Multi-variate Polynomial Interpolation

Nested application of uni-variate methods

1. Bilinear interpolation on regular grids

Applications of interpolation methods often include multi-dimensional data and thus several variables asking for extension of the uni-variate interpolation methods. These extensions normally are a nested application of uni-variate methods and thus are rather natural.

Example 1.1: In image processing software tools upsizing a raster image often is performed by a two-dimensional interpolation. The figure below shows a small sample from the pixel raster, a so-called patch: The pixel coordinates (14,20), (14,21), (15, 20), (15,21) form the corners of a square grid with known real pixel values 91/255, 162/255, 210/255, 95/255. These correspond to color information, e.g. values of the red channel in a RGB-image (R = Red, G = Green, B = Blue).

Interpolating this 2-dimensional grid means finding a polynomial $p(x,y)$ in two variables $\{x,y\}$ such that p collocates with the color values in the four corner points of the grid.

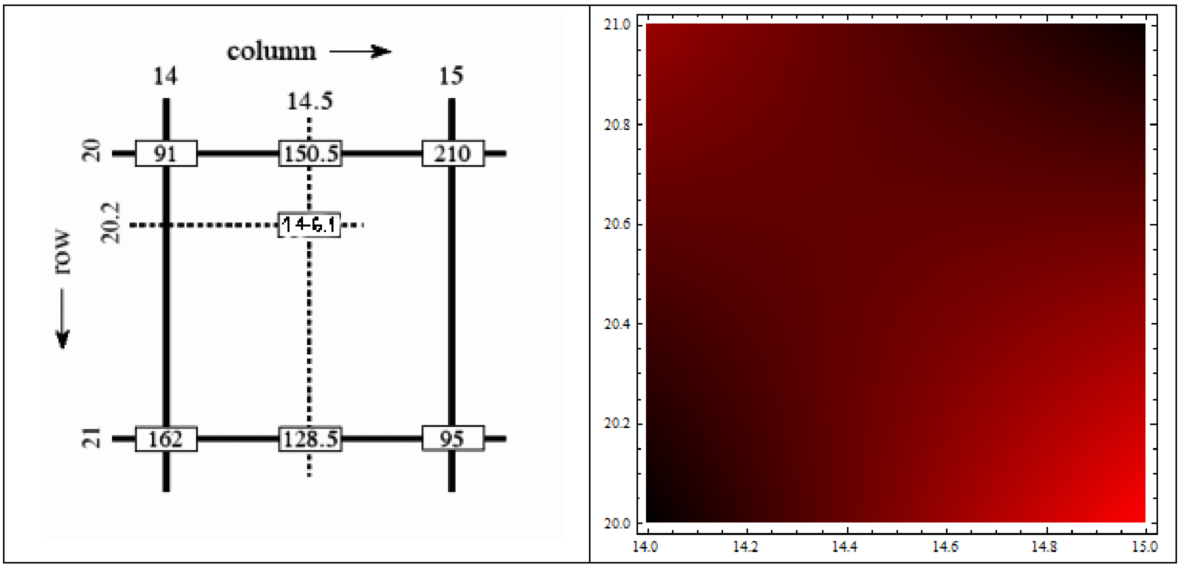

Figure 1.1ab: Square grid (patch) and its bilinear interpolation (red channel values).

Of course, the question arises how to represent the multi-variate polynomial $p(x,y)$ and how to determine the order of this polynomial. For the purpose of Example 1.1 we need four polynomial coefficients because there are four conditions. Preferring Newton polynomials leads to the following straightforward representation:

$$p(x,y) = a_{0,0}\pi_0(x)\pi_0(y) + a_{1,0}\pi_1(x)\pi_0(y) + a_{0,1}\pi_0(x)\pi_1(y) + a_{1,1}\pi_1(x)\pi_1(y)$$

$$p(x,y) = a_{0,0}1\cdot 1 + a_{1,0}(x-x_0)1 + a_{0,1}1(y-y_0) + a_{1,1}(x-x_0)(y-y_0) \qquad \textbf{(1)}$$

This kind of representation is called a 2-fold tensor product of the uni-variate basis $\{\pi_0,\pi_1\}$ in the variables $\{x,y\}$. It is also called bilinear interpolation because only uni-variate basis functions up to order 1 are used.

The figure below illustrates how 2-dimensional interpolation problems are solved by two consecutive uni-variate interpolation procedures. The xy-coordinate values have been rescaled to the unit square [0,1] x [0,1].

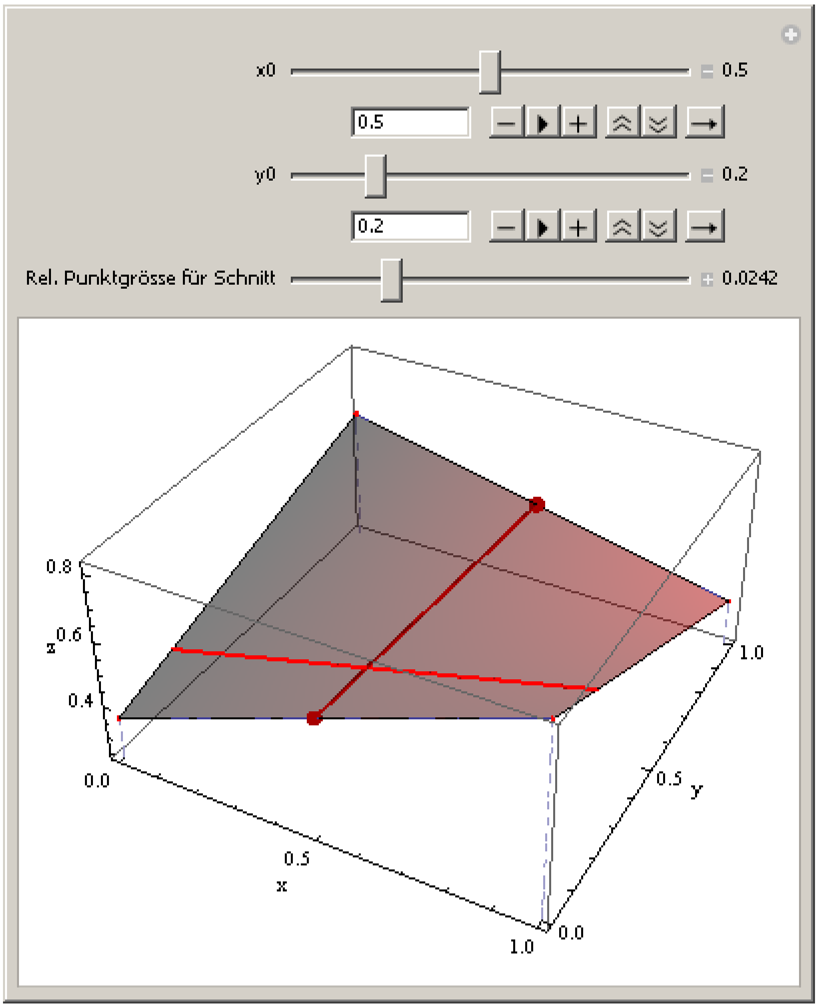

Figure 1.2: Illustration of 2-dimensional interpolation on a unit square. The red-gray surface is given by the polynomial `0.356863 + 0.466667 x + 0.278431 y - 0.729412 x y` in scaled xy-coordinates and otherwise by `-215.98 + 15.0549 x + 10.4902 y - 0.729412 x y`.

The computation proceeds along the following steps. Each step applies the method of divided differences.

(1) Uni-variate interpolation in the variable x for $y = y_0 = 20$ leading to

$$p(x, y_0) = p(x_0, y_0)\pi_0(x) + \frac{p(x_1, y_0) - p(x_0, y_0)}{x_1 - x_0}\pi_1(x) =$$

$$91/255 \cdot 1 + \frac{210/255 - 91/255}{15 - 14} \cdot (x - 14)$$

(2) Uni-variate interpolation in the variable x for $y = y_1 = 21$ leading to

$$p(x,y_1) = p(x_0,y_1)\pi_0(x) + \frac{p(x_1,y_1) - p(x_0,y_1)}{x_1 - x_0}\pi_1(x) =$$

$$162/255 \cdot 1 + \frac{95/255 - 162/255}{15-14}\cdot(x-14)$$

(3) Uni-variate interpolation in the variable y for an arbitrary value of x. This corresponds to interpolating the two high-lighted points in Fig. 1.2. From the steps (1) and (2) we compute the values $p(x, y_0)$ and $p(x, y_1)$ and then interpolate these values along the y-axis:

$$p(x,y) = p(x,y_0)\pi_0(y) + \frac{p(x,y_1) - p(x,y_0)}{y_1 - y_0}\pi_1(y) =$$

$$\left(91/255 \cdot 1 + \frac{210/255 - 91/255}{15-14}\cdot(x-14)\right)\cdot 1 +$$

$$\frac{162/255 \cdot 1 + \frac{95/255 - 162/255}{15-14}\cdot(x-14) - 91/255 \cdot 1 + \frac{210/255 - 91/255}{15-14}\cdot(x-14)}{21-20}\cdot(y-20)$$

The latter expression simplifies to `-215.98 + 15.0549 x + 10.4902 y - 0.729412 x y.`

From these computations it is seen that <u>multi-variate interpolation is a nested application of uni-variate interpolation along different coordinate axes</u>.

2. Bi-cubic interpolation on regular grids

Bi-cubic interpolation is a straightforward generalization of the method developped in the previous section. The term bi-cubic means that the basis polynomials are constituted by a 2-fold tensor product of the univariate cubic basis $\{\pi_0, \pi_1, \pi_2, \pi_3\}$ in the variables $\{x, y\}$.

$$\begin{aligned} p(x,y) = & \\ & a_{0,0}\pi_0(x)\pi_0(y) + a_{1,0}\pi_1(x)\pi_0(y) + a_{2,0}\pi_2(x)\pi_0(y) + a_{3,0}\pi_3(x)\pi_0(y) + \\ & a_{0,1}\pi_0(x)\pi_1(y) + a_{1,1}\pi_1(x)\pi_1(y) + a_{2,1}\pi_2(x)\pi_1(y) + a_{3,1}\pi_3(x)\pi_1(y) + \\ & a_{0,2}\pi_0(x)\pi_2(y) + a_{1,2}\pi_1(x)\pi_2(y) + a_{2,2}\pi_2(x)\pi_2(y) + a_{3,2}\pi_3(x)\pi_2(y) + \\ & a_{0,3}\pi_0(x)\pi_3(y) + a_{1,3}\pi_1(x)\pi_3(y) + a_{2,3}\pi_2(x)\pi_3(y) + a_{3,3}\pi_3(x)\pi_3(y) \end{aligned} \qquad \textbf{(2)}$$

From this it is seen that 16 conditions are required because there are 16 unknown coefficients.

2.1 Bi-cubic collocation

Bi-cubic collocation provides 16 function values on a <u>regular 4x4 grid</u>. These values constitute a set of 16 conditions (as required).

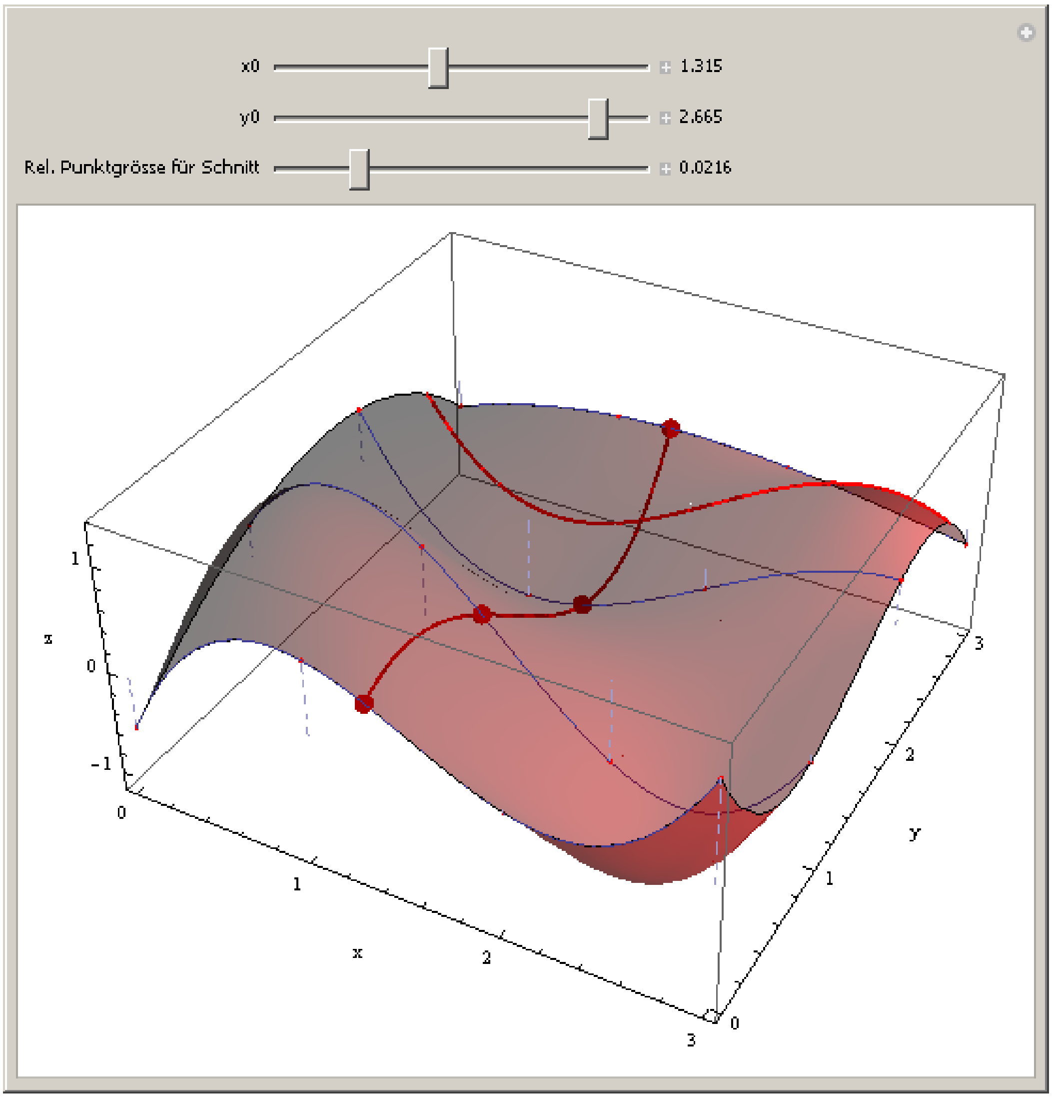

Figure 1.3: Illustration of 2-dimensional bi-cubic interpolation on a 4x4 grid.

The computation proceeds along the following uni-variate interpolation steps:

Steps (1)-(4): Uni-variate interpolations in the variable x for $y = y_0, y_1, y_2, y_3$ leading to 4 cubic polynomials $p(x, y_0), p(x, y_1), p(x, y_2), p(x, y_3)$.

Step (5): Uni-variate interpolation in the variable y for an arbitrary value of x. This corresponds to interpolating the four high-lighted points in Fig. 1.3. From the steps (1) to (4) we compute the values $p(x, y_0)$, $p(x, y_1)$, $p(x, y_2)$ and $p(x, y_3)$ and then interpolate these values along the y-axis.

This finally leads to the numerical bi-cubic polynomial

$$-0.581324 + 3.58038\, x - 2.92226\, x^2 + 0.63402\, x^3 + 1.15278\, y + 6.07605\, x\, y - 6.55837\, x^2\, y + 1.35823\, x^3\, y - 0.181566\, y^2 - 9.41581\, x\, y^2 + 8.59978\, x^2\, y^2 - 1.72666\, x^3\, y^2 - 0.0584666\, y^3 + 2.35529\, x\, y^3 - 2.04092\, x^2\, y^3 + 0.402331\, x^3\, y^3$$

in expanded form. The details of the computations are omitted.

*2.2 Bi-cubic osculation

When working with unit-square grids (as in the section on bi-linear interpolation and typical for image processing) 12 conditions on the derivative values must be added to the 4 values in the 4 corners $\{(x_0,y_0),(x_0,y_1),(x_1,y_0),(x_1,y_1)\}$.

Values for the first-order partial derivatives ...

$$\frac{\partial}{\partial x}p(x_0,y_0),\frac{\partial}{\partial x}p(x_0,y_1),\frac{\partial}{\partial x}p(x_1,y_0),\frac{\partial}{\partial x}p(x_1,y_1)$$
$$\frac{\partial}{\partial y}p(x_0,y_0),\frac{\partial}{\partial y}p(x_0,y_1),\frac{\partial}{\partial y}p(x_1,y_0),\frac{\partial}{\partial y}p(x_1,y_1)$$

… and the second-order partial derivatives

$$\frac{\partial^2}{\partial x\partial y}p(x_0,y_0),\frac{\partial^2}{\partial x\partial y}p(x_0,y_1),\frac{\partial^2}{\partial x\partial y}p(x_1,y_0),\frac{\partial^2}{\partial x\partial y}p(x_1,y_1)$$

constitute such a set of 12 conditions. Generally, the partial derivatives are approximated by finite differences of data values neighbouring the corners.
The 4 cubic boundary curves (of the polynomial surface)

$p(x,y_0),\ p(x,y_1),\ p(x_0,y)$ and $p(x_1,y)$

can be computed by Hermite interpolation using divided differences and the informations on the first-order partial derivatives.

3. Multi-variate collocation on regular grids[1]

Multi-variate collocation problems on regular grids are solved along the same lines as the 2-dimensional collocation problems described in the sections above.

The x-arguments now are vectors $\vec{x}_0,\vec{x}_1,\ldots,\vec{x}_{n-1},\vec{x}_n$ of dimension d and thus we need basis functions which take vectors as inputs (i.e. functions in several variables). For practical purposes most widely used are tensor products of uni-variate basis functions.
If $g_0,g_1,...,g_m=\{g_j\}_{j=0,...,m}$ is a uni-variate basis then the d-fold tensor product

$$\boxed{\sum_{j_1=0}^{m_1}\sum_{j_2=0}^{m_2}\cdots\sum_{j_d=0}^{m_d}a_{j_1,j_2,...,j_d}\,g_{j_1}(x^{(1)})g_{j_2}(x^{(2)})\cdots g_{j_d}(x^{(d)})} \qquad \textbf{(3)}$$

defines a basis of $m_1\times m_2\times\cdots\times m_d$ functions in d variables $x^{(1)},x^{(2)},\ldots,x^{(d-1)},x^{(d)}$.

[1] Interpolation on irregular (non-grid) data is solved by triangulation and working with triangle patches (elements). For the solution of the interpolation problem on a triangle patch barycentric coordinates are very useful.

Name of basis functions	**3d-formulas**
Polynomials	$\left\{x^{(1)j} x^{(2)k} x^{(3)l} \middle| j,k,l \in N_0\right\}$ (standard monomials) $\left\{\pi_j(x^{(1)})\pi_k(x^{(2)}\pi_l(x^{(3)}) \middle| j,k,l \in N_0\right\}$ (Newton polynomials) $\left\{T_j(x^{(1)})T_k(x^{(2)}T_l(x^{(3)}) \middle| j,k,l \in N_0\right\}$ (Chebyshev polynomials, $x^{(1)}, x^{(2)}, x^{(3)} \in (-1,1)$)

Table 1.1: A few basis functions in 3 variables. The generalizations to more than 3 variables are straightforward.

Typical applications are tri-linear and tri-cubic interpolation. The latter often is used on unit cubic grids with informations on partial derivatives.