

Spatial interpolation (1/2)


Lesson 04*

1	What is a good interpolation method for terrains?	2
2	Fitting polynomials	2
2.1	One global function	2
2.2	Splines: piecewise polynomials	3
3	Weighted-average methods	4
3.1	Nearest Neighbour Interpolation	4
3.2	Inverse distance weighting (IDW)	4
3.3	Linear Interpolation in a TIN	5
3.4	Natural Neighbour Interpolation	6
3.5	Bilinear Interpolation	8
4	Notes and comments	9

Given a set S of points p_i in \mathbb{R}^2 (also called samples or data points in the following) to which an attribute a_i is attached, spatial interpolation is the procedure used to estimate the value of the attribute at an unsampled location x . Its goal is to find a function $f(x, y)$ that fits (pass through, or close to, all the points in S) as well as possible. There is an infinity of such functions, some are global and some are piecewise, and thus we the aim is to find one that is best suited for the kind of datasets used as input. Interpolation is based on *spatial autocorrelation*, that is the attribute of two points close together in space is more likely to be similar than that of two points far from each other.

It should be noticed that the natural spatial extent of a set of sample is its *convex hull*, and that an estimation outside this convex hull is *extrapolation*. Extrapolating implies that more uncertainty is attached to the estimated value.

Spatial interpolation methods are crucial in the visualisation process (eg generation of contours lines), for the conversion of data from one format to another (eg from scattered points to raster), to have a better understanding of a dataset, or simply to identify ‘bad’ samples. The result of interpolation—usually a surface that represents the terrain—must be as accurate as possible because it often forms the basis for spatial analysis, for example runoff modelling or visibility analysis. Although interpolation helps in creating three-dimensional surfaces, in the case of terrains it is intrinsically a two-dimensional operation because only the (x, y) coordinates of each sample are used, and the elevation is the dependent attribute. Notice that the

* Hugo Ledoux, Ravi Peters, Ken Arroyo Ohori. This work is licensed under a Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>) (last update: December 11, 2018)

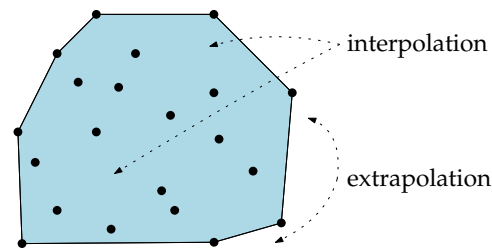


Figure 1: Spatial interpolation and extrapolation.

attribute used need not be only elevation, for other GIS applications one could use the spatial interpolation methods below for for instance rainfall amounts, percentage of humidity in the soil, maximum temperature, etc. Spatial interpolation in 3D is also possible (but out of scope for this course), in that case there are 3 independent variables (x, y, z) and one dependent variable, for instance the temperature of the sea at different depth, or the concentration of a certain chemical in the ground.

1 What is a good interpolation method for terrains?

The essential properties of an 'ideal' interpolation method for bivariate geoscientific datasets are as follows:

1. **exact**: the interpolant must 'honour' the data points, or 'pass through' them.
2. **continuous**: a single and unique value must be obtained at each location. This is called a C^0 interpolant in mathematics (see Figure 2).
3. **smooth**: it is desirable for some applications to have a function for which the first or second derivative is possible everywhere; such functions are respectively referred to as C^1 and C^2 interpolants.
4. **local**: the interpolation function uses only some neighbouring samples to estimate the value at a given location. This ensures that a sample with a gross error will not propagate its error to the whole interpolant.
5. **adaptability**: the function should give realistic results for anisotropic data distributions and/or for datasets where the data density varies greatly from one location to another.
6. **computationally efficient**: it should be possible to implement the method and get an efficient result. Efficient is of course subjective. For a student doing this course, efficiency might mean that the method generates a result in matter of minutes or an hour on a laptop, for the homework dataset. For a mapping agency, running a process for a day on a supercomputer for a whole country might be efficient.
7. **automatic**: the method must require as little input as possible from the user, ie it should not rely on user-defined parameters that require *a priori* knowledge of the dataset.

2 Fitting polynomials

2.1 One global function

We know that if we have n points in S in \mathbb{R}^3 (the samples are lifted to their elevation), there is one polynomial of degree at most $n - 1$.

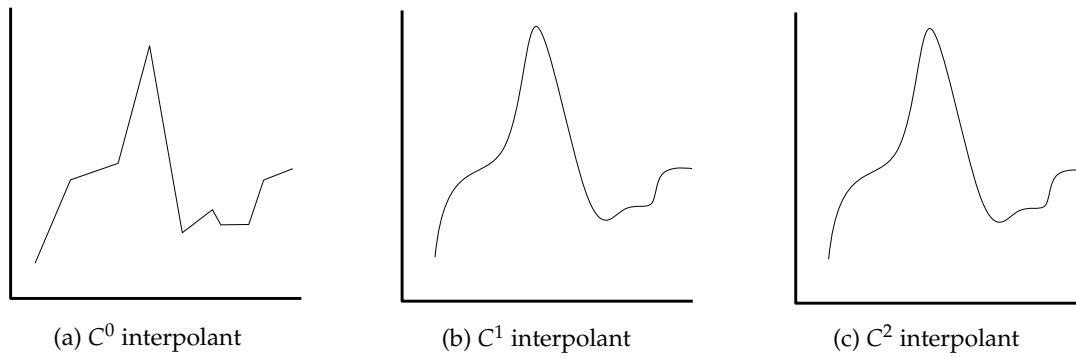


Figure 2: C^0 interpolant is a function that is continuous but the first derivative is not possible at certain locations; C^1 interpolant has its first derivative possible everywhere; C^2 interpolant has its second derivative possible everywhere (this one is more difficult to draw).

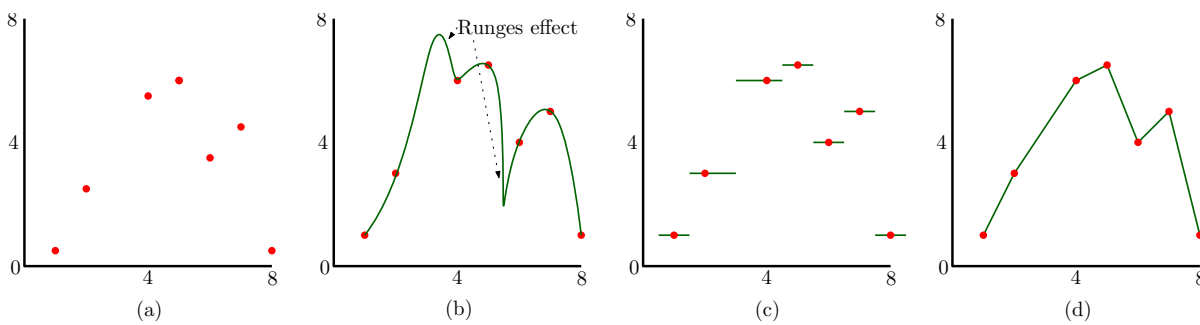


Figure 3: A few of the interpolation methods shown for a 1D dataset. (a) Input sample points. (b) Polynomial fitting, and the Runge's effect shown. (c) Natural neighbour. (d) Linear interpolation in TIN.

This interpolant will be exact, continuous and smooth (at least C^2). However, it will not be local (which is problematic for terrains), and finding the polynomial of a high degree for large datasets might be impossible (or take a lot of time).

The biggest concern polynomials is probably that while the interpolant is exact (the surface passes through the sample points), higher-degree polynomials can oscillate between the samples and 'overshoot', ie be (far) outside the minimum or maximum z values of the the set S . This is known as the Runge's phenomenon in numerical analysis, and is shown in Figure 3.

2.2 Splines: piecewise polynomials

Splines are piecewise polynomials, each piece is connected to its neighbouring piece in a *smooth* manner: along the edges and at the data points the function is usually still C^1 or C^2 (in other words, where 2 or more polynomials connect, they have the same values for their tangents).

In practice, for terrain modelling, splines are preferred over one polynomial function because of the reasons mentioned above (mostly Runge's effect) and because computing the polynomial for large datasets is very inefficient. The polynomials used in each piece of the subdivision is usually of low degree (≤ 5)

There are several types of splines (and variation of them, such as Bézier), and most of them are not suited for terrains. The most used spline in practice seems to be the *regularised spline with tension* (RST), where the dataset is decomposed into square pieces of a certain size. The Runge's effect (also called *overshoots*) are eliminated (since the degree is low), and the tension

parameter can be tuned to obtain an interpolant that is smooth. The method is available in the GRASS GIS.

3 Weighted-average methods

The five interpolation methods discussed in this section are *weighted-average methods*. These are methods that use a subset of the sample points, to which a weight (importance) are assigned, to estimate the value of the dependent variable. The interpolation function f of such methods have the following form:

$$f(x) = \frac{\sum_{i=1}^k w_i(x) a_i}{\sum_{i=1}^k w_i(x)} \quad (1)$$

where $w_i(x)$ is the weight of each neighbouring data point p_i (with respect to the interpolation location x) used in the interpolation process, and a_i the attribute of p_i . A neighbour p_i here is a sample point that is used to estimate the value of location x . In the context of terrain modelling, the attribute a is elevation.

3.1 Nearest Neighbour Interpolation

Nearest neighbour, or closest neighbour, is a simple interpolation method: the value of an attribute at location x is simply assumed to be equal to the attribute of the nearest data point. This data point gets a weight of 1.0 . Given a set S of data points, if interpolation is performed with this method at many locations close to each other, the result is the Voronoi diagram (VD) of S , where all the points inside a Voronoi cell have the same value.

Although the method possesses many of the desirable properties (it is exact, local and can handle anisotropic data distributions), the reconstruction of continuous fields can not realistically be done using it since it fails lamentably properties 2 and 3. The interpolation function is indeed discontinuous at the border of cells; if the location x is directly on an edge or vertex of the VD(S), then which value should be returned?

3.2 Inverse distance weighting (IDW)

Inverse distance weighting (IDW)—also called inverse distance to a power, or distance-based methods—is a family of interpolation methods using distance to identify the neighbours used, and to assign them weights. IDW is probably the most known interpolation method and it is widely used in many disciplines. As shown in Figure 5a, in two dimensions it often uses a

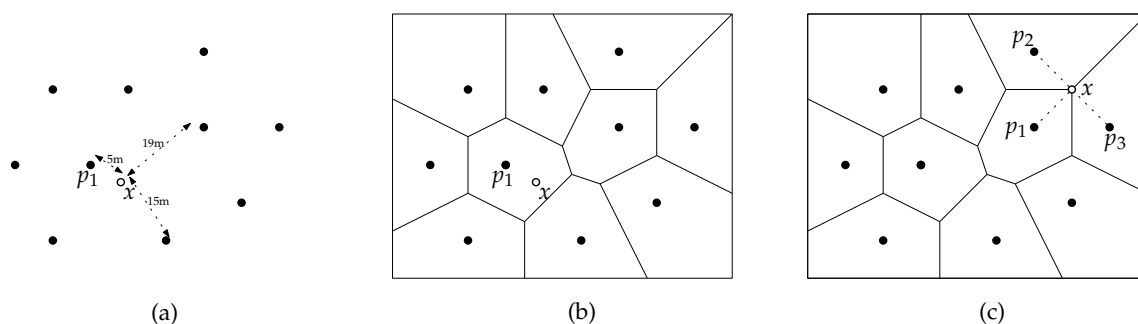


Figure 4: **(a)** Nearest neighbour: the estimated value at x is that of the closest data point. **(b)** the Voronoi diagram can be used. **(c)** Ambiguity because p_1 , p_2 , and p_3 are equidistant from x ; this causes discontinuities in the resulting surface.

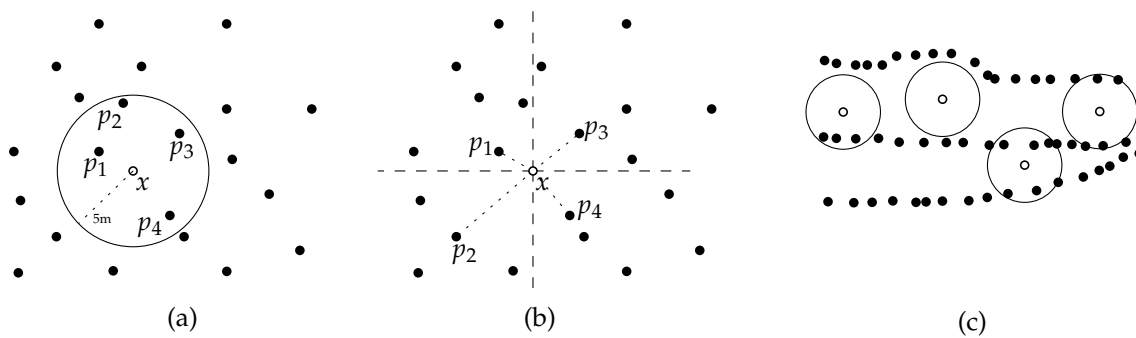


Figure 5: **(a)** IDW interpolation with a searching circle, and the weight assigned to each neighbour used in the estimation. **(b)** IDW by choosing the closest neighbour in each quadrant. **(c)** It has (serious) problems with datasets whose distribution of samples is anisotropic.

‘searching circle’, whose radius is user-defined, to select the data points p_i involved in the interpolation at location x . It is also possible to select for instance the 10 or 15 closest data points, or do that according to certain directions (ie you can select for example 3 data points in each quadrant; Figure 5b shows the case where the closest in each quadrant is used).

The weight $w_i(x)$ assigned to each p_i for a location x is:

$$w_i(x) = |xp_i|^{-h} \quad (2)$$

where h defines the power to be used, and $|ab|$ is the distance between a and b . The power h is typically 2, but other weights, such as 3, can also be used. A very high power, say 5, will assign very little importance to points that are far away.

It should be emphasised that the size of the radius of the searching circle influences greatly the result of the interpolation: a very big radius means that the resulting surface will be smooth or ‘flattened’; on the other hand, a radius that is too small might have dramatic consequences if for example no data points are inside the circle (Figure 5c shows one example). A good knowledge of the dataset is thus required to select this parameter.

This method has many flaws when the data distribution varies greatly in one dataset because a fixed-radius circle will not necessarily be appropriate everywhere in the dataset. Figure 5c shows one example where one circle, when used with a dataset extracted from contour lines, clearly gives erroneous results at some locations. The major problem with the method comes from the fact that the criterion, for both selecting data points and assigning them a weight, is one-dimensional and therefore does not take into account the spatial distribution of the data points close to the interpolation location.

IDW is exact, local, and can be implemented in an efficient manner. However, as mentioned above, there are cases where it might not be continuous (nor smooth), it suffers from the distribution of sample points, and we cannot claim that it is automatic since finding the correct parameters for the search radius is usually a trial-and-error task. If the closest data points in each quadrant are used, then the method can be made automatic and continuous.

3.3 Linear Interpolation in a TIN

This method is popular for terrain modelling applications and is based on a triangulation of the data points. As is the case for the VD, a triangulation is a piecewise subdivision (tessellation) of the plane, and in the context of interpolation a linear function is assigned to each piece (each triangle). Interpolating at location x means first finding inside which triangle x lies, and then the height is estimated by linear interpolation on the 3D plane defined by the

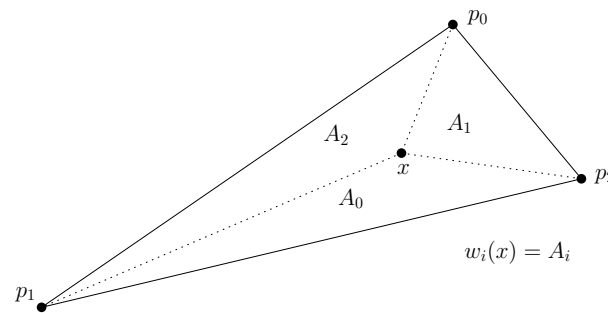


Figure 6: Barycentric coordinates. A_i defines the area of a triangle.

three vertices forming the triangle (the samples are lifted to their elevation value). The number of samples used in the interpolation is therefore always 3, and their weight is based on the barycentric value (see below). To obtain satisfactory results, this method is usually used in 2D with a Delaunay triangulation because, among all the possible triangulations of a set of points in the plane, it maximizes the minimum angle of each triangle.

The method is exact, continuous, local, adaptative, efficient, and automatic. Only the property #3 is not fulfilled (at the edges of the triangles).

Barycentric coordinates. The linear interpolation in a triangle can be efficiently implemented by using barycentric coordinates, which are local coordinates defined within a triangle. Referring to Figure 6, any point x inside a triangle $p_1p_2p_3$ can be represented as a linear combination of the 3 vertices:

$$x = w_0p_0 + w_1p_1 + w_2p_2$$

and

$$w_0 + w_1 + w_2 = 1$$

The coefficients u_i are the barycentric coordinates of the point x with respect to the triangle $p_1p_2p_3$. Finding the coefficients w_0 , w_1 , and w_2 can be done by solving a system of linear equations. If we subtract p_2 from x , and we use $w_2 = 1 - w_0 - w_1$, we obtain

$$x - p_2 = w_0(p_0 - p_2) + w_1(p_1 - p_2)$$

We obtain 2 vectors ($p_0 - p_2$ and $p_1 - p_2$), which represent 2 edges of the triangle. This equation can be solved and we find that the 3 coefficients are equal of the area of the 3 triangle subdividing the original triangle (as shown in Figure 6).

Higher-order function in each triangle. Notice that it is possible to modify the linear function inside each triangle by a higher-order function. As as the case for splines, there are *several* ways to achieve this, and the details of these is out of scope for this course. These methods are usually used more for finite element analysis where the flow of a certain fluid (eg wind) around or through a mechanical piece is studied.

Most methods would define a cubic Bézier polynomial inside each triangle (which is C^1), and then ensure that the function is C^1 along the edges and at the 3 vertices of the triangles. To achieve this the normals of each vertex is calculated by averaging the normals of the incident triangles, and the normal along an edge is computed similarly with the 2 incident triangles.

3.4 Natural Neighbour Interpolation

This is a method based on the Voronoi diagram for both selecting the data points involved in the process, and assigning them a weight. It uses two VDs: one for the set S of data points, and

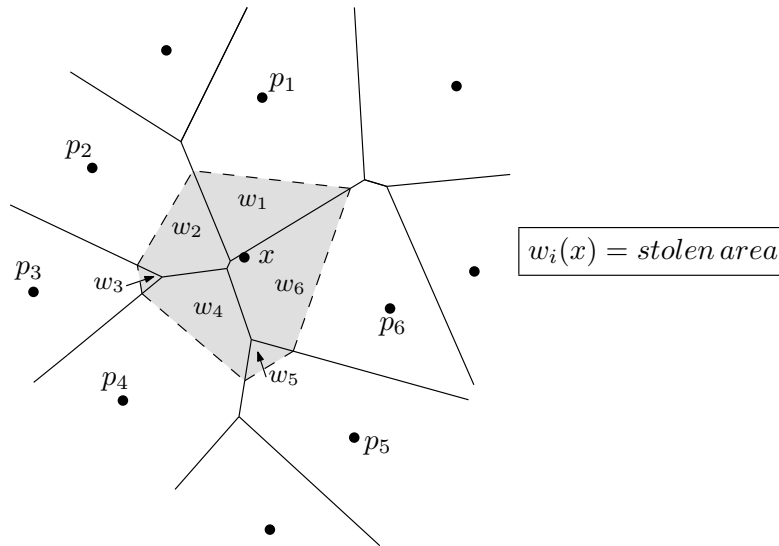


Figure 7: Natural neighbour coordinates in 2D for the interpolation point x . The shaded polygon is \mathcal{V}_x .

another one where a point x is inserted at the estimation location. The insertion of x modifies *locally* a $\text{VD}(S)$: the Voronoi cell \mathcal{V}_x of x ‘steals’ some parts of some Voronoi cells of $\text{VD}(S)$, as shown in Figure 7. This idea forms the basis of natural neighbour coordinates, which define quantitatively the amount \mathcal{V}_x steals from each of its natural neighbours. Let \mathcal{D} be the $\text{VD}(S)$, and $\mathcal{D}^+ = \mathcal{D} \cup \{x\}$. The Voronoi cell of a point p in \mathcal{D} is defined by \mathcal{V}_p , and \mathcal{V}_p^+ is its cell in \mathcal{D}^+ . The natural neighbour coordinate of x with respect to a point p_i is

$$w_i(x) = \frac{\text{Area}(\mathcal{V}_{p_i} \cap \mathcal{V}_x^+)}{\text{Area}(\mathcal{V}_x^+)} \tag{3}$$

where $\text{Area}(\mathcal{V}_{p_i})$ represents the area of \mathcal{V}_{p_i} . For any x , the value of $w_i(x)$ will always be between 0 and 1: 0 when p_i is not a natural neighbour of x , and 1 when x is exactly at the same location as p_i . A further important consideration is that the sum of the areas stolen from each of the k natural neighbours is equal to $\text{Area}(\mathcal{V}_x^+)$, in other words:

$$\sum_{i=1}^k w_i(x) = 1. \tag{4}$$

Therefore, the higher the value of $w_i(x)$ is, the stronger is the ‘influence’ of p_i on x . The natural neighbour coordinates are influenced by both the distance from x to p_i and the spatial distribution of the p_i around x .

Natural neighbour interpolation is based on the natural neighbour coordinates. The points used to estimate the value of an attribute at location x are the natural neighbours of x , and the weight of each neighbour is equal to the natural neighbour coordinate of x with respect to this neighbour. If we consider that each data point in S has an attribute a_i (its elevation), the natural neighbour interpolation function is

$$f(x) = \sum_{i=1}^k w_i(x) a_i \tag{5}$$

where $f(x)$ is the interpolated function value at the location x .

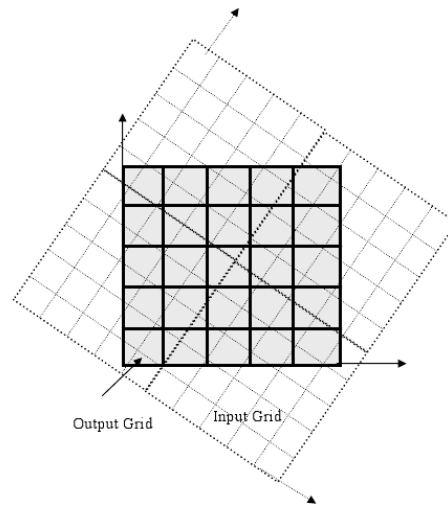


Figure 8: Resampling of an input grid, the output grid has different orientation.

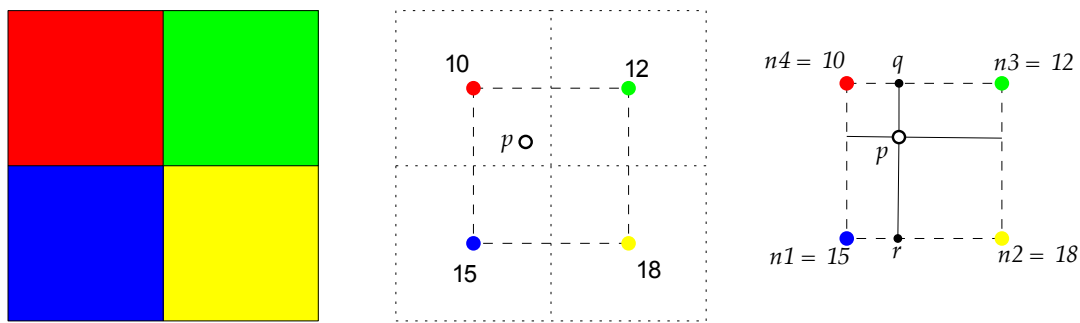


Figure 9: Bilinear interpolation.

The natural neighbour interpolant possesses all the wished properties from above, except that the first derivative is undefined at the data points. Its main disadvantage is that its implementation is rather complex, and obtaining an efficient one is not simple and involves complex manipulation of the VD.

3.5 Bilinear Interpolation

When one wants to know the value of the elevation at a location p , she can simply look at the value of the pixel (which is equivalent to using nearest neighbour interpolation), but this method has many drawbacks, for example when one needs to *resample* a grid. Resampling means transforming an input grid so that the resolution and/or the orientation are different, see Figure 8.

Bilinear interpolation has been shown to give better results. The method, which can be seen as an ‘extension’ of linear interpolation for raster data, performs linear interpolation in one dimension (say along the x axis), and then in the other dimension (y). Here one has to be careful about the meaning of a grid: does the value of a pixel represent the value of the whole pixel? or was the grid constructed by sampling the values at the middle of each pixel? In most cases, unless metadata are available, it is not known. But in the context of terrain modelling, we can assume that the value of a pixel represents the value at the centre of the pixel.

Suppose we have 4 adjacent pixels, each having an elevation, as in Figure 9. Bilinear interpolation uses the 4 centres to perform the interpolation at location $p = (p_x, p_y)$; it is thus a weighted-average method because the 4 samples are used, and their weight is based on the

linear interpolation, as explained below. We need to linearly interpolate the values at locations q and r with linear interpolation, and then linearly interpolate along the y axis with these values. Notice that the result is independent of the order of interpolation: we could start with interpolation along the y axis and then the x axis and we would get the same result. For the case in Figure 9, the calculation would go as follows:

$$f(q) = \frac{p_x - n4_x}{n3_x - n4_x} \times (n3_z - n4_z) + n4_z$$

$$f(r) = \frac{p_x - n1_x}{n2_x - n1_x} \times (n2_z - n1_z) + n1_z$$

$$f(p) = \frac{p_y - r_y}{q_y - r_y} * (q_z - r_z) + r_z$$

The interpolant can be expressed as follows:

$$f(x) = a_0 + a_1x + a_2y + a_3xy \quad (6)$$

where a_i are 4 unknown that can be determined by solving 4 equations (one for each grid node). If we assume that the lower-left pixel is at location $(0,0)$ and the top-right one at $(1,1)$, then the 4 coefficients are:

$$\begin{aligned} a_0 &= n1_z \\ a_1 &= n2_z - n1_z \\ a_2 &= n3_z - n1_z \\ a_3 &= n1_z - n2_z - n3_z + n4_z \end{aligned}$$

4 Notes and comments

Watson (1992), in his authoritative book, lists the essential properties of an ‘ideal’ interpolation method for bivariate geoscientific datasets; we have added *computationally efficient* and *automatic* to the list.

Mitasova and Mitas (1993) gives a full description of the regularised splines with tension (RST) interpolation method. This method has also been implemented in the open-source GIS GRASS.

For a discussion about influence of the power in IDW on the resulting surface, please see Watson (1992).

The natural neighbour interpolation method is also called Sibson’s interpolation, after the name of the inventor (Sibson, 1981).

To obtain a continuous interpolant even at the data points, Sibson uses the weights defined in Equation 3 in a quadratic equation where the gradient at x is considered. Other ways to remove the discontinuities at the data points have been proposed: Watson (1992) describes different methods to estimate the gradient at x and how to incorporate it in Equation 5; and Gold (1989) proposes to modify the weight of each p_i with a simple hermitian polynomial so that, as x approaches p_i , the derivative of $f(x)$ approaches 0. Modifying Equation 5 to obtain a continuous function can yield very good results in some cases, but with some datasets the resulting surface can contain unwanted effects. Different datasets require different methods and parameters, and, for this reason, modifications should be applied with great care.

The description of the barycentric coordinates is mostly taken from Eberly (2018).

The construction of a polynomial inside each triangle of a TIN can be done with several methods. The simplest method is the Clough-Tocher method (Clough and Tocher, 1965; Farin, 1985). It splits each triangle into 3 sub-triangles (by inserting a temporary point at the centroid of the triangle) and a cubic function is built over each.

References & further reading

- Clough R and Tocher J (1965). Finite element stiffness matrices for analysis of plates in bending. In *Proceedings of Conference on Matrix Methods in Structural Analysis*.
- Eberly D (2018). C^1 quadratic interpolation of meshes. <https://www.geometrictools.com/Documentation/C1QuadraticInterpolation.pdf>.
- Farin G (1985). A modified Clough-Tocher interpolant. *Computer Aided Geometric Design*, 2(1–3):19–27.
- Gold CM (1989). Surface interpolation, spatial adjacency and GIS. In Raper J, editor, *Three Dimensional Applications in Geographic Information Systems*, pages 21–35. Taylor & Francis.
- Mitasova H and Mitas L (1993). Interpolation by regularized spline with tension: I. Theory and implementation. *Mathematical Geology*, 25:641–655.
- Sibson R (1981). A brief description of natural neighbour interpolation. In Barnett V, editor, *Interpreting Multivariate Data*, pages 21–36. Wiley, New York, USA.
- Watson DF (1992). *Contouring: A guide to the analysis and display of spatial data*. Pergamon Press, Oxford, UK.