Chapter e01 – Interpolation

1. Scope of the Chapter

This chapter provides routines for interpolating data of one or two independent variables.

2. Background

Assume that we are given a set of data points x_i and a corresponding set of values y_i, $i = 1, 2, \ldots, n$. Depending on the context, the values y_i may be the values of some 'function underlying the data', or they may simply represent experimental observations — for example, soundings of the sea-floor, or stock-market prices.

The term *interpolation* can have one of two meanings:

(i) the determination of a function of x which takes the prescribed values y_i at the points x_i, for $i = 1, 2, \ldots, n$, in which case the function is often referred to as the *interpolation function* or simply the *interpolant*.
(ii) the determination of a value (the *interpolated value* or *interpolate*) of an interpolating function for any given value x within the range of the x_i, so as to estimate the value of the function underlying the data.

Before using routines from Chapter e01, the user should seriously consider whether to interpolate the data points. If the data are not known to be completely accurate (for example if they are subject to experimental error, or have been rounded), it may not make much sense to force an approximating function to pass through all the data points precisely. Indeed, such an approach may be counter-productive, since it may lead to a function which fluctuates wildly in order to track the data, rather than to a 'smooth' function. Routines in Chapter e02 can approximate the data by a spline function containing significantly fewer coefficients than the corresponding interpolating function. Evaluation of such a function will therefore be quicker, and is less likely to produce unwanted fluctuations. Some of the routines in Chapter e02 may also be used for interpolation, by suitable choice of arguments.

2.1. One independent variable

The interpolant computed by nag_1d_spline_interpolant (e01bac) is a *cubic spline*, which uses most of the data points x_i as knots. This simple choice of knots is generally good enough to avoid unwanted fluctuations in the interpolant.

The interpolant computed by nag_monotonic_interpolant is a *piecewise cubic polynomial*, with continuous first derivative. It has the special property of *preserving monotonicity* in the data. That is, if $y_i < y_{i+1}$, then the interpolant increases monotonically over the interval $[y_i, y_{i+1}]$. This is especially useful if the data values increase monotonically over the whole range.

2.2. Two independent variables

nag_2d_spline_interpolant (e01dac) determines a bicubic spline surface interpolating a set of data values given on a rectangular grid. The two required sets of knots are chosen according to the same rule as those for nag_1d_spline_interpolant (e01bac).

For scattered data values nag_2d_scat_interpolant (e01sac) determines an interpolating surface using either the method of Renka and Cline or a modified Shepard scheme. In either case the constructed surface is continuous, with continuous first derivatives. The relative merits of the two schemes vary with the data, and it is not possible to predict which will be better in any particular case.

2.3. Evaluation of interpolants

Functions to evaluate an interpolant and to compute derivatives and integrals have been provided in this chapter as well as in Chapter e02. For a particular interpolation function, Section 6 of the function document indicates how these evaluation functions can be used.

3. Available Functions

Determine a cubic-spline interpolant to a set of data points	e01bac
Determine a monotonicity-preserving interpolant to a set of data points	e01bec
Evaluate the interpolant produced by nag_monotonic_interpolant (e01bec)	e01bfc
Evaluate the interpolant produced by nag_monotonic_interpolant (e01bec) and its first derivative	e01bgc
Evaluate the definite integral of the interpolant produced by nag_monotonic_interpolant (e01bec)	e01bhc
Determine a bicubic-spline interpolant to a set of data points on a grid	e01dac
Determine a two-dimensional surface interpolant to a set of scattered data points	e01sac
Evaluate a two-dimensional surface interpolant	e01sbc
Free the memory allocated to members of the structure used by nag_2d_scat_interpolant (e01sac) and nag_2d_scat_eval (e01sbc)	e01szc