NAG Fortran Library Routine Document

E01DAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

E01DAF computes a bicubic spline interpolating surface through a set of data values, given on a rectangular grid in the x-y plane.

2 Specification

```
      SUBROUTINE E01DAF(MX, MY, X, Y, F, PX, PY, LAMDA, MU, C, WRK, IFAIL)
      INTEGER           MX, MY, PX, PY, IFAIL
      real              X(MX), Y(MY), F(MX*MY), LAMDA(MX+4), MU(MY+4),
     1                  C(MX*MY), WRK((MX+6)*(MY+6))
```

3 Description

This routine determines a bicubic spline interpolant to the set of data points $(x_q, y_r, f_{q,r})$, for $q = 1, 2, \ldots, m_x$; $r = 1, 2, \ldots, m_y$. The spline is given in the B-spline representation

$$s(x,y) = \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} c_{ij} M_i(x) N_j(y),$$

such that

$$s(x_q, y_r) = f_{q,r},$$

where $M_i(x)$ and $N_j(y)$ denote normalised cubic B-splines, the former defined on the knots λ_i to λ_{i+4} and the latter on the knots μ_j to μ_{j+4}, and the c_{ij} are the spline coefficients. These knots, as well as the coefficients, are determined by the routine, which is derived from the routine B2IRE in Anthony *et al.* (1982). The method used is described in Section 8.2.

For further information on splines, see Hayes and Halliday (1974) for bicubic splines and de Boor (1972) for normalised B-splines.

Values of the computed spline can subsequently be obtained by calling E02DEF or E02DFF as described in Section 8.3.

4 References

Anthony G T, Cox M G and Hayes J G (1982) *DASL – Data Approximation Subroutine Library* National Physical Laboratory

Cox M G (1975a) An algorithm for spline interpolation *J. Inst. Math. Appl.* **15** 95–108

de Boor C (1972) On calculating with B-splines *J. Approx. Theory* **6** 50–62

Hayes J G and Halliday J (1974) The least-squares fitting of cubic spline surfaces to general data sets *J. Inst. Math. Appl.* **14** 89–103

5 Parameters

1: MX – INTEGER *Input*

2: MY – INTEGER *Input*

On entry: MX and MY must specify m_x and m_y respectively, the number of points along the x and y axis that define the rectangular grid.

Constraint: MX ≥ 4 and MY ≥ 4.

3: X(MX) – ***real*** array *Input*

4: Y(MY) – ***real*** array *Input*

On entry: X(q) and Y(r) must contain x_q, for $q = 1, 2, \ldots, m_x$, and y_r, for $r = 1, 2, \ldots, m_y$, respectively.

Constraints:

X$(q) <$ X$(q+1)$, for $q = 1, 2, \ldots, m_x - 1$,
Y$(r) <$ Y$(r+1)$, for $r = 1, 2, \ldots, m_y - 1$.

5: F(MX*MY) – ***real*** array *Input*

On entry: F$(m_y \times (q-1) + r)$ must contain $f_{q,r}$, for $q = 1, 2, \ldots, m_x$; $r = 1, 2, \ldots, m_y$.

6: PX – INTEGER *Output*

7: PY – INTEGER *Output*

On exit: PX and PY contain $m_x + 4$ and $m_y + 4$, the total number of knots of the computed spline with respect to the x and y variables, respectively.

8: LAMDA(MX+4) – ***real*** array *Output*

9: MU(MY+4) – ***real*** array *Output*

On exit: LAMDA contains the complete set of knots λ_i associated with the x variable, i.e., the interior knots LAMDA(5), LAMDA(6),..., LAMDA(PX − 4), as well as the additional knots

LAMDA(1) = LAMDA(2) = LAMDA(3) = LAMDA(4) = X(1)

and

LAMDA(PX − 3) = LAMDA(PX − 2) = LAMDA(PX − 1) = LAMDA(PX) = X(MX)

needed for the B-spline representation. MU contains the corresponding complete set of knots μ_i associated with the y variable.

10: C(MX*MY) – ***real*** array *Output*

On exit: the coefficients of the spline interpolant. C$(m_y \times (i-1) + j)$ contains the coefficient c_{ij} described in Section 3.

11: WRK((MX+6)*(MY+6)) – ***real*** array *Workspace*

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, MX < 4,
or MY < 4.

IFAIL $= 2$

On entry, either the values in the X array or the values in the Y array are not in increasing order if not already there.

IFAIL $= 3$

A system of linear equations defining the B-spline coefficients was singular; the problem is too ill-conditioned to permit solution.

7 Accuracy

The main sources of rounding errors are in steps 2, 3, 6 and 7 of the algorithm described in Section 8.2. It can be shown (Cox (1975a)) that the matrix A_x formed in step 2 has elements differing relatively from their true values by at most a small multiple of 3ϵ, where ϵ is the ***machine precision***. A_x is 'totally positive', and a linear system with such a coefficient matrix can be solved quite safely by elimination without pivoting. Similar comments apply to steps 6 and 7. Thus the complete process is numerically stable.

8 Further Comments

8.1 Timing

The time taken by this routine is approximately proportional to $m_x m_y$.

8.2 Outline of method used

The process of computing the spline consists of the following steps:

1. choice of the interior x-knots λ_5, $\lambda_6, \ldots, \lambda_{m_x}$ as $\lambda_i = x_{i-2}$, for $i = 5, 6, \ldots, m_x$,
2. formation of the system
$$A_x E = F,$$
where A_x is a band matrix of order m_x and bandwidth 4, containing in its qth row the values at x_q of the B-splines in x, F is the m_x by m_y rectangular matrix of values $f_{q,r}$, and E denotes an m_x by m_y rectangular matrix of intermediate coefficients,
3. use of Gaussian elimination to reduce this system to band triangular form,
4. solution of this triangular system for E,
5. choice of the interior y knots μ_5, $\mu_6, \ldots, \mu_{m_y}$ as $\mu_i = y_{i-2}$, for $i = 5, 6, \ldots, m_y$,
6. formation of the system
$$A_y C^T = E^T,$$
where A_y is the counterpart of A_x for the y variable, and C denotes the m_x by m_y rectangular matrix of values of c_{ij},
7. use of Gaussian elimination to reduce this system to band triangular form,

8. solution of this triangular system for C^T and hence C.

For computational convenience, steps 2 and 3, and likewise steps 6 and 7, are combined so that the formation of A_x and A_y and the reductions to triangular form are carried out one row at a time.

8.3 Evaluation of Computed Spline

The values of the computed spline at the points $(\mathrm{TX}(r), \mathrm{TY}(r))$, for $r = 1, 2, \ldots, \mathrm{N}$, may be obtained in the ***real*** array FF, of length at least N, by the following call:

```
IFAIL = 0
CALL E02DEF(N,PX,PY,TX,TY,LAMDA,MU,C,FF,WRK,IWRK,IFAIL)
```

where PX, PY, LAMDA, MU and C are the output parameters of E01DAF, WRK is a ***real*** workspace array of length at least PY − 4, and IWRK is an integer workspace array of length at least PY − 4.

To evaluate the computed spline on an NX by NY rectangular grid of points in the x-y plane, which is defined by the x co-ordinates stored in TX(q), for $q = 1, 2, \ldots$,NX, and the y co-ordinates stored in TY(r), for $r = 1, 2, \ldots$,NY, returning the results in the ***real*** array FG which is of length at least NX× NY, the following call may be used:

```
 IFAIL = 0
 CALL E02DFF(NX,NY,PX,PY,TX,TY,LAMDA,MU,C,FG,WRK,LWRK,
*            IWRK,LIWRK,IFAIL)
```

where PX, PY, LAMDA, MU and C are the output parameters of E01DAF, WRK is a ***real*** workspace array of length at least $\mathrm{LWRK} = \min(\mathrm{NWRK1}, \mathrm{NWRK2})$, $\mathrm{NWRK1} = \mathrm{NX} \times 4 + \mathrm{PX}$, $\mathrm{NWRK2} = \mathrm{NY} \times 4 + \mathrm{PY}$, and IWRK is an integer workspace array of length at least $\mathrm{LIWRK} = \mathrm{NY} + \mathrm{PY} - 4$ if NWRK1 $>$ NWRK2, or $\mathrm{NX} + \mathrm{PX} - 4$ otherwise. The result of the spline evaluated at grid point (q, r) is returned in element $(\mathrm{NY} \times (q - 1) + r)$ of the array FG.

9 Example

This program reads in values of m_x, x_q for $q = 1, 2, \ldots, m_x$, m_y and y_r for $r = 1, 2, \ldots, m_y$, followed by values of the ordinates $f_{q,r}$ defined at the grid points (x_q, y_r).

It then calls E01DAF to compute a bicubic spline interpolant of the data values, and prints the values of the knots and B-spline coefficients. Finally it evaluates the spline at a small sample of points on a rectangular grid.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     E01DAF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MXMAX, MYMAX
      PARAMETER        (MXMAX=20,MYMAX=MXMAX)
      INTEGER          LIWRK, LWRK
      PARAMETER        (LIWRK=MXMAX+2*(MXMAX-3)*(MYMAX-3),LWRK=(MXMAX+6)
     +                 *(MYMAX+6))
*     .. Local Scalars ..
      real             STEP, XHI, XLO, YHI, YLO
      INTEGER          I, IFAIL, J, MX, MY, NX, NY, PX, PY
*     .. Local Arrays ..
      real             C(MXMAX*MYMAX), F(MXMAX*MYMAX), FG(MXMAX*MYMAX),
     +                 LAMDA(MXMAX+4), MU(MYMAX+4), TX(MXMAX),
     +                 TY(MYMAX), WRK(LWRK), X(MXMAX), Y(MYMAX)
      INTEGER          IWRK(LIWRK)
      CHARACTER*10     CLABS(MYMAX), RLABS(MXMAX)
*     .. External Subroutines ..
      EXTERNAL         E01DAF, E02DFF, X04CBF
```

```
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01DAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*     Read the number of X points, MX, and the values of the
*     X co-ordinates.
      READ (NIN,*) MX
      READ (NIN,*) (X(I),I=1,MX)
*     Read the number of Y points, MY, and the values of the
*     Y co-ordinates.
      READ (NIN,*) MY
      READ (NIN,*) (Y(I),I=1,MY)
*     Read the function values at the grid points.
      DO 20 J = 1, MY
         READ (NIN,*) (F(MY*(I-1)+J),I=1,MX)
   20 CONTINUE
      IFAIL = 0
*
*     Generate the (X,Y,F) interpolating bicubic B-spline.
      CALL E01DAF(MX,MY,X,Y,F,PX,PY,LAMDA,MU,C,WRK,IFAIL)
*
*     Print the knot sets, LAMDA and MU.
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +  '               I   Knot LAMDA(I)      J     Knot MU(J)'
      DO 40 J = 4, MAX(PX,PY) - 3
         IF (J.LE.PX-3 .AND. J.LE.PY-3) THEN
            WRITE (NOUT,99997) J, LAMDA(J), J, MU(J)
         ELSE IF (J.LE.PX-3) THEN
            WRITE (NOUT,99997) J, LAMDA(J)
         ELSE IF (J.LE.PY-3) THEN
            WRITE (NOUT,99996) J, MU(J)
         END IF
   40 CONTINUE
*     Print the spline coefficients.
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'The B-Spline coefficients:'
      WRITE (NOUT,99999) (C(I),I=1,MX*MY)
      WRITE (NOUT,*)
*     Evaluate the spline on a regular rectangular grid at NX*NY
*     points over the domain (XLO to XHI) x (YLO to YHI).
      READ (NIN,*) NX, XLO, XHI
      READ (NIN,*) NY, YLO, YHI
      IF (NX.LE.MXMAX .AND. NY.LE.MYMAX) THEN
         STEP = (XHI-XLO)/(NX-1)
         DO 60 I = 1, NX
*           Generate NX equispaced X co-ordinates.
            TX(I) = MIN(XLO+(I-1)*STEP,XHI)
*           Generate X axis labels for printing results.
            WRITE (CLABS(I),99998) TX(I)
   60    CONTINUE
         STEP = (YHI-YLO)/(NY-1)
         DO 80 I = 1, NY
            TY(I) = MIN(YLO+(I-1)*STEP,YHI)
            WRITE (RLABS(I),99998) TY(I)
   80    CONTINUE
*
*        Evaluate the spline.
         CALL E02DFF(NX,NY,PX,PY,TX,TY,LAMDA,MU,C,FG,WRK,LWRK,IWRK,
     +               LIWRK,IFAIL)
*
*        Print the results.
         CALL X04CBF('General','X',NY,NX,FG,NY,'F8.3',
     +          'Spline evaluated on a regular mesh (X across, Y down):'
     +               ,'Character',RLABS,'Character',CLABS,80,0,IFAIL)
*
      END IF
      STOP
*
```

```
99999 FORMAT (1X,8F9.4)
99998 FORMAT (F5.2)
99997 FORMAT (1X,I16,F12.4,I11,F12.4)
99996 FORMAT (1X,I39,F12.4)
      END
```

9.2 Program Data

```
E01DAF Example Program Data
  7                                                 MX
  1.00  1.10  1.30  1.50  1.60  1.80  2.00          X(1) .. X(MX)
  6                                                 MY
  0.00  0.10  0.40  0.70  0.90  1.00                Y(1) .. Y(MY)
  1.00  1.21  1.69  2.25  2.56  3.24  4.00          (F(MY*(I-1)+J),I=1..MX),J=1..MY
  1.10  1.31  1.79  2.35  2.66  3.34  4.10
  1.40  1.61  2.09  2.65  2.96  3.64  4.40
  1.70  1.91  2.39  2.95  3.26  3.94  4.70
  1.90  2.11  2.59  3.15  3.46  4.14  4.90
  2.00  2.21  2.69  3.25  3.56  4.24  5.00
  6  1.0  2.0                                       NX  XLO  XHI
  6  0.0  1.0                                       NY  YLO  YHI
```

9.3 Program Results

```
 E01DAF Example Program Results

                I   Knot LAMDA(I)      J     Knot MU(J)
                4      1.0000          4      0.0000
                5      1.3000          5      0.4000
                6      1.5000          6      0.7000
                7      1.6000          7      1.0000
                8      2.0000

 The B-Spline coefficients:
    1.0000   1.1333   1.3667   1.7000   1.9000   2.0000   1.2000   1.3333
    1.5667   1.9000   2.1000   2.2000   1.5833   1.7167   1.9500   2.2833
    2.4833   2.5833   2.1433   2.2767   2.5100   2.8433   3.0433   3.1433
    2.8667   3.0000   3.2333   3.5667   3.7667   3.8667   3.4667   3.6000
    3.8333   4.1667   4.3667   4.4667   4.0000   4.1333   4.3667   4.7000
    4.9000   5.0000

 Spline evaluated on a regular mesh (X across, Y down):
          1.00    1.20    1.40    1.60    1.80    2.00
 0.00    1.000   1.440   1.960   2.560   3.240   4.000
 0.20    1.200   1.640   2.160   2.760   3.440   4.200
 0.40    1.400   1.840   2.360   2.960   3.640   4.400
 0.60    1.600   2.040   2.560   3.160   3.840   4.600
 0.80    1.800   2.240   2.760   3.360   4.040   4.800
 1.00    2.000   2.440   2.960   3.560   4.240   5.000
```