**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_scat (e02jdc)*.

2 Specification

```c
#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_scat (e02jdc)*. 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 function is derived from the TSFIT package of O. Davydov and F. Zeilfelder.

4 References


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: \(\text{nxeval} \geq 1\).

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: \(\text{nyeval} \geq 1\).
3: \texttt{xeval[m][nxeval]} – const double \hspace{1cm} \textit{Input}

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

\textit{Constraint:} for all \(i\), \texttt{xeval}[i - 1] must lie inside, or on the boundary of, the spline’s bounding box as determined by \texttt{nag_2d_spline_fit_ts_scat (e02jdc)}.

4: \texttt{yeval[m][nyeval]} – const double \hspace{1cm} \textit{Input}

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

\textit{Constraint:} for all \(j\), \texttt{yeval}[j - 1] must lie inside, or on the boundary of, the spline’s bounding box as determined by \texttt{nag_2d_spline_fit_ts_scat (e02jdc)}.

5: \texttt{coefs[dim]} – const double \hspace{1cm} \textit{Communication Array}

\textit{Note:} the dimension, \texttt{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 \texttt{coefs} in the previous call to \texttt{nag_2d_spline_fit_ts_scat (e02jdc)}.

\textit{On entry:} the computed spline coefficients as output from \texttt{nag_2d_spline_fit_ts_scat (e02jdc)}.

6: \texttt{feval[m][nxeval][nyeval]} – double \hspace{1cm} \textit{Output}

\textit{Note:} the \((i,j)\)th element of the matrix is stored in \texttt{feval}[(\(j - 1\)) \times \texttt{nxeval} + \(i - 1\)]

\textit{On exit:} if \texttt{fail.code} = \texttt{NE_NOERROR} on exit \texttt{feval}[(\(j - 1\)) \times \texttt{nxeval} + \(i - 1\)] contains the computed spline value at \((x_i, y_j)\).

7: \texttt{iopts[dim]} – const Integer \hspace{1cm} \textit{Communication Array}

\textit{Note:} the dimension, \texttt{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 \texttt{iopts} in the previous call to \texttt{nag_fit_opt_set (e02zkc)}.

\textit{On entry:} the contents of the array MUST NOT have been modified either directly or indirectly, by a call to \texttt{nag_fit_opt_set (e02zkc)}, between calls to \texttt{nag_2d_spline_fit_ts_scat (e02jdc)} and \texttt{nag_2d_spline_ts_eval_rect (e02jfc)}.

8: \texttt{opts[dim]} – const double \hspace{1cm} \textit{Communication Array}

\textit{Note:} the dimension, \texttt{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 \texttt{opts} in the previous call to \texttt{nag_fit_opt_set (e02zkc)}.

\textit{On entry:} the contents of the array MUST NOT have been modified either directly or indirectly, by a call to \texttt{nag_fit_opt_set (e02zkc)}, between calls to \texttt{nag_2d_spline_fit_ts_scat (e02jdc)} and \texttt{nag_2d_spline_ts_eval_rect (e02jfc)}.

9: \texttt{fail} – \texttt{NagError *} \hspace{1cm} \textit{Input/Output}

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

\textbf{NE_ALLOC_FAIL}

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

\textbf{NE_BAD_PARAM}

On entry, argument \texttt{\langle value\rangle} had an illegal value.
NE_INITIALIZATION

Option arrays are not initialized or are corrupted.

NE_INT

On entry, \(\text{nxeval} = \langle\text{value}\rangle\).
Constraint: \(\text{nxeval} \geq 1\).

On entry, \(\text{nyeval} = \langle\text{value}\rangle\).
Constraint: \(\text{nyeval} \geq 1\).

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, \(\text{xevalm}[i] = \langle\text{value}\rangle\) was outside the bounding box.
Constraint: \(\langle\text{value}\rangle \leq \text{xevalm}[i - 1] \leq \langle\text{value}\rangle\) for all \(i\).

On entry, \(\text{yevalm}[] = \langle\text{value}\rangle\) was outside the bounding box.
Constraint: \(\langle\text{value}\rangle \leq \text{yevalm}[j - 1] \leq \langle\text{value}\rangle\) for all \(j\).

7 Accuracy

\text{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 \textbf{Global Smoothing Level} = 1), a real array of length \(O(1)\) is dynamically allocated by each invocation of \text{nag\_2d\_spline\_ts\_eval\_rect (e02jfc)}. No memory is allocated internally when evaluating a \(C^2\) approximation.

10 Example

See Section 10 in \text{nag\_2d\_spline\_fit\_ts\_scat (e02jdc)}. 

Mark 25