5. Let $[a,b] = [-\pi,\pi]$, $f_1(x) = |x^3|$, $f_2(x) = \sin x$, and let the interpolation grid be uniform with size h. For both functions f_1 and f_2 implement on the computer the local spline (2.42), (2.58) with $s = 2$, $j = 1$, and the nonlocal cubic spline (2.60) with any type of boundary conditions: (2.63), (2.64), or (2.65). Demonstrate that for either type of spline, the convergence rate is $\mathscr{O}(h^2)$ for $f_1(x)$ and $\mathscr{O}(h^3)$ for $f_2(x)$.

2.4 Interpolation of Functions of Two Variables

The problem of reconstructing a function of continuous argument from its discrete table of values can be formulated in the multi-dimensional case as well, for example, when $f = f(x,y)$, i.e., when there are two independent variables. The principal objective remains the same as in the case of one dimension, namely, to build a procedure for (approximately) evaluating the function in between the given interpolation nodes. However, in the case of two variables one can consider a much wider variety of interpolation grids. All these grids basically fall into one of the two categories: structured or unstructured.

2.4.1 Structured Grids

Typically, structured grids on the (x,y) plane are composed of rectangular cells. In other words, the nodes (x_k, y_l), $k = 0, \pm 1, \ldots$, $l = 0, \pm 1, \ldots$, are obtained as intersections of the two families of straight lines: the vertical lines $x = x_k$, $k = 0, \pm 1, \ldots$, and the horizontal lines $y = y_l$, $l = 0, \pm 1, \ldots$. In so doing, we always assume that $\forall k : x_{k+1} > x_k$, and $\forall l : y_{l+1} > y_l$. In the literature, such grids are also referred to as rectangular or Cartesian. The grid sizes $h_k^{(x)} = x_{k+1} - x_k$ and $h_l^{(y)} = y_{l+1} - y_l$ may but do not have to be constant. In the case of constant size(s), the grid is called uniform or equally spaced (in the corresponding direction). The simplest example of a uniform two-dimensional grid is a grid with square cells: $h_k^{(x)} = h_l^{(y)} = \text{const}$.

To approximately compute the value of the function f at the point $(\bar{x}, \bar{y})$ that does not coincide with any of the nodes (x_k, y_l) of a structured rectangular grid, one can, in fact, use the apparatus of piecewise polynomial interpolation for the functions of one variable. To do so, we first select the parameters s (degree of interpolation) and j, as in Section 2.2. We also need to determine which cell of the grid contains the point of interest. Let us assume that $x_k < \bar{x} < x_{k+1}$ and $y_l < \bar{y} < y_{l+1}$ for some particular values of k and l. Then, we interpolate along the horizontal grid lines:

$$\bar{f}(\bar{x}, y_{l-j+i}) = P_s(\bar{x}, f(\,\cdot\,, y_{l-j+i}), x_{k-j}, x_{k-j+1}, \ldots, x_{k-j+s}), \quad i = 0, 1, \ldots, s.$$

and obtain the intermediate values $\bar{f}$. Having done that, we interpolate along the vertical grid lines and obtain the approximate value f:

$$f(\bar{x}, \bar{y}) \approx P_s(\bar{y}, \bar{f}(\bar{x}, \cdot\,), y_{l-j}, y_{l-j+1}, \ldots, y_{l-j+s}).$$

Clearly, the foregoing formulae can be used to approximate the function f at any point $(\bar{x},\bar{y})$ inside the rectangular grid cell $\{(x,y)|x_k < x < x_{k+1}, y_l < y < y_{l+1}\}$. For example, if we choose piecewise linear interpolation along x and y, i.e., $s=1$, then

$$f(\bar{x},\bar{y}) \approx f(x_k,y_l)\frac{(\bar{x}-x_{k+1})(\bar{y}-y_{l+1})}{(x_k-x_{k+1})(y_l-y_{l+1})} + f(x_{k+1},y_l)\frac{(\bar{x}-x_k)(\bar{y}-y_{l+1})}{(x_{k+1}-x_k)(y_l-y_{l+1})} + f(x_k,y_{l+1})\frac{(\bar{x}-x_{k+1})(\bar{y}-y_l)}{(x_k-x_{k+1})(y_{l+1}-y_l)} + f(x_{k+1},y_{l+1})\frac{(\bar{x}-x_k)(\bar{y}-y_l)}{(x_{k+1}-x_k)(y_{l+1}-y_l)}. \quad (2.83)$$

Note, however, that in general the degree of interpolation does not necessarily have to be the same for both dimensions. Also note that the procedure is obviously symmetric. In other words, it does not matter whether we first interpolate along x and then along y, as shown above, or the other way around.

The piecewise polynomial interpolation on the plane, built dimension-by-dimension on a rectangular grid as explained above, inherits the key properties of the one-dimensional interpolation. For example, if the function $f = f(x,y)$ is twice differentiable, with bounded second partial derivatives, then the interpolation error of formula (2.83) on a square-cell grid with size h will be $\mathcal{O}(h^2)$. For piecewise polynomial interpolation of a higher degree, the rate of convergence will accordingly be faster, provided that the interpolated function is sufficiently smooth. On the other hand, similarly to the one-dimensional case, the two-dimensional piecewise polynomial interpolation is also prone to the saturation by smoothness.

Again, similarly to the one-dimensional case, one can also construct a smooth piecewise polynomial interpolation in two dimensions. As before, this interpolation may be either local or nonlocal. Local splines that extend the methodology of Section 2.3.1 can be built on the plane dimension-by-dimension, in much the same way as the conventional piecewise polynomials outlined previously. Their key properties will be preserved from the one-dimensional case, specifically, the relation between their degree and smoothness, the minimum number of grid nodes in each direction, the convergence rate, and susceptibility to saturation (see [Rya75] for detail).

The construction of nonlocal cubic splines can also be extended to two dimensions; in this case the splines are called bi-cubic. On a domain of rectangular shape, they can be obtained by solving multiple tri-diagonal linear systems of type (2.62) along the x and y coordinate lines of the Cartesian grid. The approximation properties of bi-cubic splines remain the same as those of the one-dimensional cubic splines.

Similar constructions, standard piecewise polynomials, local splines, and nonlocal splines, are also available for the interpolation of multivariable functions (more than two arguments). We should emphasize, however, that in general the size of the tables that would guarantee a given accuracy of interpolation for a function of certain smoothness rapidly grows as the number of arguments of the function increases. The corresponding interpolation algorithms also become more cumbersome.