The Universe on your computer

Workshop conducted for Sampark, Shaastra

Pranav Satheesh   Shreenath Guard

Indian Institute of Technology Madras

Outline

Interpolation

Why Interpolation?

We are often met with situation where we know the values of a function at discrete points, but we would like to evaluate the function at general points x.

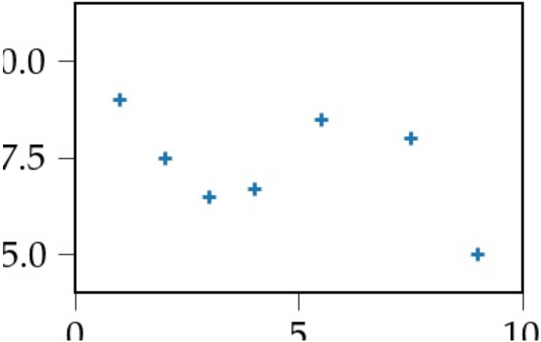

Figure: Discrete Points

To solve this, we need to find a polynomial $p(x)$ that is an approximation and *interpolates* $f(x)$ between the x_i with $p(x_i) = f(x_i)$

In general, our interpolation polynomial of degree n that passes through "n+1" points can be written as:

$$p(x) = a_o + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

The a_i are $n+1$ real constants can be determined by solving a set of $n+1$ linear equations:

$$\begin{pmatrix} 1 & x_o^1 & x_o^2 & \cdots & x_o^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n^1 & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_o) \\ \vdots \\ f(x_n) \end{pmatrix}$$

The matrix of polynomial variable is called the *Vandermonde Matrix* For large n this gets very complicated and we look at two simplest cases, linear (n=1) and quadratic (n=2) interpolation.

Linear Interpolation

We can obtain the linear approximation $p(x)$ for $f(x)$ in the interval $[x_i, x_{i+1}]$ by

$$p(x) = f(x_i) + \underbrace{\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i)}_{\text{1-st order forward difference}} + O(h^2)$$

Where, h $= x_{i+1} - x_i$. Linear interpolation is the simplest method for interpolation.

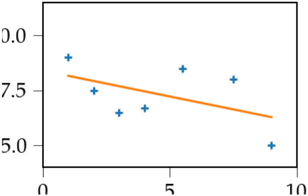

Figure: Linear interpolation done on the points

The $p(x)$ for $f(x)$ in the interval $[x_i, x_{i+1}]$ by

$$p(x) = \frac{(x - x_{i+1})(x - x_{i+2})}{(x_i - x_{i+1})(x_i - x_{i+2})} f(x_i) + \frac{(x - x_i)(x - x_{i+2})}{(x_{i+1} - x_i)(x_{i+1} - x_{i+2})} f(x_{i+1}) + \frac{(x - x_i)(x - x_{i+1})}{(x_{i+2} - x_i)(x_{i+2} - x_{i+1})} f(x_{i+2}) + O(h^3)$$

Where, h $= max\,[x_{i+2} - x_{i+1}, x_{i+1} - x_i]$.
The results depend on which three points are chosen, as there are two choice : x_i, x_{i+1}, x_{i+2} or x_{i-1}, x_i, x_{i+1}

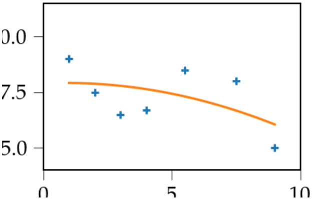

Figure: Quadratic Interpolation

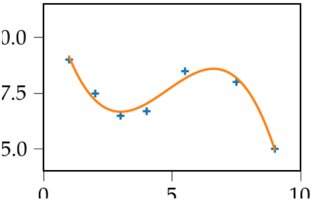

Figure: Cubic Interpolation

Finding Roots

Newton Raphson Method

Newton Raphson method is a root finding algorithm used to find roots of polynomials. You start with a guess value x_n and use a taylor series approximation:

$$y = f^{'}(x_n)(x - x_n) + f(x_n)$$

where $f^{'}$ is the tangent to the curve.
The x-intercept of this line (the value of x which makes y = 0) is taken as the next approximation, x_{n+1}, to the root, so that the equation of the tangent line is satisfied when $(x, y) = (x_{n+1}, 0)(x, y) = (x_{n+1}, 0)$

$$0 = f^{'}(x_n)(x_{n+1} - x_n) + f(x_n)$$

Solving for x_n gives,

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

Fourier Transform

A function f(x) can be expressed as a series of sines and cosines:

$$f(x) = \frac{1}{2}a_o + \sum_{n=1}^{\infty} a_n cos(nx) + \sum_{n=1}^{\infty} a_n sin(nx)$$

where:

$$a_o = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)cos(nx)dx$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)sin(nx)dx$$

Generalised Fourier Transforms

The Fourier series can be generalised to the whole complex plane:
Forward Fourier Transform:

$$F(k) = \int_{-\infty}^{\infty} f(x)e^{-2\pi ikx}dx$$

Inverse Fourier Transform:

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(k)e^{2\pi ikx}dk$$

Since we can't have continuous values, we need to deal with discrete values and we have just the right algorithm for it.

Forward DFT:

$$F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i n \frac{k}{N}}$$

Inverse DFT:

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi i n \frac{k}{N}}$$