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 \\
 -3/h^2 & -2/h & 3/h^2 & -1/h \\
 2/h^3 & 1/h^2 & -2/h^3 & 1/h^2
 \end{bmatrix}, \quad (2.7)$$

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 $u_{i,j}$, $i=1,\dots,n$, $j=1,\dots,m$ of a function $u(x,y)$ on the rectangular grid

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

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

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

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

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

$$y = 1.0$$

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

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),	MAN	10
* UXY(10,20), WK(120), C(4,4)	MAN	20
C	MAN	30
C NOTE THAT IN THE ABOVE STATEMENT THE ARRAYS UX,UY, AND UXY HAVE	MAN	40
C TO BE DIMENSIONED N BY M	MAN	50
C	MAN	60
C SET THE NUMBER OF DATA POINTS IN THE X DIRECTION	MAN	70
C	MAN	80
N = 10	MAN	90
C	MAN	100
C SET THE NUMBER OF DATA POINTS IN THE Y DIRECTION	MAN	110
C	MAN	120
M = 20	MAN	130
C	MAN	140
C SET THE X DATA POINTS. (NOTE THAT THEY ARE NOT EQUALLY SPACED)	MAN	150
C	MAN	160
DO 10 I=1,4	MAN	170
X(I) = FLOAT(I-1)*0.2	MAN	180
X(I+6) = 0.8 + FLOAT(I)*0.05	MAN	190
10 CONTINUE	MAN	200
X(5) = 0.7	MAN	210
X(6) = 0.8	MAN	220
C	MAN	230
C SET THE Y DATA POINTS. (NOTE THAT THEY DO NOT HAVE TO BE EQUALLY	MAN	240
C SPACED. ONLY THE FIRST 4 AND THE LAST 4	MAN	250
C X AND Y DATA POINTS MUST BE EQUALLY SPACED)	MAN	260
C	MAN	270
DO 20 I=1,M	MAN	280
Y(I) = FLOAT(I-1)/19.0	MAN	290
20 CONTINUE	MAN	300
C	MAN	310
C SET THE FUNCTION VALUES U(X(I),Y(J)) (IN THIS CASE	MAN	320
C U(X,Y)=Y*X**3)	MAN	330
C	MAN	340
DO 40 J=1,M	MAN	350
YJ = Y(J)	MAN	360
DO 30 I=1,N	MAN	370
XI = X(I)	MAN	380
U(I,J) = YJ*XI**3	MAN	390
30 CONTINUE	MAN	400
40 CONTINUE	MAN	410
C	MAN	420
C SET THE LEADING DIMENSION OF U	MAN	430
C	MAN	440
LDU = 30	MAN	450
C	MAN	460
C SET THE LENGTH OF THE WORKSPACE ARRAY WK	MAN	470
C	MAN	480
IWK = 120	MAN	490
C	MAN	500
C COMPUTE THE DERIVATIVES USING DERIVS (NOTE THAT DERIVS IS ONLY	MAN	510

C	CALLED ONCE)	MAN	520
C		MAN	530
	CALL DERIVS(N, M, X, Y, LDU, U, IWK, WK, UX, UY, UXY,	MAN	540
	* IFAIL)	MAN	550
	IF (IFAIL.GT.0) GO TO 110	MAN	560
C		MAN	570
C	SET THE VALUES YO,J,HY,AND YJM1	MAN	580
C		MAN	590
	YO = 1.0	MAN	600
	J = 2	MAN	610
	50 IF (YO.LT.Y(J)) GO TO 60	MAN	620
	J = J + 1	MAN	630
	IF (J.LE.M) GO TO 50	MAN	640
	J = M	MAN	650
	60 HY = Y(J) - Y(J-1)	MAN	660
	YJM1 = Y(J-1)	MAN	670
C		MAN	680
C	SET THE VALUES XO,I,HX,AND XIM1	MAN	690
C		MAN	700
	WRITE (6,99999)	MAN	710
	ILAST = 1	MAN	720
	DO 100 II=1,31	MAN	730
	XO = FLOAT(II-1)*0.01	MAN	740
	I = ILAST	MAN	750
	70 IF (XO.LT.X(I)) GO TO 80	MAN	760
	I = I + 1	MAN	770
	IF (I.LE.N) GO TO 70	MAN	780
	I = N	MAN	790
C		MAN	800
C	NOTE THAT THE COEFFICIENTS ARE NOT RECOMPUTED IF THE POINT	MAN	810
C	(XO,YO) IS STILL IN THE RECTANGLE R(I,J)	MAN	820
C		MAN	830
	80 IF (I.EQ.ILAST) GO TO 90	MAN	840
	HX = X(I) - X(I-1)	MAN	850
	XIM1 = X(I-1)	MAN	860
C		MAN	870
C	COMPUTE THE COEFFICIENTS USING COEFF	MAN	880
C		MAN	890
	CALL COEFF(N, M, LDU, U, UX, UY, UXY, I, J, HX, HY,	MAN	900
	* C, IFAIL)	MAN	910
	IF (IFAIL.GT.0) GO TO 110	MAN	920
C		MAN	930
C	COMPUTE THE BICUBIC SPLINE AND ITS DERIVATIVES AT XO,YO	MAN	940
C		MAN	950
	90 CALL BICUBE(C, XIM1, YJM1, XO, YO, S, SX, SY, SXY,	MAN	960
	* SXX, SY, IFAIL)	MAN	970
	IF (IFAIL.GT.0) GO TO 110	MAN	980
	WRITE (6,99998) XO, S, SX, SY	MAN	990
	ILAST = I	MAN	1000
	100 CONTINUE	MAN	1010
	110 STOP	MAN	1020
	99999 FORMAT (10X, 2HX, 17X, 1HS, 16X, 2HSX, 16X, 3HSYY)	MAN	1030
	99998 FORMAT (4(4X, E14.6))	MAN	1040
	END	MAN	1050

Figure 2. Sample Output

X0	S	SX	SY
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, PFORT³.

The subroutines have run successfully on a variety of test problems, and they have been analyzed for errors using the University of Colorado's DAVE⁴ (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 POLISH⁵.

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

```

SUBROUTINE DERIVS(N, M, X, Y, LDU, U, IWK, WK, UX, UY,          DER  10
*   UXY, IFAIL)                                              DER  20
INTEGER N, M, LDU, IWK                                       DER  30
REAL X(N), Y(M), U(LDU,M), WK(IWK), UX(N,M), UY(N,M),       DER  40
*   UXY(N,M)                                                 DER  50
C *****                                                    DER  60
C                                                            DER  70
C **SUBROUTINE DERIVS**                                       DER  80
C                                                            DER  90
C THE PURPOSE OF THIS SUBROUTINE IS TO COMPUTE APPROXIMATIONS DER 100
C TO THE DERIVATIVES OF A FUNCTION U(X,Y) GIVEN VALUES OF THE DER 110
C FUNCTION AT THE GRID POINTS OF A RECTANGULAR MESH R:      DER 120
C                                                            DER 130
C                   X(1).LE.X.LT.X(N)                        DER 140
C                   Y(1).LE.Y.LT.Y(M).                      DER 150
C                                                            DER 160
C THE SUBROUTINE MAY BE USED FOR COMPUTING A BICUBIC SPLINE DER 170
C FUNCTION WHICH INTERPOLATES THE GIVEN DATA. IN THIS     DER 180
C CASE IT SHOULD ONLY BE CALLED ONCE FOR EACH MESH R.      DER 190
C                                                            DER 200
C **METHOD**                                                 DER 210
C                                                            DER 220
C THE APPROXIMATIONS ARE COMPUTED USING CUBIC SPLINE       DER 230
C INTERPOLATION IN THE X AND Y DIRECTIONS.                 DER 240
C                                                            DER 250
C **NOTE**                                                  DER 260
C                                                            DER 270
C THIS IS A DRIVER SUBROUTINE WHICH CALLS DERIVZ IN ORDER  DER 280
C TO COMPUTE THE NECESSARY DERIVATIVES AT THE MESH POINTS, DER 290
C OF THE RECTANGULAR GRID.                                  DER 300
C                                                            DER 310
C **INPUT**                                                 DER 320
C                                                            DER 330
C N IS THE NUMBER OF DATA POINTS IN THE X-DIRECTION.     DER 340
C RESTRICTION: N.GE.4 N IS NOT ALTERED BY THE SUBROUTINE. DER 350
C M IS THE NUMBER OF DATA POINTS IN THE Y-DIRECTION.     DER 360
C RESTRICTION: M.GE.4 M IS NOT ALTERED BY THE SUBROUTINE. DER 370
C X IS A LINEAR ARRAY WHICH CONTAINS THE N DATA POINTS   DER 380
C X(1) TO X(N).                                            DER 390
C RESTRICTIONS:                                           DER 400
C                   X(1).LT.X(2),...,LT.X(N),              DER 410
C                   X(2)-X(1) = X(3)-X(2) = X(4)-X(3)      DER 420
C                   X(N-2)-X(N-3) = X(N-1)-X(N-2) = X(N)-X(N-1) DER 430
C X IS NOT ALTERED BY THE SUBROUTINE.                      DER 440
C Y IS A LINEAR ARRAY WHICH CONTAINS THE M DATA POINTS   DER 450
C Y(1) TO Y(M).                                            DER 460
C RESTRICTIONS:                                           DER 470
C                   Y(1).LT.Y(2),...,LT.Y(M),              DER 480
C                   Y(2)-Y(1) = Y(3)-Y(2) = Y(4)-Y(3)      DER 490
C                   Y(M-2)-Y(M-3) = Y(M-1)-Y(M-2) = Y(M)-Y(M-1) DER 500
C Y IS NOT ALTERED BY THE SUBROUTINE.                      DER 510

```

C	LDU IS THE LEADING DIMENSION OF THE ARRAY U. LDU IS NOT	DER	520
C	ALTERED BY THE SUBROUTINE.	DER	530
C	U IS A TWO DIMENSIONAL ARRAY WHICH CONTAINS THE GIVEN	DER	540
C	FUNCTION VALUES U(X(I),Y(J)) I=1 TO N, J=1 TO M.	DER	550
C	U IS NOT ALTERED BY THE SUBROUTINE.	DER	560
C	IWK IS THE LENGTH OF THE WORK ARRAY WK. THE VALUE OF IWK	DER	570
C	SHOULD BE AT LEAST 6 * MAX(N,M). IWK IS NOT ALTERED	DER	580
C	BY THE SUBROUTINE.	DER	590
C	WK IS A WORK ARRAY OF LENGTH IWK.	DER	600
C		DER	610
C	**OUTPUT**	DER	620
C		DER	630
C	UX,UY, AND UXY ARE TWO-DIMENSIONAL ARRAYS WHICH MUST BE	DER	640
C	DIMENSIONED N BY M.	DER	650
C	UX CONTAINS DU/DX AT THE MESH POINTS	DER	660
C	UY CONTAINS DU/DY AT THE MESH POINTS	DER	670
C	UXY CONTAINS DU/DXDY AT THE MESH POINTS	DER	680
C	IFAIL IS AN ERROR FLAG. ON OUTPUT FROM THE	DER	690
C	SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:	DER	700
C	IFAIL=0 SUCCESSFUL ENTRY	DER	710
C	IFAIL=1 N OR M IS TOO SMALL	DER	720
C	IFAIL=2 THE X DATA POINTS ARE NOT IN ASCENDING ORDER	DER	730
C	IFAIL=3 THE FIRST FOUR X DATA POINTS ARE NOT EQUALLY	DER	740
C	SPACED.	DER	750
C	IFAIL=4 THE LAST FOUR X DATA POINTS ARE NOT EQUALLY	DER	760
C	SPACED.	DER	770
C	IFAIL=5 THE Y DATA POINTS ARE NOT IN ASCENDING ORDER	DER	780
C	IFAIL=6 THE FIRST FOUR Y DATA POINTS ARE NOT EQUALLY	DER	790
C	SPACED.	DER	800
C	IFAIL=7 THE LAST FOUR Y DATA POINTS ARE NOT EQUALLY	DER	810
C	SPACED.	DER	820
C	IFAIL=8 IWK IS NOT .GE. 6*MAX(N,M)	DER	830
C		DER	840
C	**QUALITY ASSURANCE AND SOFTWARE STANDARD**	DER	850
C		DER	860
C	THIS SUBROUTINE HAS BEEN WRITTEN TO CONFORM	DER	870
C	TO THE FORTRAN IV ANSI STANDARD 1966, AND	DER	880
C	IT HAS BEEN VERIFIED USING THE BELL	DER	890
C	TELEPHONE LABORATORIES FORTRAN VERIFIER:	DER	900
C	PFORT.	DER	910
C	THE SUBROUTINE HAS RUN SUCCESSFULLY ON	DER	920
C	A VARIETY OF TEST PROBLEMS, AND IT HAS	DER	930
C	BEEN ANALYSED FOR ERRORS USING THE	DER	940
C	DAVE SYSTEM FROM THE UNIVERSITY OF	DER	950
C	COLORADO.	DER	960
C		DER	970
C		DER	980
C	P.W.GAFFNEY 1ST. MARCH 1979	DER	990
C		DER	1000
C	*****	DER	1010
C		DER	1020
C	SET THE OUTPUT STREAM FOR DIAGNOSTIC PRINTING	DER	1030
C	TO SUPPRESS PRINTING SET NOUT.LT.0	DER	1040
C		DER	1050
C	DATA NOUT /6/	DER	1060

C		DER 1070
C	SET SOME CONSTANTS	DER 1080
C		DER 1090
	DATA P5, ONE, FOUR /0.5,1.0,4.0/	DER 1100
C		DER 1110
C	CHECK INPUT DATA	DER 1120
C		DER 1130
	IF (N.LT.4 .OR. M.LT.4) GO TO 40	DER 1140
	NM1 = N - 1	DER 1150
	DO 10 I=1,NM1	DER 1160
	IF (X(I).GE.X(I+1)) GO TO 50	DER 1170
10	CONTINUE	DER 1180
	MM1 = M - 1	DER 1190
	DO 20 I=1,MM1	DER 1200
	IF (Y(I).GE.Y(I+1)) GO TO 80	DER 1210
20	CONTINUE	DER 1220
C		DER 1230
C	IN ORDER TO CHECK THAT THE FIRST FOUR AND THE LAST FOUR X AND Y	DER 1240
C	DATA POINTS ARE EQUALLY SPACED WE REQUIRE THE MACHINE PRECISION EPS,	DER 1250
C	WHICH WE NOW COMPUTE.	DER 1260
C		DER 1270
	EPS = ONE	DER 1280
30	EPS = P5*EPS	DER 1290
	EPSP1 = ONE + EPS	DER 1300
	IF (EPSP1.GT.ONE) GO TO 30	DER 1310
	EPS = FOUR*EPS	DER 1320
C		DER 1330
C	NOW CHECK FOR EQUALITY	DER 1340
C		DER 1350
	DIFF = X(2) - X(1)	DER 1360
	IF (ABS(X(3)-X(2)-DIFF).GT.EPS .OR. ABS(X(4)-X(3)-DIFF)	DER 1370
*	.GT.EPS) GO TO 60	DER 1380
	DIFF = X(N) - X(N-1)	DER 1390
	IF (ABS(X(N-1)-X(N-2)-DIFF).GT.EPS .OR.	DER 1400
*	ABS(X(N-2)-X(N-3)-DIFF).GT.EPS) GO TO 70	DER 1410
	DIFF = Y(2) - Y(1)	DER 1420
	IF (ABS(Y(3)-Y(2)-DIFF).GT.EPS .OR. ABS(Y(4)-Y(3)-DIFF)	DER 1430
*	.GT.EPS) GO TO 90	DER 1440
	DIFF = Y(M) - Y(M-1)	DER 1450
	IF (ABS(Y(M-1)-Y(M-2)-DIFF).GT.EPS .OR.	DER 1460
*	ABS(Y(M-2)-Y(M-3)-DIFF).GT.EPS) GO TO 100	DER 1470
C		DER 1480
	ITEMP = 6*MAX0(N,M)	DER 1490
	IF (IWK.LT.ITEMP) GO TO 110	DER 1500
	IFAIL = 0	DER 1510
C		DER 1520
C	PARTITION WORKSPACE ARRAY	DER 1530
C	N1=MAX(N,M)	DER 1540
C	WK(I)=H(I)	DER 1550
C	WK(N1+I)=TH(I)	DER 1560
C	WK(2*N1+I)=T(I)	DER 1570
C	WK(3*N1+I)=ALPHA(I)	DER 1580
C	WK(4*N1+I)=BETAS(I)	DER 1590
C	WK(5*N1+I)=BETA(I)	DER 1600
C		DER 1610

N1 = MAX0(N,M)	DER 1620
N2 = N1 + 1	DER 1630
N3 = N2 + N1	DER 1640
N4 = N3 + N1	DER 1650
N5 = N4 + N1	DER 1660
N6 = N5 + N1	DER 1670
CALL DERIVZ(N, M, X, Y, LDU, U, N1, WK(1), WK(N2),	DER 1680
* WK(N3), WK(N4), WK(N5), WK(N6), UX, UY, UXY)	DER 1690
GO TO 120	DER 1700
C	DER 1710
C DIAGNOSTIC PRINTING	DER 1720
C	DER 1730
40 IF (NOUT.GE.0) WRITE (NOUT,99999)	DER 1740
IFAIL = 1	DER 1750
GO TO 120	DER 1760
50 IF (NOUT.GE.0) WRITE (NOUT,99998)	DER 1770
IFAIL = 2	DER 1780
GO TO 120	DER 1790
60 IF (NOUT.GE.0) WRITE (NOUT,99997)	DER 1800
IFAIL = 3	DER 1810
GO TO 120	DER 1820
70 IF (NOUT.GE.0) WRITE (NOUT,99996)	DER 1830
IFAIL = 4	DER 1840
GO TO 120	DER 1850
80 IF (NOUT.GE.0) WRITE (NOUT,99995)	DER 1860
IFAIL = 5	DER 1870
GO TO 120	DER 1880
90 IF (NOUT.GE.0) WRITE (NOUT,99994)	DER 1890
IFAIL = 6	DER 1900
GO TO 120	DER 1910
100 IF (NOUT.GE.0) WRITE (NOUT,99993)	DER 1920
IFAIL = 7	DER 1930
GO TO 120	DER 1940
110 IF (NOUT.GE.0) WRITE (NOUT,99992)	DER 1950
IFAIL = 8	DER 1960
120 RETURN	DER 1970
C	DER 1980
C	DER 1990
C	DER 2000
C ***END OF SUBROUTINE DERIVS**	DER 2010
C	DER 2020
99999 FORMAT (23H MESSAGE FROM DERIVS,/19H THE VALUE OF N O,	DER 2030
* 16HR M IS TOO SMALL)	DER 2040
99998 FORMAT (23H MESSAGE FROM DERIVS,/19H THE DATA POINTS ,	DER 2050
* 40HX(I),I=1 TO N ARE NOT IN ASCENDING ORDER)	DER 2060
99997 FORMAT (23H MESSAGE FROM DERIVS,/12H THE FIRST,	DER 2070
* 42H FOUR X DATA POINTS ARE NOT EQUALLY SPACED)	DER 2080
99996 FORMAT (23H MESSAGE FROM DERIVS,/18H THE LAST FOUR X,	DER 2090
* 35H DATA POINTS ARE NOT EQUALLY SPACED)	DER 2100
99995 FORMAT (23H MESSAGE FROM DERIVS,/19H THE DATA POINTS ,	DER 2110
* 40HY(I),I=1 TO M ARE NOT IN ASCENDING ORDER)	DER 2120
99994 FORMAT (23H MESSAGE FROM DERIVS,/19H THE FIRST FOUR Y,	DER 2130
* 35H DATA POINTS ARE NOT EQUALLY SPACED)	DER 2140
99993 FORMAT (23H MESSAGE FROM DERIVS,/18H THE LAST FOUR Y,	DER 2150
* 35H DATA POINTS ARE NOT EQUALLY SPACED)	DER 2160

```

99992 FORMAT (23H MESSAGE FROM DERIVS,/19H THE VALUE OF IWK, DER 2170
* 23H IS NOT .GE. 6*MAX(N,M)) DER 2180
END DER 2190
SUBROUTINE DERIVZ(N, M, X, Y, LDU, U, N1, H, TH, T, DER 10
* ALPHA, BETA, BETAS, P, Q, S) DER 20
INTEGER N, M DER 30
REAL X(N), Y(M), U(LDU,M), H(N1), TH(N1), T(N1), DER 40
* ALPHA(N1), BETA(N1), BETAS(N1), P(N,M), Q(N,M), DER 50
* S(N,M) DER 60
C DER 70
C FIRST COMPUTE THE BOUNDARY CONDITIONS: DER 80
C DER 90
C P(V,W) = DU/DX AT X=X(V),Y=Y(W) V=1 AND N, W=1 TO M DER 100
C DER 110
C Q(V,W) = DU/DY AT X=X(V),Y=Y(W) V=1 TO N, W=1 AND M DER 120
C DER 130
C S(V,W) = D2U/DXDY AT X=X(V),Y=Y(W) V=1 AND N, W=1 AND M DER 140
C DER 150
C BY APPROXIMATING DERIVATIVES WITH THE FOUR-POINT FORMULA: DER 160
C DER 170
C (-11*F1 + 18*F2 - 9*F3 + 2*F4)/(6*H) DER 180
C DER 190
C WHICH IS OBTAINED BY DIFFERENTIATING THE LAGRANGE POLYNOMIAL DER 200
C OF DEGREE 3 THAT PASSES THROUGH THE VALUES F1,F2,F3,F4. DER 210
C NOTE THAT THIS FORMULA ASSUMES THAT THE FOUR DATA POINTS DER 220
C ASSOCIATED WITH F1,F2,F3,F4, ARE EQUALLY SPACED. DER 230
C IF IT IS INCONVENIENT TO USE A FORMULA THAT IS BASED ON DER 240
C FOUR EQUALLY SPACED DATA POINTS THEN THE ALTERNATIVE DER 250
C FORMULA DER 260
C A*F1 + B*F2 + C*F3 + D*F4 DER 270
C MAY BE USED, WHERE: DER 280
C DER 290
C A = 1/(X1-X2) + 1/(X1-X3) + 1/(X1-X4) DER 300
C DER 310
C B = (X1-X3)(X1-X4)/(X2-X1)(X2-X3)(X2-X4) DER 320
C DER 330
C C = (X1-X2)(X1-X4)/((X3-X1)(X3-X2)(X3-X4)) DER 340
C DER 350
C D = (X1-X2)(X1-X3)/((X4-X1)(X4-X2)(X4-X3)) DER 360
C DER 370
C AND X1,X2,X3,X4 ARE THE DATA POINTS ASSOCIATED DER 380
C WITH F1,F2,F3,F4. DER 390
C DER 400
C **NOTE THAT IF SOME OR ALL OF THE ABOVE BOUNDARY DER 410
C DERIVATIVES ARE KNOWN IN ADVANCE THEN THEY NEED DER 420
C NOT BE RECOMPUTED. IN THIS CASE THE USER SHOULD DER 430
C INSERT APPROPRIATE STATEMENTS IN THE FOLLOWING DER 440
C CODE TO SKIP THE RELEVANT DERIVATIVE COMPUTATIONS. DER 450
C DER 460
C SET SOME CONSTANTS DER 470
C DER 480
C DATA E2, E3, E6, E9, E11, E18 /2.,3.,6.,9.,11.,18./ DER 490
C E6HX1 = E6*(X(2)-X(1)) DER 500
C E6HXN = E6*(X(N)-X(N-1)) DER 510
C DER 520

```

C COMPUTE THE APPROXIMATION TO DU(X,Y)/DX AT	DER 530
C X=X(1), Y=Y(J),J=1,...,M AND	DER 540
C X=X(N), Y=Y(J),J=1,...,M.	DER 550
C	DER 560
DO 10 J=1,M	DER 570
P(1,J) = (-E11*U(1,J)+E18*U(2,J)-E9*U(3,J)+E2*U(4,J))	DER 580
* /E6HX1	DER 590
P(N,J) = (E11*U(N,J)-E18*U(N-1,J)+E9*U(N-2,J)-E2*	DER 600
* U(N-3,J))/E6HXN	DER 610
10 CONTINUE	DER 620
E6HY1 = E6*(Y(2)-Y(1))	DER 630
E6HYM = E6*(Y(M)-Y(M-1))	DER 640
C	DER 650
C COMPUTE THE APPROXIMATION TO DU(X,Y)/DY AT	DER 660
C X=X(I),I=1,...,N, Y=Y(1)	DER 670
C X=X(I),I=1,...,N Y=Y(M)	DER 680
C	DER 690
DO 20 I=1,N	DER 700
Q(I,1) = (-E11*U(I,1)+E18*U(I,2)-E9*U(I,3)+E2*U(I,4))	DER 710
* /E6HY1	DER 720
Q(I,M) = (E11*U(I,M)-E18*U(I,M-1)+E9*U(I,M-2)-E2*U(I,	DER 730
* M-3))/E6HYM	DER 740
20 CONTINUE	DER 750
C	DER 760
C COMPUTE THE APPROXIMATION TO D2U(X,Y)/DXDY AT THE FOUR CORNERS	DER 770
C OF THE RECTANGLE.	DER 780
C	DER 790
S(1,1) = (-E11*Q(1,1)+E18*Q(2,1)-E9*Q(3,1)+E2*Q(4,1))/	DER 800
* E6HX1	DER 810
S(N,1) = (E11*Q(N,1)-E18*Q(N-1,1)+E9*Q(N-2,1)-E2*Q(N-3,1))	DER 820
* /E6HXN	DER 830
S(1,M) = (E11*P(1,M)-E18*P(1,M-1)+E9*P(1,M-2)-E2*P(1,M-3))	DER 840
* /E6HYM	DER 850
S(N,M) = (E11*P(N,M)-E18*P(N,M-1)+E9*P(N,M-2)-E2*P(N,M-3))	DER 860
* /E6HYM	DER 870
C	DER 880
C END OF COMPUTING BOUNDARY CONDITIONS	DER 890
C	DER 900
C	DER 910
C COMPUTE APPROXIMATIONS TO THE DERIVATIVES OF U, AT THE INTERIOR	DER 920
C GRID POINTS OF THE MESH, USING CUBIC SPLINE INTERPOLATION.	DER 930
C	DER 940
C SET UP THE BIDIAGONAL MATRIX, WHICH IS THE RESULT OF PERFORMING	DER 950
C GAUSSIAN ELIMINATION ON THE TRIDIAGONAL MATRIX WHOSE ELEMENTS	DER 960
C ARE FUNCTIONS OF X.	DER 970
C ALPHA=DIAGONAL	DER 980
C H=OFF-DIAGONAL	DER 990
C AT THE SAME TIME STORE THE QUANTITIES H(I-1)/H(I),I=2,...,N-1	DER 1000
C WHICH ARE REQUIRED FOR SETTING UP THE RIGHT HAND SIDE, AND ALSO	DER 1010
C STORE THE QUANTITIES H(I)/ALPHA(I-1),I=3,...,N-1, WHICH	DER 1020
C APPEAR IN THE FORWARD AND BACKWARD SUBSTITUTION.	DER 1030
C	DER 1040
H(1) = X(2) - X(1)	DER 1050
H(2) = X(3) - X(2)	DER 1060
TH(2) = H(1)/H(2)	DER 1070

```

      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

```

```

DER 1080
DER 1090
DER 1100
DER 1110
DER 1120
DER 1130
DER 1140
DER 1150
DER 1160
DER 1170
DER 1180
DER 1190
DER 1200
DER 1210
DER 1220
DER 1230
DER 1240
DER 1250
DER 1260
DER 1270
DER 1280
DER 1290
DER 1300
DER 1310
DER 1320
DER 1330
DER 1340
DER 1350
DER 1360
DER 1370
DER 1380
DER 1390
DER 1400
DER 1410
DER 1420
DER 1430
DER 1440
DER 1450
DER 1460
DER 1470
DER 1480
DER 1490
DER 1500
DER 1510
DER 1520
DER 1530
DER 1540
DER 1550
DER 1560
DER 1570
DER 1580
DER 1590
DER 1600
DER 1610
DER 1620

```

```

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

RETURN
END

DER 2180
DER 2190
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      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.

```

```

COE 10
COE 20
COE 30
COE 40
COE 50
COE 60
COE 70
COE 80
COE 90
COE 100
COE 110
COE 120
COE 130
COE 140
COE 150
COE 160
COE 170
COE 180
COE 190
COE 200
COE 210
COE 220
COE 230
COE 240
COE 250
COE 260
COE 270
COE 280
COE 290
COE 300
COE 310
COE 320
COE 330
COE 340
COE 350
COE 360
COE 370
COE 380
COE 390
COE 400
COE 410
COE 420
COE 430
COE 440
COE 450
COE 460
COE 470
COE 480
COE 490
COE 500
COE 510
COE 520
COE 530
COE 540
COE 550

```

C	SIMILARLY IF YO.NE.Y(M) THEN J IS THE UNIQUE	COE 560
C	INTEGER WHICH SATISFIES THE INEQUALITIES	COE 570
C	Y(J-1).LE.YO.LT.Y(J).	COE 580
C	OTHERWISE J=M.	COE 590
C	RESTRICTION: 2.LE.I.LE.N AND 2.LE.J.LE.M	COE 600
C	NEITHER I NOR J IS ALTERED BY THE SUBROUTINE.	COE 610
C	HX IS THE VALUE OF X(I)-X(I-1). IT IS NOT	COE 620
C	ALTERED BY THE SUBROUTINE.	COE 630
C	HY IS THE VALUE OF Y(J)-Y(J-1). IT IS NOT	COE 640
C	ALTERED BY THE SUBROUTINE.	COE 650
C		COE 660
C	**OUTPUT**	COE 670
C		COE 680
C	C IS A 4 BY 4 ARRAY WHICH CONTAINS THE COEFFICIENTS OF THE	COE 690
C	BICUBIC POLYNOMIAL:	COE 700
C	S(XBAR,YBAR) = XBAR*C*YBAR	COE 710
C		COE 720
C	WHERE XBAR IS A ROW VECTOR WITH COMPONENTS:	COE 730
C		COE 740
C	XBAR(L) = (X-X(I-1))**(L-1) L=1,2,3,4	COE 750
C		COE 760
C	AND YBAR IS A COLUMN VECTOR WITH COMPONENTS:	COE 770
C		COE 780
C	YBAR(L) = (Y-Y(J-1))**(L-1) L=1,2,3,4.	COE 790
C		COE 800
C	IFAIL IS AN ERROR FLAG. ON OUTPUT FROM THE	COE 810
C	SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:	COE 820
C	IFAIL=0 SUCCESSFUL ENTRY	COE 830
C	IFAIL=1 N OR M IS TOO SMALL	COE 840
C	IFAIL=2 I OR J IS EITHER TOO SMALL OR TOO LARGE	COE 850
C		COE 860
C	**QUALITY ASSURANCE AND SOFTWARE STANDARD**	COE 870
C		COE 880
C	THIS SUBROUTINE HAS BEEN WRITTEN TO CONFORM	COE 890
C	TO THE FORTRAN IV ANSI STANDARD 1966, AND	COE 900
C	IT HAS BEEN VERIFIED USING THE BELL	COE 910
C	TELEPHONE LABORATORIES FORTRAN VERIFIER:	COE 920
C	PFFT.	COE 930
C	THE SUBROUTINE HAS RUN SUCCESSFULLY ON	COE 940
C	A VARIETY OF TEST PROBLEMS, AND IT HAS	COE 950
C	BEEN ANALYSED FOR ERRORS USING THE	COE 960
C	DAVE SYSTEM FROM THE UNIVERSITY OF	COE 970
C	COLORADO.	COE 980
C		COE 990
C	P.W.GAFFNEY 1ST. MARCH 1979	COE 1000
C		COE 1010
C	*****	COE 1020
C		COE 1030
C		COE 1040
C	THE FOLLOWING ARRAY IS USED FOR WORKSPACE	COE 1050
C	REAL WK(8)	COE 1060
C		COE 1070
C	SET THE OUTPUT STREAM FOR DIAGNOSTIC PRINTING	COE 1080
C	TO SUPPRESS PRINTING SET NOUT.LT.0	COE 1090
C		COE 1100

DATA NOUT /6/	COE 1110
C	COE 1120
C SET SOME CONSTANTS	COE 1130
C	COE 1140
DATA TWO, THREE /2.,3./	COE 1150
C	COE 1160
C CHECK THE INPUT DATA	COE 1170
C	COE 1180
IF (N.LT.4 .OR. M.LT.4) GO TO 70	COE 1190
IF (I.LT.2 .OR. I.GT.N .OR. J.LT.2 .OR. J.GT.M) GO TO 80	COE 1200
IFAIL = 0	COE 1210
C	COE 1220
C COMPUTE THE MATRIX K(I,J) IN THE ARRAY C	COE 1230
C	COE 1240
DO 20 IROW=1,3,2	COE 1250
I1 = I - 1	COE 1260
IF (IROW.EQ.3) I1 = I	COE 1270
DO 10 JCOL=1,3,2	COE 1280
J1 = J - 1	COE 1290
IF (JCOL.EQ.3) J1 = J	COE 1300
C(IROW,JCOL) = U(I1,J1)	COE 1310
C(IROW,JCOL+1) = UY(I1,J1)	COE 1320
C(IROW+1,JCOL) = UX(I1,J1)	COE 1330
C(IROW+1,JCOL+1) = UXY(I1,J1)	COE 1340
10 CONTINUE	COE 1350
20 CONTINUE	COE 1360
C	COE 1370
C COMPUTE THE COEFFICIENTS IN THE ARRAY C	COE 1380
C	COE 1390
DO 30 JJ=1,4	COE 1400
WK(JJ) = (-THREE*C(1,JJ)-TWO*HX*C(2,JJ)+THREE*C(3,JJ)	COE 1410
* -HX*C(4,JJ))/(HX*HX)	COE 1420
WK(JJ+4) = (TWO*C(1,JJ)+HX*C(2,JJ)-TWO*C(3,JJ)+HX*	COE 1430
* C(4,JJ))/(HX**3)	COE 1440
30 CONTINUE	COE 1450
C	COE 1460
DO 40 JJ=1,4	COE 1470
C(3,JJ) = WK(JJ)	COE 1480
C(4,JJ) = WK(JJ+4)	COE 1490
40 CONTINUE	COE 1500
C	COE 1510
DO 50 II=1,4	COE 1520
WK(II) = (-THREE*C(II,1)-TWO*HY*C(II,2)+THREE*C(II,3)	COE 1530
* -HY*C(II,4))/(HY*HY)	COE 1540
WK(II+4) = (TWO*C(II,1)+HY*C(II,2)-TWO*C(II,3)+HY*	COE 1550
* C(II,4))/(HY**3)	COE 1560
50 CONTINUE	COE 1570
C	COE 1580
DO 60 II=1,4	COE 1590
C(II,3) = WK(II)	COE 1600
C(II,4) = WK(II+4)	COE 1610
60 CONTINUE	COE 1620
GO TO 90	COE 1630
C	COE 1640
C DIAGNOSTIC PRINTING	COE 1650

C		COE 1660
	70 IF (NOUT.GE.0) WRITE (NOUT,99999)	COE 1670
	IFAIL = 1	COE 1680
	GO TO 90	COE 1690
	80 IF (NOUT.GE.0) WRITE (NOUT,99998)	COE 1700
	IFAIL = 2	COE 1710
	90 RETURN	COE 1720
C		COE 1730
C		COE 1740
C		COE 1750
C	***END OF SUBROUTINE COEFF**	COE 1760
C		COE 1770
	99999 FORMAT (22H MESSAGE FROM COEFF,/20H THE VALUE OF N OR,	COE 1780
	* 15H M IS TOO SMALL)	COE 1790
	99998 FORMAT (22H MESSAGE FROM COEFF,/20H THE VALUE OF I OR,	COE 1800
	* 35H J IS EITHER TOO SMALL OR TOO LARGE)	COE 1810
	END	COE 1820

```

SUBROUTINE BICUBE(C, XIM1, YJM1, XO, YO, S, SX, SY, SXY,          BIC 10
*   SXX, SY, IFAIL)                                           BIC 20
  REAL C(4,4), XIM1, YJM1, XO, YO, S, SX, SY, SXY, SXX, SY     BIC 30
C *****                                                    BIC 40
C                                                                 BIC 50
C ***SUBROUTINE BICUBE***                                       BIC 60
C                                                                 BIC 70
C                                                                 BIC 80
C   THIS SUBROUTINE EVALUATES A BICUBIC POLYNOMIAL S, AND ITS  BIC 90
C   DERIVATIVES THROUGH ORDER 2. SPECIFICALLY, GIVEN VALUES  BIC 100
C   X=XO AND Y=YO IN THE SUBRECTANGLE:                          BIC 110
C       R(I,J): X(I-1).LE.X.LT.X(I)                             BIC 120
C               Y(J-1).LE.Y.LT.Y(J)                             BIC 130
C   OF THE GRID WHICH IS DEFINED BY THE VERTICAL LINES X=X(1)  BIC 140
C   TO X=X(N), AND THE HORIZONTAL LINES Y=Y(1) TO Y=Y(M),      BIC 150
C   AND GIVEN THE 16 COEFFICIENTS OF THE BICUBIC INTERPOLATING BIC 160
C   POLYNOMIAL ASSOCIATED WITH R(I,J), THIS SUBROUTINE COMPUTES BIC 170
C   THE VALUE:                                                  BIC 180
C       S = S(XO,YO)                                           BIC 190
C   AND THE DERIVATIVES:                                        BIC 200
C       SX = DS/DX                                             BIC 210
C       SY = DS/DY                                             BIC 220
C       SXY = DS/DXDY                                         BIC 230
C       SXX = DDS/DXDX                                         BIC 240
C       SYY = DDS/DYDY                                         BIC 250
C   EVALUATED AT THE POINT XO,YO.                               BIC 260
C                                                                 BIC 270
C ***METHOD***                                                BIC 280
C                                                                 BIC 290
C   THE BICUBIC POLYNOMIAL IS EXPRESSED IN THE FORM:          BIC 300
C       S(XBAR,YBAR) = XBAR*C*YBAR                             BIC 310
C                                                                 BIC 320
C   WHERE XBAR IS A ROW VECTOR WITH COMPONENTS:                BIC 330
C                                                                 BIC 340
C       XBAR(L) = (X-X(I-1))**(L-1) L=1 TO 4                   BIC 350
C                                                                 BIC 360
C   AND YBAR IS A COLUMN VECTOR WITH COMPONENTS:               BIC 370
C                                                                 BIC 380
C       YBAR(L) = (Y-Y(J-1))**(L-1) L=1 TO 4.                 BIC 390
C                                                                 BIC 400
C   THE RESPECTIVE DERIVATIVES ARE OBTAINED BY DIFFERENTIATING BIC 410
C   THE COMPONENTS OF THE APPROPRIATE VECTORS XBAR AND YBAR.  BIC 420
C                                                                 BIC 430
C   NOTE THAT THIS ROUTINE DOES NOT EXPLICITLY REQUIRE THE    BIC 440
C   VALUE OF I AND J. IT IS ONLY NECESSARY TO SUPPLY THE      BIC 450
C   VALUE OF X(I-1) AND Y(J-1).                                BIC 460
C                                                                 BIC 470
C ***INPUT***                                                 BIC 480
C                                                                 BIC 490
C   C   IS A 4 BY 4 ARRAY WHICH CONTAINS THE COEFFICIENTS OF   BIC 500
C   THE BICUBIC POLYNOMIAL S. C IS NOT ALTERED BY THE         BIC 510
C   SUBROUTINE.                                               BIC 520
C   XIM1 IS THE VALUE OF X(I-1) WHERE I IS DEFINED ABOVE.     BIC 530
C   IF XO=X(N) THEN XIM1=X(N-1). THE VALUE OF XIM1 IS NOT    BIC 540
C   ALTERED BY THE SUBROUTINE.                                 BIC 550

```

C	YJM1	IS THE VALUE OF Y(J-1) WHERE J IS DEFINED ABOVE.	BIC	560
C		IF YO=Y(M) THEN YJM1=Y(M-1). THE VALUE OF YJM1 IS NOT	BIC	570
C		ALTERED BY THE SUBROUTINE.	BIC	580
C	XO	IS THE VALUE OF THE X-ARGUMENT AT WHICH S IS TO BE	BIC	590
C		EVALUATED. RESTRICTION: THE VALUE OF XO MUST BE	BIC	600
C		CONSISTENT WITH THE VALUE OF I AS DEFINED ABOVE.	BIC	610
C		XO IS NOT ALTERED BY THE SUBROUTINE.	BIC	620
C	YO	IS THE VALUE OF THE Y-ARGUMENT AT WHICH S IS TO BE	BIC	630
C		EVALUATED. RESTRICTION: THE VALUE OF YO MUST BE	BIC	640
C		CONSISTENT WITH THE VALUE OF J AS DEFINED ABOVE.	BIC	650
C		YO IS NOT ALTERED BY THE SUBROUTINE.	BIC	660
C	**OUTPUT**		BIC	670
C			BIC	680
C	S, SX, SY, SXY, SXX, SYY	AS DEFINED ABOVE.	BIC	690
C	IFAIL	IS AN ERROR FLAG. ON OUTPUT FROM THE	BIC	700
C		SUBROUTINE IT HAS ONE OF THE FOLLOWING VALUES:	BIC	710
C	IFAIL=0	SUCCESSFUL ENTRY	BIC	720
C	IFAIL=1	XIM1 OR YJM1 IS NOT CORRECT	BIC	730
C			BIC	740
C	**QUALITY ASSURANCE AND SOFTWARE STANDARD**		BIC	750
C			BIC	760
C	THIS SUBROUTINE	HAS BEEN WRITTEN TO CONFORM	BIC	770
C	TO THE FORTRAN IV	ANSI STANDARD 1966, AND	BIC	780
C	IT HAS BEEN	VERIFIED USING THE BELL	BIC	790
C	TELEPHONE	LABORATORIES FORTRAN VERIFIER:	BIC	800
C	PFORT.		BIC	810
C	THE SUBROUTINE	HAS RUN SUCCESSFULLY ON	BIC	820
C	A VARIETY OF	TEST PROBLEMS, AND IT HAS	BIC	830
C	BEEN ANALYSED	FOR ERRORS USING THE	BIC	840
C	DAVE SYSTEM	FROM THE UNIVERSITY OF	BIC	850
C	COLORADO.		BIC	860
C			BIC	870
C	P.W.GAFFNEY	1ST. MARCH 1979	BIC	880
C			BIC	890
C	*****		BIC	900
C			BIC	910
C			BIC	920
C	THE FOLLOWING	ARRAYS ARE USED FOR WORKSPACE	BIC	930
C	REAL	XX(4), YY(4), DX(4), DY(4), D2X(4), D2Y(4)	BIC	940
C			BIC	950
C	SET THE	OUTPUT STREAM FOR DIAGNOSTIC PRINTING	BIC	960
C	TO	SUPPRESS PRINTING SET NOUT.LT.0	BIC	970
C			BIC	980
C	DATA	NOUT /6/	BIC	990
C			BIC	1000
C	SET	SOME CONSTANTS	BIC	1010
C			BIC	1020
C	DATA	ZERO, ONE, TWO, THREE, SIX /0.,1.,2.,3.,6./	BIC	1030
C			BIC	1040
C	CHECK	THE INPUT DATA. NOTE THAT, BECAUSE THIS SUBROUTINE	BIC	1050
C	DOES	NOT COMPUTE THE COEFFICIENTS OF THE BICUBIC, IT IS	BIC	1060
C	LEFT	TO THE USER TO CHECK THAT THE ARRAY C CONTAINS THE	BIC	1070
C	COEFFICIENTS	THAT ARE ASSOCIATED WITH R(I,J). MOREOVER	BIC	1080
C	SINCE	THIS SUBROUTINE DOES NOT HAVE ACCESS TO THE	BIC	1090
C	DATA	POINTS X(I),Y(J) OR TO THE QUANTITIES N,M,I,OR J	BIC	1100

C	IT IS NOT POSSIBLE TO CHECK THAT X0,Y0,I AND J	BIC 1110
C	HAVE BEEN DEFINED CORRECTLY BY THE USER.	BIC 1120
C		BIC 1130
	IF (X0.LT.XIM1 .OR. Y0.LT.YJM1) GO TO 30	BIC 1140
	IFAIL = 0	BIC 1150
C		BIC 1160
C	SET THE COMPONENTS OF XBAR AND YBAR AND THEIR DERIVATIVES	BIC 1170
C		BIC 1180
	XBAR = X0 - XIM1	BIC 1190
	YBAR = Y0 - YJM1	BIC 1200
C		BIC 1210
	XX(1) = ONE	BIC 1220
	XX(2) = XBAR	BIC 1230
	XX(3) = XBAR*XBAR	BIC 1240
	XX(4) = XBAR**3	BIC 1250
C		BIC 1260
	YY(1) = ONE	BIC 1270
	YY(2) = YBAR	BIC 1280
	YY(3) = YBAR*YBAR	BIC 1290
	YY(4) = YBAR**3	BIC 1300
C		BIC 1310
	DX(1) = ZERO	BIC 1320
	DX(2) = ONE	BIC 1330
	DX(3) = TWO*XBAR	BIC 1340
	DX(4) = THREE*XBAR*XBAR	BIC 1350
	DY(1) = ZERO	BIC 1360
	DY(2) = ONE	BIC 1370
	DY(3) = TWO*YBAR	BIC 1380
	DY(4) = THREE*YBAR*YBAR	BIC 1390
C		BIC 1400
	D2X(1) = ZERO	BIC 1410
	D2X(2) = ZERO	BIC 1420
	D2X(3) = TWO	BIC 1430
	D2X(4) = SIX*XBAR	BIC 1440
	D2Y(1) = ZERO	BIC 1450
	D2Y(2) = ZERO	BIC 1460
	D2Y(3) = TWO	BIC 1470
	D2Y(4) = SIX*YBAR	BIC 1480
C		BIC 1490
C	COMPUTE S AND ITS DERIVATIVES	BIC 1500
C		BIC 1510
	S = ZERO	BIC 1520
	SX = ZERO	BIC 1530
	SY = ZERO	BIC 1540
	SXY = ZERO	BIC 1550
	SXX = ZERO	BIC 1560
	SYY = ZERO	BIC 1570
	DO 20 JJ=1,4	BIC 1580
	YJ1 = YY(JJ)	BIC 1590
	DO 10 II=1,4	BIC 1600
	X1I = XX(II)	BIC 1610
	CIJ = C(II,JJ)	BIC 1620
	S = S + X1I*CIJ*YJ1	BIC 1630
	SX = SX + DX(II)*CIJ*YJ1	BIC 1640
	SY = SY + X1I*CIJ*DY(JJ)	BIC 1650

	SXY = SXY + DX(II)*CIJ*DY(JJ)	BIC 1660
	SXX = SXX + D2X(II)*CIJ*YJ1	BIC 1670
	SYX = SYX + X1I*CIJ*D2Y(JJ)	BIC 1680
10	CONTINUE	BIC 1690
20	CONTINUE	BIC 1700
	GO TO 40	BIC 1710
C		BIC 1720
C	DIAGNOSTIC PRINTING	BIC 1730
C		BIC 1740
	30 IF (NOUT.GE.0) WRITE (NOUT,99999)	BIC 1750
	IFAIL = 1	BIC 1760
	40 RETURN	BIC 1770
C		BIC 1780
C		BIC 1790
C		BIC 1800
C	***END OF SUBROUTINE BICUBE**	BIC 1810
C		BIC 1820
99999	FORMAT (23H MESSAGE FROM BICUBE,/19H XIM1 OR YJM1 HAS,	BIC 1830
	* 28H NOT BEEN DEFINED CORRECTLY.)	BIC 1840
	END	BIC 1850

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.

INTERNAL DISTRIBUTION

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

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