Chapter 8

Curve and Surface Fitting

1 Scope of the Chapter

This chapter is concerned with the interpolation and approximation of data sets in one, two or three dimensions. The approximating functions available are piecewise cubic Hermite polynomials, cubic splines, bicubic splines, and interpolants produced by a modification of Shepard's method (Renka [2]).

2 Available Modules

Module 8.1: `nag_pch_interp` — **Piecewise cubic Hermite interpolation**

Provides procedures for computing and evaluating piecewise cubic Hermite interpolants to arbitrary data sets in one dimension. In particular, the module contains procedures for:

- generating a monotonicity-preserving interpolant;
- evaluating a piecewise cubic Hermite polynomial;
- integrating a piecewise cubic Hermite polynomial.

Module 8.2: `nag_spline_1d` — **One-dimensional spline fitting**

Provides procedures for computing and evaluating spline approximations to arbitrary data sets in one dimension. Procedures are available for:

- generating a cubic spline interpolant;
- generating a least-squares cubic spline fit with given interior knots;
- generating a cubic spline approximation with automatic knot placement;
- computing values of a cubic spline;
- computing the definite integral of a cubic spline.

Module 8.3: `nag_spline_2d` — **Two-dimensional spline fitting**

Provides procedures for computing and evaluating spline surface approximations to arbitrary data sets in two dimensions. Procedures are available for:

- generating a bicubic spline interpolant;
- generating a least-squares bicubic spline fit with given interior knots;
- generating a bicubic spline approximation with automatic knot placement;
- computing values of a bicubic spline;
- computing the definite integral of a bicubic spline.

Module 8.4: `nag_scat_interp` — **Interpolation of scattered data**

Provides procedures for computing and evaluating interpolants to scattered data sets. Procedures are available for:

- generating a 2-d or 3-d modified Shepard interpolant;
- computing values of a 2-d or 3-d modified Shepard interpolant, and optionally its first partial derivatives.

3 Background

3.1 Terminology and Notation

The main aim of this chapter is to provide facilities for fitting a function of one or two variables to a given set of data points. This process will be referred to as *curve fitting* in the case of a single independent variable x, and *surface fitting* if there are two independent variables x and y.

In the curve-fitting problems considered in this chapter we have a dependent variable f and an independent variable x, and are given a set of data points (x_r, f_r), for $r = 1, 2, \ldots, m$. The aim is to construct a curve $\phi(x)$ which interpolates or approximates these points.

For surface fitting there is also a single dependent variable f, but there are now two independent variables x and y. The data points are denoted (x_r, y_r, f_r), for $r = 1, 2, \ldots, m$. In the special case where these data points lie on a rectangular mesh in the (x, y) plane the alternative notation (x_q, y_r, f_{qr}), for $q = 1, 2, \ldots, m_x$, $r = 1, 2, \ldots, m_y$ may be adopted. The aim is to construct a surface $\phi(x, y)$ which interpolates or approximates the given data set.

The preliminary matters to be considered here will, for simplicity, be discussed in the context of curve-fitting problems. In fact, however, these considerations apply equally well to surface and higher-dimensional problems. Indeed, the discussion presented carries over essentially as it stands if, for these cases, we interpret x as a vector of several independent variables and correspondingly each x_r as a vector containing the rth data value of each independent variable.

3.2 Interpolation

The process of one-dimensional interpolation may be defined as:

the determination of a curve $\phi(x)$ which takes the value f_r at $x = x_r$, for $r = 1, 2, \ldots, m$.

Interpolation in higher dimensions is similarly defined.

A danger with interpolation is that the fitting function may tend to exhibit unwanted fluctuations, essentially following random errors in the data and oscillating between the data points. This problem is particularly common if the fitting function is a polynomial defined over the entire region of interest, and becomes more severe as the number of data points increases. For this reason the interpolating functions chosen in this chapter are *piecewise* polynomials, such as cubic splines. A cubic spline can often be used satisfactorily to interpolate a large number of data points over the whole of the data range. Unwanted fluctuations can arise, but much less frequently and much less severely than with global polynomials. An interpolating spline is not uniquely defined for a given data set, but depends on the choice made for the knots.

Unwanted fluctuations may be avoided altogether by a method using piecewise cubic polynomials having only first derivative continuity. It is designed especially for monotonic data, but for other data still provides an interpolant which increases, or decreases, over the same intervals as the data.

For two-dimensional interpolation the choice of procedure generally depends on the location of the data points. If these lie at the intersections of a rectangular mesh in the (x, y) plane then a bicubic spline interpolant such as `nag_spline_2d_interp` in the module `nag_spline_2d` is ideal. If however the data points are arbitrarily scattered in the (x, y) plane then `nag_scat_2d_interp` from the module `nag_scat_interp` should be used instead.

For three-dimensional interpolation the procedure `nag_scat_3d_interp` is provided in the module `nag_scat_interp`. This interpolates a 3-d scattered data set using a localized Shepard method due to Renka [2].

Interpolation or approximation

Before undertaking interpolation, in other than the simplest cases, it is advisable to consider the alternative of fitting the data by an approximant, which does not pass *exactly* through the data points, but involves significantly fewer coefficients than the corresponding interpolant. This approach is much less liable to produce unwanted fluctuations and so can often provide a better approximation to the function underlying the data.

3.3 Approximation

The aim of approximation is to fit the set of data points as closely as possible with a specified function, $\phi(x)$ say, which is as smooth as possible. The requirements of smoothness and closeness conflict, however, and a balance has to be struck between them. Most often, the smoothness requirement is met simply by limiting the number of coefficients allowed in the approximating function. Given a particular number of coefficients in the function in question, the approximation procedures of this chapter generally determine the values of the coefficients such that the 'distance' of the function from the data points is as small as possible. The necessary balance should be struck by comparing a selection of such approximants having different numbers of coefficients. If the number of coefficients is too low, the approximation to the data will be poor. If the number is too high, the fit will be too close to the data, essentially following any random errors and tending to have unwanted fluctuations between the data points, as for interpolation. Between these extremes, there is often a group of fits all similarly close to the data points and then the choice is clear: it is the approximant from this group having the smallest number of coefficients.

The above process can be seen as the minimization of the smoothness measure (i.e., the number of coefficients) subject to the distance from the data points being acceptably small. Some of the procedures, however, do this task themselves. They use a different measure of smoothness (in each case one that is continuous) and minimize it subject to the distance being less than a given threshold. This is a much more automatic process, requiring only some experimentation with the threshold.

Fitting criteria: norms

A measure of the above 'distance' between the set of data points and the function $\phi(x)$ is needed. The distance from a single data point (x_r, f_r) to the function can simply be taken as

$$\epsilon_r = f_r - \phi(x_r), \qquad (1)$$

and is called the *residual* of the point. However, we need a measure of distance for the set of data points as a whole. With ϵ_r defined in (1), a suitable measure, or *norm*, is

$$\sqrt{\sum_{r=1}^{m} \epsilon_r^2}, \qquad (2)$$

which is known as the l_2 norm.

Minimization of this norm usually provides the fitting criterion, the minimization being carried out with respect to the coefficients in the mathematical form used for $\phi(x)$. The approximant which results from minimizing (2) is the l_2 fit, the well known least-squares approximation. Note that minimizing (2) is equivalent to minimizing the square of (2), i.e., the sum of squares of residuals. It is the latter which is used in practice.

Strictly speaking, implicit in the use of the above norm is the statistical assumption that the random errors in the f_r are independent of one another and that any errors in the x_r are negligible by comparison. From this point of view, the use of the l_2 norm is appropriate when the random errors in the f_r have a Normal distribution.

Some of the procedures in this chapter do not minimize the l_2 norm itself, but instead minimize some (intuitively acceptable) measure of smoothness subject to the norm being less than some given threshold. These procedures fit with cubic or bicubic splines, and the smoothing measures relate to the size of the discontinuities in their third derivatives. A much more automatic approximation algorithm follows from this approach.

Weighting of data points

The use of the above norm also assumes that the data values f_r are of equal (absolute) accuracy. Some of the procedures enable an allowance to be made to take account of differing accuracies. The allowance takes the form of 'weights' applied to the f-values so that those values known to be more accurate have a greater influence on the fit than others. These weights should be calculated from estimates of the absolute accuracies of the f-values, these estimates being expressed as standard deviations, probable errors or some other measure which has the same dimensions as f. Specifically, for each f_r the corresponding weight w_r should be inversely proportional to the accuracy estimate of f_r. For example, if the percentage accuracy is the same for all f_r, then the absolute accuracy of f_r is proportional to f_r (assuming f_r to be positive, as it usually is in such cases) and so $w_r = K/f_r$, for $r = 1, 2, \ldots, m$, for an arbitrary positive constant K. (This definition of weight is stressed because often weight is defined as the square of that used here.) The norm (2) above is then replaced by

$$\sqrt{\sum_{r=1}^{m} w_r^2 \epsilon_r^2}. \tag{3}$$

Again it is the square of (3) which is used in practice rather than (3) itself.

3.4 Data Considerations

A satisfactory fit cannot be expected by any means if the number and arrangement of the data points do not adequately represent the character of the underlying relationship: sharp changes in behaviour, in particular, such as sharp peaks, should be well covered. Data points should extend over the whole range of interest of the independent variable(s): extrapolation outside the data ranges is most unwise. All fits should be tested graphically before accepting them as satisfactory.

For this purpose it should be noted that it is not sufficient to plot the values of the fitted function only at the data values of the independent variable(s); at the least, its values at a similar number of intermediate points should also be plotted, as unwanted fluctuations may otherwise go undetected. Such fluctuations are the less likely to occur the lower the number of coefficients chosen in the fitting function. No firm guide can be given, but as a rough rule, at least initially, the number of coefficients defining an approximant should not exceed half the number of data points (points with equal or nearly equal values of the independent variable, or both independent variables in surface fitting, counting as a single point for this purpose). However, the situation may be such, particularly with a small number of data points, that a satisfactorily close fit to the data cannot be achieved without unwanted fluctuations occurring. In such cases, it is often possible to improve the situation by a transformation of one or more of the variables, as discussed in the next paragraph; otherwise it will be necessary to provide extra data points. Further advice on curve fitting is given in Cox and Hayes [1]. Much of the advice applies also to surface fitting; see also the module documents.

3.5 Transformation of Variables

Before starting the fitting, consideration should be given to the choice of a good form in which to deal with each of the variables: often it will be satisfactory to use the variables as they stand, but sometimes the use of the logarithm, square root, or some other function of a variable will lead to a better behaved relationship. This question is customarily taken into account in preparing graphs and tables of a relationship and the same considerations apply when curve or surface fitting. The practical context will often give a guide. In general, it is best to avoid having to deal with a relationship whose behaviour in one region is radically different from that in another. A steep rise at the left-hand end of a curve, for example, can often best be treated by curve fitting in terms of $\log(x + c)$ with some suitable value of the constant c. According to the features exhibited in any particular case, transformation of either dependent variable or independent variable(s) or both may be beneficial. When there is a choice it is usually better to transform the independent variable(s): if the dependent variable is transformed, any weights associated with the data points must be adjusted. Thus if the f_r to be fitted have been obtained by a transformation $f = g(F)$ from original data values F_r, with weights W_r, for $r = 1, 2, \ldots, m$, we must take

$$w_r = W_r/(df/dF), \tag{4}$$

where the derivative is evaluated at F_r. Strictly, the transformation of F and the adjustment of weights are valid only when the data errors in the F_r are small compared with the range spanned by the F_r, but this is usually the case.

4 References

[1] Cox M G and Hayes J G (1973) Curve fitting: A guide and suite of algorithms for the non-specialist user *NPL Report NAC 26* National Physical Laboratory

[2] Renka R J (1988) Multivariate interpolation of large sets of scattered data *ACM Trans. Math. Software* **14** 139–148