NAG Library Routine Document

E02JFF

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

E02JFF calculates a mesh of values of a spline computed by E02JDF.

2 Specification

```
SUBROUTINE E02JFF (NXEVAL, NYEVAL, XEVALM, YEVALM, COEFS, FEVALM, IOPTS,    &
                   OPTS, IFAIL)
INTEGER            NXEVAL, NYEVAL, IOPTS(*), IFAIL
REAL (KIND=nag_wp) XEVALM(NXEVAL), YEVALM(NYEVAL), COEFS(*),                &
                   FEVALM(NXEVAL,NYEVAL), OPTS(*)
```

3 Description

E02JFF calculates values on a rectangular mesh of a bivariate spline computed by E02JDF. The points in the mesh are defined by x coordinates (x_i), for $i = 1, 2, \ldots, n_x$, and y coordinates (y_j), for $j = 1, 2, \ldots, n_y$. This routine is derived from the TSFIT package of O. Davydov and F. Zeilfelder.

4 References

Davydov O, Morandi R and Sestini A (2006) Local hybrid approximation for scattered data fitting with bivariate splines *Comput. Aided Geom. Design* **23** 703–721

Davydov O, Sestini A and Morandi R (2005) Local RBF approximation for scattered data fitting with bivariate splines *Trends and Applications in Constructive Approximation* M. G. de Bruin, D. H. Mache, and J. Szabados, Eds **ISNM Vol. 151** Birkhauser 91–102

Davydov O and Zeilfelder F (2004) Scattered data fitting by direct extension of local polynomials to bivariate splines *Advances in Comp. Math.* **21** 223–271

Farin G and Hansford D (2000) *The Essentials of CAGD* Natic, MA: A K Peters, Ltd.

5 Arguments

1: NXEVAL – INTEGER *Input*

On entry: n_x, the number of values in the x direction forming the mesh on which the spline is to be evaluated.

Constraint: NXEVAL ≥ 1.

2: NYEVAL – INTEGER *Input*

On entry: n_y, the number of values in the y direction forming the mesh on which the spline is to be evaluated.

Constraint: NYEVAL ≥ 1.

3: XEVALM(NXEVAL) – REAL (KIND=nag_wp) array *Input*

On entry: the (x_i) values forming the mesh on which the spline is to be evaluated.

Constraint: for all i, XEVALM(i) must lie inside, or on the boundary of, the spline's bounding box as determined by E02JDF.

4: YEVALM(NYEVAL) – REAL (KIND=nag_wp) array *Input*

On entry: the (y_j) values forming the mesh on which the spline is to be evaluated.

Constraint: for all j, YEVALM(j) must lie inside, or on the boundary of, the spline's bounding box as determined by E02JDF.

5: COEFS$(*)$ – REAL (KIND=nag_wp) array *Communication Array*

Note: the dimension of this array is dictated by the requirements of associated functions that must have been previously called. This array **must** be the same array passed as argument COEFS in the previous call to E02JDF.

On entry: the computed spline coefficients as output from E02JDF.

6: FEVALM(NXEVAL, NYEVAL) – REAL (KIND=nag_wp) array *Output*

On exit: if IFAIL $= 0$ on exit FEVALM(i,j) contains the computed spline value at (x_i, y_j).

7: IOPTS$(*)$ – INTEGER array *Communication Array*

Note: the dimension of this array is dictated by the requirements of associated functions that must have been previously called. This array **must** be the same array passed as argument IOPTS in the previous call to E02ZKF.

On entry: the contents of the array **must not** have been modified either directly or indirectly, by a call to E02ZKF, between calls to E02JDF and E02JFF.

8: OPTS$(*)$ – REAL (KIND=nag_wp) array *Communication Array*

Note: the dimension of this array is dictated by the requirements of associated functions that must have been previously called. This array **must** be the same array passed as argument OPTS in the previous call to E02ZKF.

On entry: the contents of the array **must not** have been modified either directly or indirectly, by a call to E02ZKF, between calls to E02JDF and E02JFF.

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (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 or warnings detected by the routine:

IFAIL $= 6$

On entry, NXEVAL $= \langle value \rangle$.
Constraint: NXEVAL ≥ 1.

IFAIL $= 7$

On entry, NYEVAL $= \langle value \rangle$.
Constraint: NYEVAL ≥ 1.

IFAIL $= 9$

Option arrays are not initialized or are corrupted.

IFAIL $= 10$

The fitting routine has not been called, or the array of coefficients has been corrupted.

IFAIL $= 13$

On entry, XEVALM$(\langle value \rangle) = \langle value \rangle$ was outside the bounding box.
Constraint: $\langle value \rangle \leq$ XEVALM$(i) \leq \langle value \rangle$ for all i.

IFAIL $= 14$

On entry, YEVALM$(\langle value \rangle) = \langle value \rangle$ was outside the bounding box.
Constraint: $\langle value \rangle \leq$ YEVALM$(j) \leq \langle value \rangle$ for all j.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

E02JFF uses the de Casteljau algorithm and thus is numerically stable. See Farin and Hansford (2000) for details.

8 Parallelism and Performance

E02JFF is not threaded in any implementation.

9 Further Comments

To evaluate a C^1 approximation (i.e., when **Global Smoothing Level** $= 1$), a real array of length $O(1)$ is dynamically allocated by each invocation of E02JFF. No memory is allocated internally when evaluating a C^2 approximation.

10 Example

See Section 10 in E02JDF.