

# Interpolation

From Wikipedia, the free encyclopedia

In the mathematical field of numerical analysis, **interpolation** is a method of constructing new data points within the range of a discrete set of known data points.

In engineering and science, one often has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to **interpolate** (i.e., estimate) the value of that function for an intermediate value of the independent variable.

A different problem which is closely related to interpolation is the approximation of a complicated function by a simple function. Suppose the formula for some given function is known, but too complex to evaluate efficiently. A few known data points from the original function can be used to create an interpolation based on a simpler function. Of course, when a simple function is used to estimate data points from the original, interpolation errors are usually present; however, depending on the problem domain and the interpolation method used, the gain in simplicity may be of greater value than the resultant loss in precision.

In the examples below if we consider  $x$  as a topological space and the function  $f$  forms a different kind of Banach spaces then the problem is treated as "interpolation of operators". The classical results about interpolation of operators are the Riesz–Thorin theorem and the Marcinkiewicz theorem. There are also many other subsequent results.

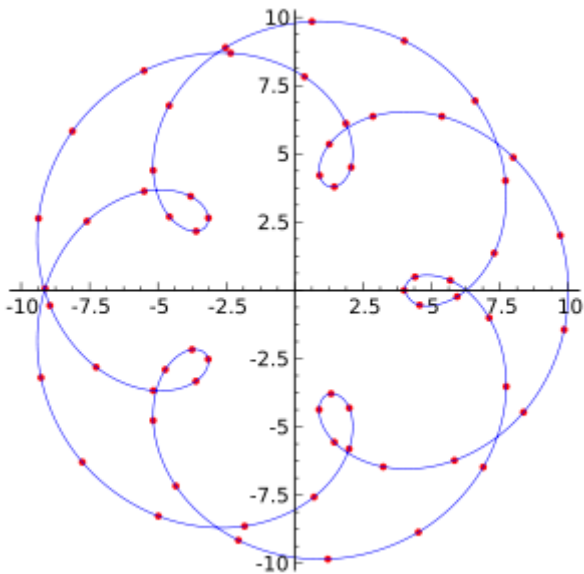
## Contents

- 1 Example
  - 1.1 Piecewise constant interpolation
  - 1.2 Linear interpolation
  - 1.3 Polynomial interpolation
  - 1.4 Spline interpolation
- 2 Interpolation via Gaussian processes
- 3 Other forms of interpolation
- 4 In higher dimensions
- 5 Interpolation in digital signal processing
- 6 Related concepts
- 7 See also
- 8 References
- 9 External links

## Example

For example, suppose we have a table like this, which gives some values of an unknown function  $f$ .

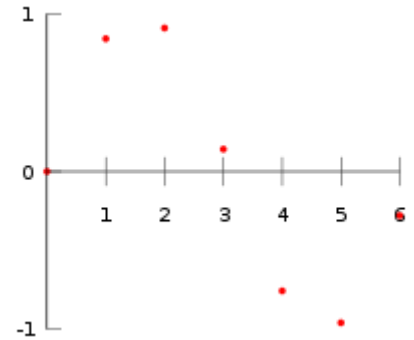
$x$	$f(x)$
0	0
1	0.8415
2	0.9093
3	0.1411
4	−0.7568
5	−0.9589
6	−0.2794



An interpolation of a finite set of points on an epitrochoid. The points through which the curve is splined are red; the blue curve connecting them is interpolation.

Interpolation provides a means of estimating the function at intermediate points, such as  $x = 2.5$ .

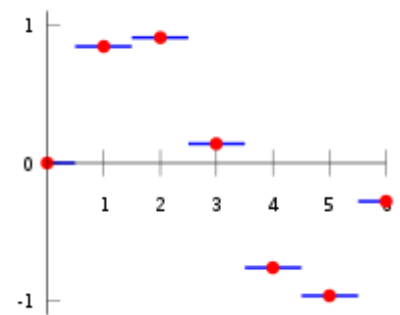
There are many different interpolation methods, some of which are described below. Some of the concerns to take into account when choosing an appropriate algorithm are: How accurate is the method? How expensive is it? How smooth is the interpolant? How many data points are needed?



Plot of the data points as given in the table.

## Piecewise constant interpolation

The simplest interpolation method is to locate the nearest data value, and assign the same value. In simple problems, this method is unlikely to be used, as linear interpolation (see below) is almost as easy, but in higher-dimensional multivariate interpolation, this could be a favourable choice for its speed and simplicity.



Piecewise constant interpolation, or nearest-neighbor interpolation

## Linear interpolation

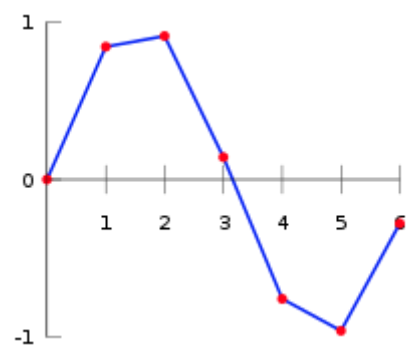
One of the simplest methods is linear interpolation (sometimes known as *lerp*). Consider the above example of estimating  $f(2.5)$ . Since 2.5 is midway between 2 and 3, it is reasonable to take  $f(2.5)$  midway between  $f(2) = 0.9093$  and  $f(3) = 0.1411$ , which yields 0.5252.

Generally, linear interpolation takes two data points, say  $(x_a, y_a)$  and  $(x_b, y_b)$ , and the interpolant is given by:

$$y = y_a + (y_b - y_a) \frac{x - x_a}{x_b - x_a} \text{ at the point } (x, y)$$

$$\frac{y - y_a}{y_b - y_a} = \frac{x - x_a}{x_b - x_a}$$

$$\frac{y - y_a}{x - x_a} = \frac{y_b - y_a}{x_b - x_a}$$



Plot of the data with linear interpolation superimposed

This previous equation states that the slope of the new line between  $(x_a, y_a)$  and  $(x, y)$  is the same as the slope of the line between  $(x_a, y_a)$  and  $(x_b, y_b)$

Linear interpolation is quick and easy, but it is not very precise. Another disadvantage is that the interpolant is not differentiable at the point  $x_k$ .

The following error estimate shows that linear interpolation is not very precise. Denote the function which we want to interpolate by  $g$ , and suppose that  $x$  lies between  $x_a$  and  $x_b$  and that  $g$  is twice continuously differentiable. Then the linear interpolation error is

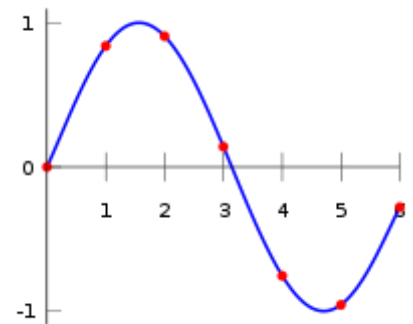
$$|f(x) - g(x)| \leq C(x_b - x_a)^2 \quad \text{where} \quad C = \frac{1}{8} \max_{r \in [x_a, x_b]} |g''(r)|.$$

In words, the error is proportional to the square of the distance between the data points. The error in some other methods, including polynomial interpolation and spline interpolation (described below), is proportional to higher powers of the distance between the data points. These methods also produce smoother interpolants.

## Polynomial interpolation

Polynomial interpolation is a generalization of linear interpolation. Note that the linear interpolant is a linear function. We now replace this interpolant with a polynomial of higher degree.

Consider again the problem given above. The following sixth degree polynomial goes through all the seven points:



Plot of the data with polynomial interpolation applied

$$f(x) = -0.0001521x^6 - 0.003130x^5 + 0.07321x^4 - 0.3577x^3 + 0.2255x^2 + 0.9038x.$$

Substituting  $x = 2.5$ , we find that  $f(2.5) = 0.5965$ .

Generally, if we have  $n$  data points, there is exactly one polynomial of degree at most  $n-1$  going through all the data points. The interpolation error is proportional to the distance between the data points to the power  $n$ . Furthermore, the interpolant is a polynomial and thus infinitely differentiable. So, we see that polynomial interpolation overcomes most of the problems of linear interpolation.

However, polynomial interpolation also has some disadvantages. Calculating the interpolating polynomial is computationally expensive (see computational complexity) compared to linear interpolation. Furthermore, polynomial interpolation may exhibit oscillatory artifacts, especially at the end points (see Runge's phenomenon).

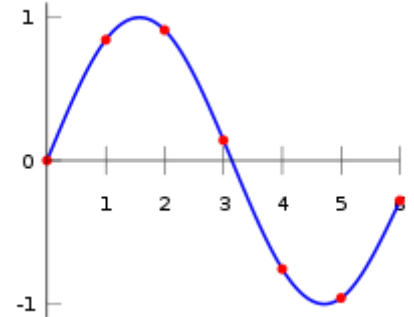
Polynomial interpolation can estimate local maxima and minima that are outside the range of the samples, unlike linear interpolation. For example, the interpolant above has a local maximum at  $x \approx 1.566$ ,  $f(x) \approx 1.003$  and a local minimum at  $x \approx 4.708$ ,  $f(x) \approx -1.003$ . However, these maxima and minima may exceed the theoretical range of the function—for example, a function that is always positive may have an interpolant with negative values, and whose inverse therefore contains false vertical asymptotes.

More generally, the shape of the resulting curve, especially for very high or low values of the independent variable, may be contrary to commonsense, i.e. to what is known about the experimental system which has generated the data points. These disadvantages can be reduced by using spline interpolation or restricting attention to Chebyshev polynomials.

## Spline interpolation

Remember that linear interpolation uses a linear function for each of intervals  $[x_k, x_{k+1}]$ . Spline interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a spline.

For instance, the natural cubic spline is piecewise cubic and twice continuously differentiable. Furthermore, its second derivative is zero at the end points. The natural cubic spline interpolating the points in the table above is given by



Plot of the data with spline interpolation applied

$$f(x) = \begin{cases} -0.1522x^3 + 0.9937x, & \text{if } x \in [0, 1], \\ -0.01258x^3 - 0.4189x^2 + 1.4126x - 0.1396, & \text{if } x \in [1, 2], \\ 0.1403x^3 - 1.3359x^2 + 3.2467x - 1.3623, & \text{if } x \in [2, 3], \\ 0.1579x^3 - 1.4945x^2 + 3.7225x - 1.8381, & \text{if } x \in [3, 4], \\ 0.05375x^3 - 0.2450x^2 - 1.2756x + 4.8259, & \text{if } x \in [4, 5], \\ -0.1871x^3 + 3.3673x^2 - 19.3370x + 34.9282, & \text{if } x \in [5, 6]. \end{cases}$$

In this case we get  $f(2.5) = 0.5972$ .

Like polynomial interpolation, spline interpolation incurs a smaller error than linear interpolation and the interpolant is smoother. However, the interpolant is easier to evaluate than the high-degree polynomials used in polynomial interpolation. It also does not suffer from Runge's phenomenon.

## Interpolation via Gaussian processes

Gaussian process is a powerful non-linear interpolation tool. Many popular interpolation tools are actually equivalent to particular Gaussian processes. Gaussian processes can be used not only for fitting an interpolant that passes exactly through the given data points but also for regression, i.e., for fitting a curve through noisy data. In the geostatistics community Gaussian process regression is also known as Kriging.

## Other forms of interpolation

Other forms of interpolation can be constructed by picking a different class of interpolants. For instance, rational interpolation is **interpolation** by rational functions using Padé approximant, and trigonometric interpolation is interpolation by trigonometric polynomials using Fourier series. Another possibility is to use wavelets.

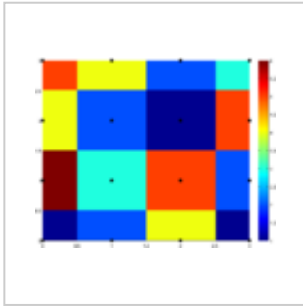
The Whittaker–Shannon interpolation formula can be used if the number of data points is infinite.

Sometimes, we know not only the value of the function that we want to interpolate, at some points, but also its derivative. This leads to Hermite interpolation problems.

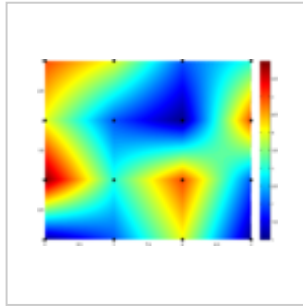
When each data point is itself a function, it can be useful to see the interpolation problem as a partial advection problem between each data point. This idea leads to the displacement interpolation problem used in transportation theory.

## In higher dimensions

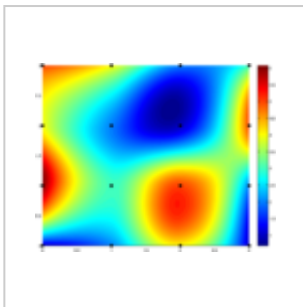
Multivariate interpolation is the interpolation of functions of more than one variable. Methods include bilinear interpolation and bicubic interpolation in two dimensions, and trilinear interpolation in three dimensions. They can be applied to gridded or scattered data.



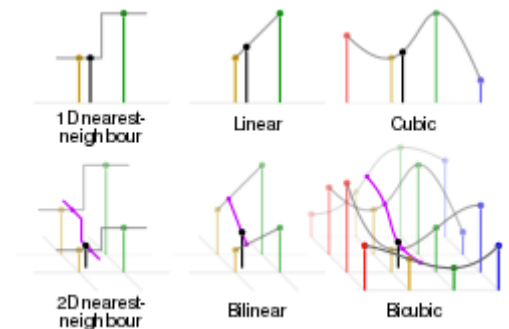
Nearest neighbor



Bilinear



Bicubic



Comparison of some 1- and 2-dimensional interpolations. Black and red/yellow/green/blue dots correspond to the interpolated point and neighbouring samples, respectively. Their heights above the ground correspond to their values.

## Interpolation in digital signal processing

In the domain of digital signal processing, the term interpolation refers to the process of converting a sampled digital signal (such as a sampled audio signal) to that of a higher sampling rate (Upsampling) using various digital filtering techniques (e.g., convolution with a frequency-limited impulse signal). In this application there is a specific requirement that the harmonic content of the original signal be preserved without creating aliased harmonic content of the original signal above the original Nyquist limit of the signal (i.e., above  $f_s/2$  of the original signal sample rate). An early and fairly elementary discussion on this subject can be found in Rabiner and Crochiere's book *Multirate Digital Signal Processing*.<sup>[1]</sup>

## Related concepts

The term *extrapolation* is used to find data points outside the range of known data points.

In curve fitting problems, the constraint that the interpolant has to go exactly through the data points is relaxed. It is only required to approach the data points as closely as possible (within some other constraints). This requires parameterizing the potential interpolants and having some way of measuring the error. In the simplest case this leads to least squares approximation.

Approximation theory studies how to find the best approximation to a given function by another function from some predetermined class, and how good this approximation is. This clearly yields a bound on how well the interpolant can approximate the unknown function.

## See also

- Barycentric coordinates – for interpolating within on a triangle or tetrahedron
- Bilinear interpolation
- Brahmagupta's interpolation formula
- Extrapolation
- Imputation (statistics)
- Lagrange interpolation
- Missing data
- Multivariate interpolation
- Newton–Cotes formulas
- Polynomial interpolation
- Simple rational approximation

## References

1. R.E. Crochiere and L.R. Rabiner. (1983). Multirate Digital Signal Processing. Englewood Cliffs, NJ: Prentice–Hall. (<https://www.amazon.com/Multirate-Digital-Signal-Processing-Crochiere/dp/0136051626>)

## External links

- Online tools for linear (<http://tools.timodenk.com/linear-interpolation>), quadratic (<http://tools.timodenk.com/quadratic-interpolation>), cubic spline (<http://tools.timodenk.com/cubic-spline-interpolation>), and polynomial (<http://tools.timodenk.com/polynomial-interpolation>) interpolation with visualisation and JavaScript source code.
- Sol Tutorials - Interpolation Tricks (<http://sol.gfxile.net/interpolation/index.html>)

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Interpolation&oldid=798320122>"

- 
- This page was last edited on 1 September 2017, at 06:26.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.