

E02DFF – NAG Fortran Library Routine Document

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

E02DFF calculates values of a bicubic spline from its B-spline representation. The spline is evaluated at all points on a rectangular grid.

2 Specification

```

SUBROUTINE E02DFF(MX, MY, PX, PY, X, Y, LAMDA, MU, C, FF, WRK,
1                LWRK, IWRK, LIWRK, IFAIL)
INTEGER          MX, MY, PX, PY, LWRK, IWRK(LIWRK), LIWRK, IFAIL
  real          X(MX), Y(MY), LAMDA(PX), MU(PY),
1                C((PX-4)*(PY-4)), FF(MX*MY), WRK(LWRK)

```

3 Description

This routine calculates values of the bicubic spline $s(x, y)$ on a rectangular grid of points in the x - y plane, from its augmented knot sets $\{\lambda\}$ and $\{\mu\}$ and from the coefficients c_{ij} , for $i = 1, 2, \dots, PX - 4$; $j = 1, 2, \dots, PY - 4$, in its B-spline representation

$$s(x, y) = \sum_{ij} c_{ij} M_i(x) N_j(y).$$

Here $M_i(x)$ 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} .

The points in the grid are defined by co-ordinates x_q , for $q = 1, 2, \dots, m_x$, along the x axis, and co-ordinates y_r , for $r = 1, 2, \dots, m_y$ along the y axis.

This routine may be used to calculate values of a bicubic spline given in the form produced by E01DAF, E02DAF, E02DCF and E02DDF. It is derived from the routine B2VRE in Anthony *et al.* [1].

4 References

- [1] Anthony G T, Cox M G and Hayes J G (1982) *DASL – Data Approximation Subroutine Library* National Physical Laboratory
- [2] Cox M G (1978) The numerical evaluation of a spline from its B-spline representation *J. Inst. Math. Appl.* **21** 135–143

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: $MX \geq 1$ and $MY \geq 1$.

3: PX — INTEGER *Input*
4: PY — INTEGER *Input*

On entry: PX and PY must specify the total number of knots associated with the variables x and y respectively. They are such that $PX - 8$ and $PY - 8$ are the corresponding numbers of interior knots.

Constraint: $PX \geq 8$ and $PY \geq 8$.

- 5: X(MX) — *real* array *Input*
 6: Y(MY) — *real* array *Input*

On entry: X and Y must contain x_q , for $q = 1, 2, \dots, m_x$, and y_r , for $r = 1, 2, \dots, m_y$, respectively. These are the x and y co-ordinates that define the rectangular grid of points at which values of the spline are required.

Constraint: X and Y must satisfy

$$\text{LAMDA}(4) \leq X(q) < X(q+1) \leq \text{LAMDA}(\text{PX} - 3), \text{ for } q = 1, 2, \dots, m_x - 1,$$

and

$$\text{MU}(4) \leq Y(r) < Y(r+1) \leq \text{MU}(\text{PY} - 3), \text{ for } r = 1, 2, \dots, m_y - 1.$$

The spline representation is not valid outside these intervals.

- 7: LAMDA(PX) — *real* array *Input*
 8: MU(PY) — *real* array *Input*

On entry: LAMDA and MU must contain the complete sets of knots $\{\lambda\}$ and $\{\mu\}$ associated with the x and y variables respectively.

Constraint: the knots in each set must be in non-decreasing order, with $\text{LAMDA}(\text{PX} - 3) > \text{LAMDA}(4)$ and $\text{MU}(\text{PY} - 3) > \text{MU}(4)$.

- 9: C((PX-4)*(PY-4)) — *real* array *Input*

On entry: C((PY - 4) × (i - 1) + j) must contain the coefficient c_{ij} described in Section 3, for $i = 1, 2, \dots, \text{PX} - 4$; $j = 1, 2, \dots, \text{PY} - 4$.

- 10: FF(MX*MY) — *real* array *Output*

On exit: FF(MY × (q-1) + r) contains the value of the spline at the point (x_q, y_r) , for $q = 1, 2, \dots, m_x$; $r = 1, 2, \dots, m_y$.

- 11: WRK(LWRK) — *real* array *Workspace*
 12: LWRK — INTEGER *Input*

On entry: the dimension of the array WRK as declared in the (sub)program from which E02DFF is called.

Constraint: $\text{LWRK} \geq \min(\text{NWRK1}, \text{NWRK2})$, where $\text{NWRK1} = 4 \times \text{MX} + \text{PX}$, $\text{NWRK2} = 4 \times \text{MY} + \text{PY}$.

- 13: IWRK(LIWRK) — INTEGER array *Workspace*
 14: LIWRK — INTEGER *Input*

On entry: the dimension of the array IWRK as declared in the (sub)program from which E02DFF is called.

Constraint: $\text{LIWRK} \geq \text{MY} + \text{PY} - 4$ if $\text{NWRK1} > \text{NWRK2}$, or $\text{MX} + \text{PX} - 4$ otherwise, where NWRK1 and NWRK2 are as defined in the description of argument LWRK.

- 15: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

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 detected by the routine:

`IFAIL = 1`

On entry, `MX < 1`,
 or `MY < 1`,
 or `PY < 8`,
 or `PX < 8`.

`IFAIL = 2`

On entry, `LWRK` is too small,
 or `LIWRK` is too small.

`IFAIL = 3`

On entry, the knots in array `LAMDA`, or those in array `MU`, are not in non-decreasing order, or $LAMDA(PX - 3) \leq LAMDA(4)$, or $MU(PY - 3) \leq MU(4)$.

`IFAIL = 4`

On entry, the restriction $LAMDA(4) \leq X(1) < \dots < X(MX) \leq LAMDA(PX - 3)$, or the restriction $MU(4) \leq Y(1) < \dots < Y(MY) \leq MU(PY - 3)$, is violated.

7 Accuracy

The method used to evaluate the B-splines is numerically stable, in the sense that each computed value of $s(x_r, y_r)$ can be regarded as the value that would have been obtained in exact arithmetic from slightly perturbed B-spline coefficients. See Cox [2] for details.

8 Further Comments

Computation time is approximately proportional to $m_x m_y + 4(m_x + m_y)$.

9 Example

This program reads in knot sets `LAMDA(1),...,LAMDA(PX)` and `MU(1),...,MU(PY)`, and a set of bicubic spline coefficients c_{ij} . Following these are values for m_x and the x co-ordinates x_q , for $q = 1, 2, \dots, m_x$, and values for m_y and the y co-ordinates y_r , for $r = 1, 2, \dots, m_y$, defining the grid of points on which the spline is to be evaluated.

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.

```
*      E02DFF Example Program Text
*      Mark 14 Release.  NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER       (NIN=5,NOUT=6)
      INTEGER          MXMAX, MYMAX, PXMAX, PYMAX
      PARAMETER       (MXMAX=20,MYMAX=MXMAX,PXMAX=MXMAX,PYMAX=PXMAX)
      INTEGER          LIWRK, LWRK
```

```

PARAMETER      (LIWRK=MXMAX+PXMAX-4,LWRK=4*MXMAX+PXMAX+8)
*
* .. Local Scalars ..
INTEGER        I, IFAIL, MX, MY, PX, PY
*
* .. Local Arrays ..
real          C((PXMAX-4)*(PYMAX-4)), FF(MXMAX*MYMAX),
+            LAMDA(PXMAX), MU(PYMAX), WRK(LWRK), X(MXMAX),
+            Y(MYMAX)
INTEGER        IWRK(LIWRK)
CHARACTER*10   CLABS(MYMAX), RLABS(MXMAX)
*
* .. External Subroutines ..
EXTERNAL       E02DFF, X04CBF
*
* .. Executable Statements ..
WRITE (NOUT,*) 'E02DFF Example Program Results'
*
* Skip heading in data file
READ (NIN,*)
WRITE (NOUT,*)
*
* Read PX and PY, the number of knots in the X and Y directions.
READ (NIN,*) PX, PY
IF (PX.LE.PXMAX .AND. PY.LE.PYMAX) THEN
*
*   Read the knots LAMDA(1) .. LAMDA(PX) and MU(1) .. MU(PY).
READ (NIN,*) (LAMDA(I),I=1,PX)
READ (NIN,*) (MU(I),I=1,PY)
*
*   Read C, the bicubic spline coefficients.
READ (NIN,*) (C(I),I=1,(PX-4)*(PY-4))
*
*   Read MX and MY, the number of grid points in the X and Y
*   directions respectively.
READ (NIN,*) MX, MY
IF (MX.LE.MXMAX .AND. MY.LE.MYMAX) THEN
*
*   Read the X and Y co-ordinates defining the evaluation grid.
READ (NIN,*) (X(I),I=1,MX)
READ (NIN,*) (Y(I),I=1,MY)
IFAIL = 0
*
*
*   Evaluate the spline at the MX by MY points.
CALL E02DFF(MX,MY,PX,PY,X,Y,LAMDA,MU,C,FF,WRK,LWRK,IWRK,
+           LIWRK,IFAIL)
*
*   Generate column and row labels to print the results with.
DO 20 I = 1, MX
    WRITE (CLABS(I),99999) X(I)
20  CONTINUE
DO 40 I = 1, MY
    WRITE (RLABS(I),99999) Y(I)
40  CONTINUE
*
*   Print the result array.
CALL X04CBF('G','X',MY,MX,FF,MY,'F8.3',
+           'Spline evaluated on X-Y grid (X across, Y down):'
+           ',Character',RLABS,'Character',CLABS,80,0,IFAIL)
*
    END IF
END IF
STOP
*
99999 FORMAT (F5.1)
END

```

9.2 Program Data

```

E02DFF Example Program Data
11 10
1.0 1.0 1.0 1.0 1.3 1.5 1.6 2.0 2.0 2.0 2.0
0.0 0.0 0.0 0.0 0.4 0.7 1.0 1.0 1.0 1.0
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
7 6
1.0 1.1 1.3 1.4 1.5 1.7 2.0
0.0 0.2 0.4 0.6 0.8 1.0
PX PY
LAMDA(1) .. LAMDA(PX)
MU(1) .. MU(PY)

Spline coefficients, C
MX MY
X(1) .. X(MX)
Y(1) .. Y(MY)

```

9.3 Program Results

E02DFF Example Program Results

Spline evaluated on X-Y grid (X across, Y down):

	1.0	1.1	1.3	1.4	1.5	1.7	2.0
0.0	1.000	1.210	1.690	1.960	2.250	2.890	4.000
0.2	1.200	1.410	1.890	2.160	2.450	3.090	4.200
0.4	1.400	1.610	2.090	2.360	2.650	3.290	4.400
0.6	1.600	1.810	2.290	2.560	2.850	3.490	4.600
0.8	1.800	2.010	2.490	2.760	3.050	3.690	4.800
1.0	2.000	2.210	2.690	2.960	3.250	3.890	5.000
