# NAG Library Routine Document

# E02JEF

**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

E02JEF calculates a vector of values of a spline computed by E02JDF.

## 2 Specification

```
SUBROUTINE E02JEF (NEVALV, XEVALV, YEVALV, COEFS, FEVALV, IOPTS, OPTS,          &
                   IFAIL)
INTEGER            NEVALV, IOPTS(*), IFAIL
REAL (KIND=nag_wp) XEVALV(NEVALV), YEVALV(NEVALV), COEFS(*),                     &
                   FEVALV(NEVALV), OPTS(*)
```

## 3 Description

E02JEF calculates values at prescribed points $(x_i,y_i)$, for $i = 1,2,\ldots,n$, of a bivariate spline computed by E02JDF. It is derived from the TSFIT package of O. Davydov and F. Zeilfelder.

## 4 References

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 Parameters

1: NEVALV – INTEGER *Input*

*On entry*: $n$, the number of values at which the spline is to be evaluated.

*Constraint*: NEVALV $\geq 1$.

2: XEVALV(NEVALV) – REAL (KIND=nag_wp) array *Input*

*On entry*: the $(x_i)$ values at which the spline is to be evaluated.

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

3: YEVALV(NEVALV) – REAL (KIND=nag_wp) array *Input*

*On entry*: the $(y_i)$ values at which the spline is to be evaluated.

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

4: COEFS$(*)$ – REAL (KIND=nag_wp) array *Input*

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

5: FEVALV(NEVALV) – REAL (KIND=nag_wp) array *Output*

*On exit*: if IFAIL $= 0$ on exit FEVALV$(i)$ contains the computed spline value at $(x_i, y_i)$.

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

*On entry*: must be the same array IOPTS supplied in a previous call to E02JDF. The contents of the array **must not** have been modified either directly or indirectly, by a call to E02ZKF, between calls to E02JDF and E02JEF.

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

*On entry*: must be the same array OPTS supplied in a previous call to E02JDF. The contents of the array **must not** have been modified either directly or indirectly, by a call to E02ZKF, between calls to E02JDF and E02JEF.

8: IFAIL – INTEGER *Input/Output*

*On entry*: IFAIL must be set to 0, $-1$ or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 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.**

*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 $= 2$

On entry, NEVALV $= \langle$*value*$\rangle$.
Constraint: NEVALV $\geq 1$.

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

IFAIL $= 14$

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

IFAIL $= -999$

Dynamic memory allocation failed.

## 7 Accuracy

E02JEF uses the de Casteljau algorithm and thus is numerically stable. See Farin and Hansford (2000) for details.

## 8 Further Comments

A real array of length $O(1)$ is dynamically allocated by each invocation of E02JEF.

## 9 Example

See Section 9 in E02JDF.