

## NAG Library Function Document

### nag\_2d\_spline\_ts\_eval\_rect (e02jfc)

#### 1 Purpose

nag\_2d\_spline\_ts\_eval\_rect (e02jfc) calculates a mesh of values of a spline computed by nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).

#### 2 Specification

```
#include <nag.h>
#include <nage02.h>

void nag_2d_spline_ts_eval_rect (Integer nxeval, Integer nyeval,
    const double xevalm[], const double yevalm[], const double coefs[],
    double fevalm[], const Integer iopts[], const double opts[],
    NagError *fail)
```

#### 3 Description

nag\_2d\_spline\_ts\_eval\_rect (e02jfc) calculates values on a rectangular mesh of a bivariate spline computed by nag\_2d\_spline\_fit\_ts\_scatter (e02jdc). The points in the mesh are defined by  $x$  coordinates ( $x_i$ ), for  $i = 1, 2, \dots, n_x$ , and  $y$  coordinates ( $y_j$ ), for  $j = 1, 2, \dots, n_y$ . This function 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**  $\geq 1$ .

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**  $\geq 1$ .

- 3: **xevalm**[**nxeval**] – const double *Input*  
*On entry:* the  $(x_i)$  values forming the mesh on which the spline is to be evaluated.  
*Constraint:* for all  $i$ , **xevalm**[ $i - 1$ ] must lie inside, or on the boundary of, the spline's bounding box as determined by nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).
- 4: **yevalm**[**nyeval**] – const double *Input*  
*On entry:* the  $(y_j)$  values forming the mesh on which the spline is to be evaluated.  
*Constraint:* for all  $j$ , **yevalm**[ $j - 1$ ] must lie inside, or on the boundary of, the spline's bounding box as determined by nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).
- 5: **coefs**[ $dim$ ] – const double *Communication Array*  
**Note:** the dimension,  $dim$ , 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 nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).  
*On entry:* the computed spline coefficients as output from nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).
- 6: **fevalm**[**nxeval** × **nyeval**] – double *Output*  
**Note:** the  $(i, j)$ th element of the matrix is stored in **fevalm**[ $(j - 1) \times \mathbf{nxeval} + i - 1$ ].  
*On exit:* if **fail.code** = NE\_NOERROR on exit **fevalm**[ $(j - 1) \times \mathbf{nxeval} + i - 1$ ] contains the computed spline value at  $(x_i, y_j)$ .
- 7: **iopts**[ $dim$ ] – const Integer *Communication Array*  
**Note:** the dimension,  $dim$ , 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 nag\_fit\_opt\_set (e02zkc).  
*On entry:* the contents of the array MUST NOT have been modified either directly or indirectly, by a call to nag\_fit\_opt\_set (e02zkc), between calls to nag\_2d\_spline\_fit\_ts\_scatter (e02jdc) and nag\_2d\_spline\_ts\_eval\_rect (e02jfc).
- 8: **opts**[ $dim$ ] – const double *Communication Array*  
**Note:** the dimension,  $dim$ , 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 nag\_fit\_opt\_set (e02zkc).  
*On entry:* the contents of the array MUST NOT have been modified either directly or indirectly, by a call to nag\_fit\_opt\_set (e02zkc), between calls to nag\_2d\_spline\_fit\_ts\_scatter (e02jdc) and nag\_2d\_spline\_ts\_eval\_rect (e02jfc).
- 9: **fail** – NagError \* *Input/Output*  
The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

### NE\_BAD\_PARAM

On entry, argument  $\langle value \rangle$  had an illegal value.

**NE\_INITIALIZATION**

Option arrays are not initialized or are corrupted.

**NE\_INT**

On entry, **nxeval** =  $\langle value \rangle$ .

Constraint: **nxeval**  $\geq 1$ .

On entry, **nyeval** =  $\langle value \rangle$ .

Constraint: **nyeval**  $\geq 1$ .

**NE\_INTERNAL\_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.  
See Section 3.6.6 in the Essential Introduction for further information.

**NE\_INVALID\_SPLINE**

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

**NE\_NO\_LICENCE**

Your licence key may have expired or may not have been installed correctly.  
See Section 3.6.5 in the Essential Introduction for further information.

**NE\_POINT\_OUTSIDE\_RECT**

On entry, **xevalm**[ $\langle value \rangle$ ] =  $\langle value \rangle$  was outside the bounding box.

Constraint:  $\langle value \rangle \leq \mathbf{xevalm}[i - 1] \leq \langle value \rangle$  for all  $i$ .

On entry, **yevalm**[ $\langle value \rangle$ ] =  $\langle value \rangle$  was outside the bounding box.

Constraint:  $\langle value \rangle \leq \mathbf{yevalm}[j - 1] \leq \langle value \rangle$  for all  $j$ .

**7 Accuracy**

nag\_2d\_spline\_ts\_eval\_rect (e02jfc) uses the de Casteljau algorithm and thus is numerically stable. See Farin and Hansford (2000) for details.

**8 Parallelism and Performance**

Not applicable.

**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 nag\_2d\_spline\_ts\_eval\_rect (e02jfc). No memory is allocated internally when evaluating a  $C^2$  approximation.

**10 Example**

See Section 10 in nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).

---