Numerical Calculus

**2D Interpolation**

--Trufanda Bogdan—

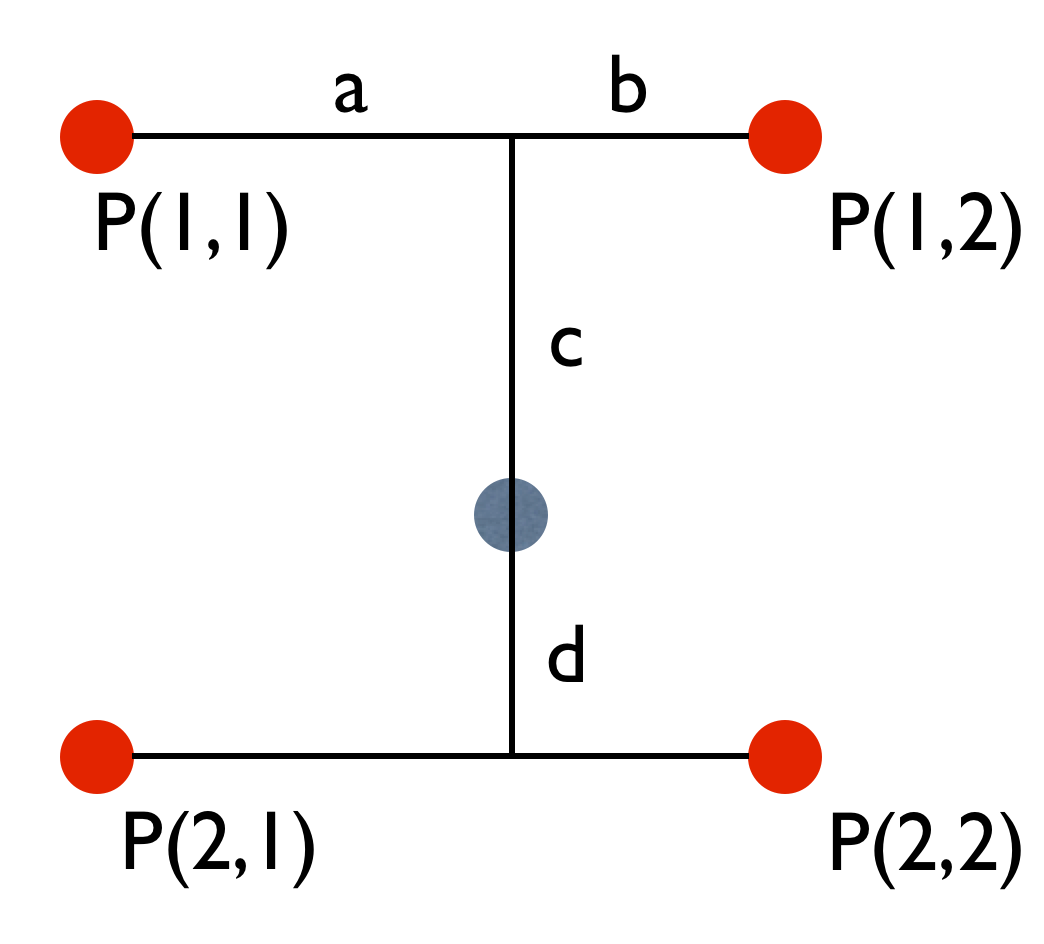
--III E1--

1. **Definition:**

Interpolation is the process of estimating a value within two known points in a pool of data. When this contains a gap, but data is available on either side of the gap or at a few specific points within the gap, interpolation allows us to estimate the unknown values.

Given a set of discrete data points representing the known values v(xi) at certain positions xi,  ( i = 1, 2 …. n) in an n-D space, we can estimate the missing value typically by fitting the data set by a continuous function f(x), such as a polynomial, that either passes through the data points f(xi) = v(xi), or approximates the data points f(Xi) v(Xi). The function f(x) should be continuous (continuous) and preferably smooth (continuous with k > 0).

2D interpolation is an extension of linear interpolation, with v(Xi) = f(Xi, Yi), for every i = (1,2 …. n). It is performed using linear interpolation first in one direction, and then again in the other direction. Although each step is linear in the sampled values and in the position, the interpolation as a whole is not linear but rather quadratic.

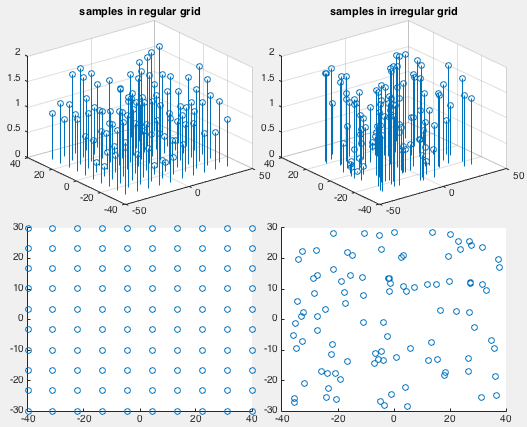


1. **Representation:**

The four red dots show the data points and the grey dot is the point at which we want to interpolate.

1. **Algorithm:**

2D interpolation can be applied, as we can see in the picture below, both on a regular grid or an irregular grid (scattered data points). Depending on the data grid, we will apply a different algorithm. We need to keep in mind that an algorithm that can be applied to a regular grid cannot be applied on an irregular grid, but the reverse can.



**3.1. Methods based on sample points in a regular grid**

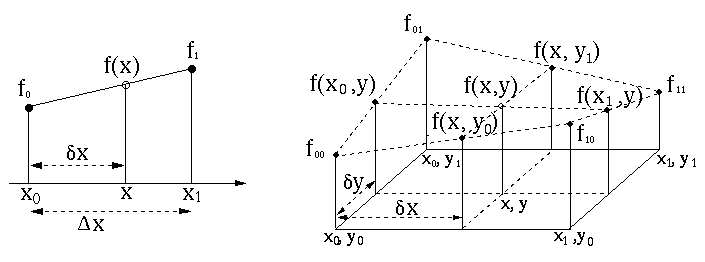
**Bilinear Interpolation**

Given a set of 2D sample points in a regular grid, we can use the methods of bilinear/bicubic 2D interpolation to obtain the value of the interpolating function f(x, y) at any point (x, y) inside each of the rectangles in a 2D grid with the four corners at (xi ,yj), (xi+1 ,yj+1), (xi ,yj+1) and (xi+1 ,yj).

In the following, for convenience and without loss of generality, we only consider i = j = 0 and define.

F(x), in every point x between x0 < x < x1, can be approximated by linear interpolation based on f0 = f(x0) and f1 = f(x1):

, i.e. f(x) =



­­­This method of 1-D linear interpolation can be extended to the bilinear interpolation method to calculate f(x, y) at any point with x0 < x < x1 and y0 < x < y1, based on the known values f00 = f (x0, y0), f01 = f (x0, y1), f10 = f (x1, y0), f11 = f (x1, y1).

**Linear interpolation in x-dimension:**

f(x, ) =

f(x, ) =

**Linear interpolation in y-dimension:**

f(x, y) =

=

=

The order in which we do this linear interpolation doesn`t matter; if interpolation is first carried out in y-direction and then in x-direction, or vice versa the result is exactly the same.

**3.2. Methods based on sample points in irregular grid**

**Radial Basic Function Method**

A radial basis function (RBF) is a function that is symmetric with respect to a specific point x0 = [x0, y1]T  i.e., the value of the RBF at any 2D point can be simply represented by ɸ0 (x) = ɸ(|| x – x0 ||), as a function of the distance between the point and x0 ( d(x,x0) = || x – x0 || = ).

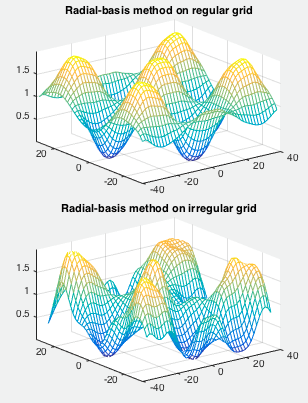
Typical RBFs include the Gaussian and Butterworth functions:

ɸc (x) = exp(-), ɸc (x) = , where σ is a parameter that controls the width of the RBF, and k controls the shape of the function. Based on a given ɸc (x), we can construct an interpolating function f(x, y) = f(x) as the weighted sum of such RBFs each centered around one of the given sample points: v(, (I = 1,2,…n):

The weights are determined based on the requirement that the interpolating function takes the same value as the sample point at each of the n positions:

, (j = 1, 2, …. n)

Solving this equation system of n linear equations with coefficients , we get the weights and thereby the interpolation function f(x).

****

Here we can see the difference on applying RBF on regular (top) and irregular grid (bottom).

## Application

In computer vision and image processing, interpolation is used to resample images and textures. It is used to map a screen pixel location to a corresponding point on the map. A weighted average of the attributes (color, transparency, etc.) of the four surrounding texture element is computed and applied to the screen pixel. This process is repeated for each pixel forming the object being textured.

When an image needs to be scaled up, each pixel of the original image needs to be moved in a certain direction based on the scale constant. However, when scaling up an image by a non-integral scale factor, there are pixels that are not assigned appropriate pixel values. In this case, those pixels should be assigned appropriate RGB or grayscale values so that the output image does not have non-valued pixels.

Bilinear interpolation can be used where perfect image transformation with pixel matching is impossible, so that we can calculate and assign appropriate intensity values to pixels. Unlike other interpolation techniques such as nearest-neighbor interpolation, bilinear interpolation uses values of only the 4 nearest pixels, located in diagonal directions from a given pixel, in order to find the appropriate color intensity values of that pixel.

Bibliography:

* https://whatis.techtarget.com/definition/extrapolation-and-interpolation
* http://fourier.eng.hmc.edu/e176/lectures/ch7/node1.html
* https://www.mathworks.com/matlabcentral/fileexchange/43533-bilinear-interpolation-of-an-image-or-matrix
* https://en.wikipedia.org/wiki/Bilinear\_interpolation
* http://fourier.eng.hmc.edu/e176/lectures/ch7/node7.html