Data interpolation

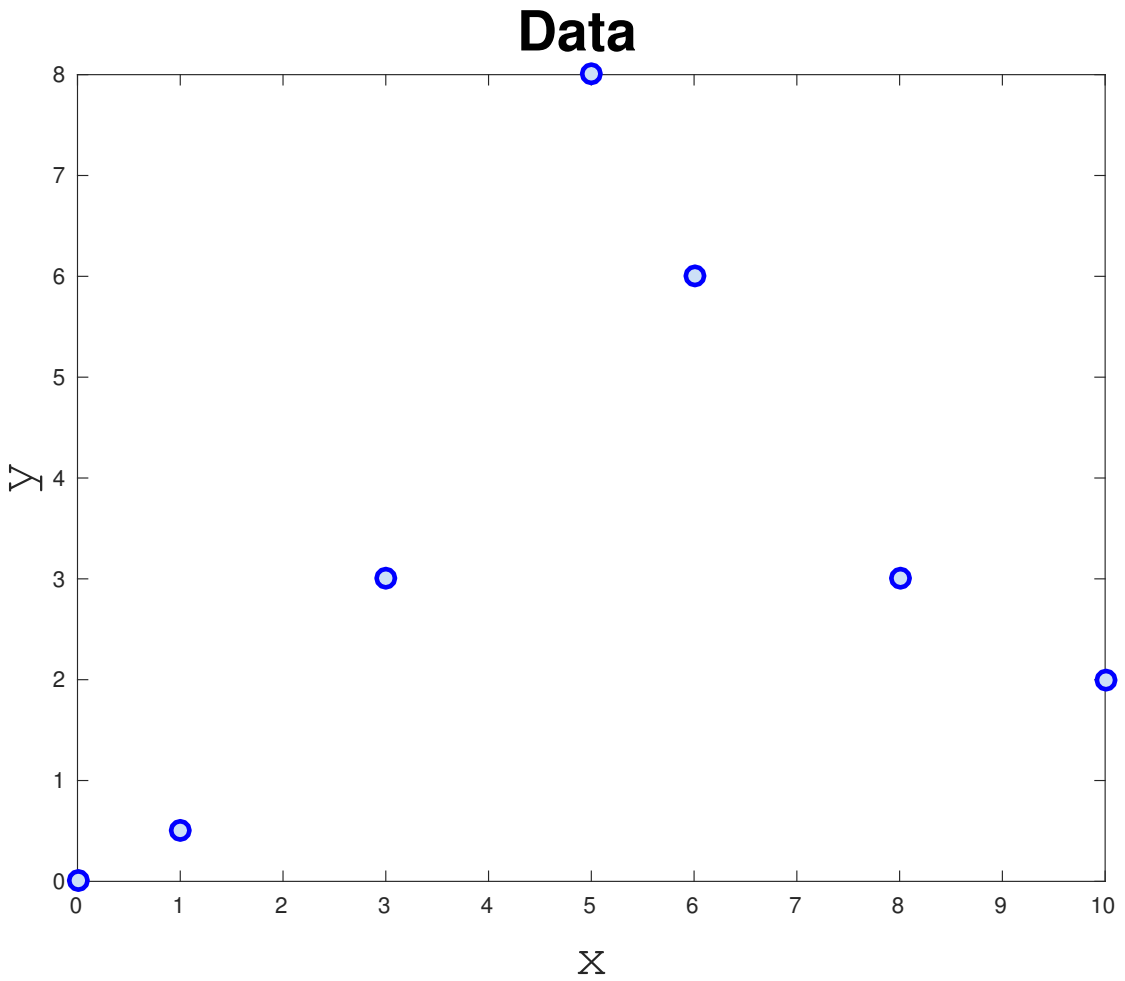
Eugeniy E. Mikhailov

The College of William & Mary

Lecture 22

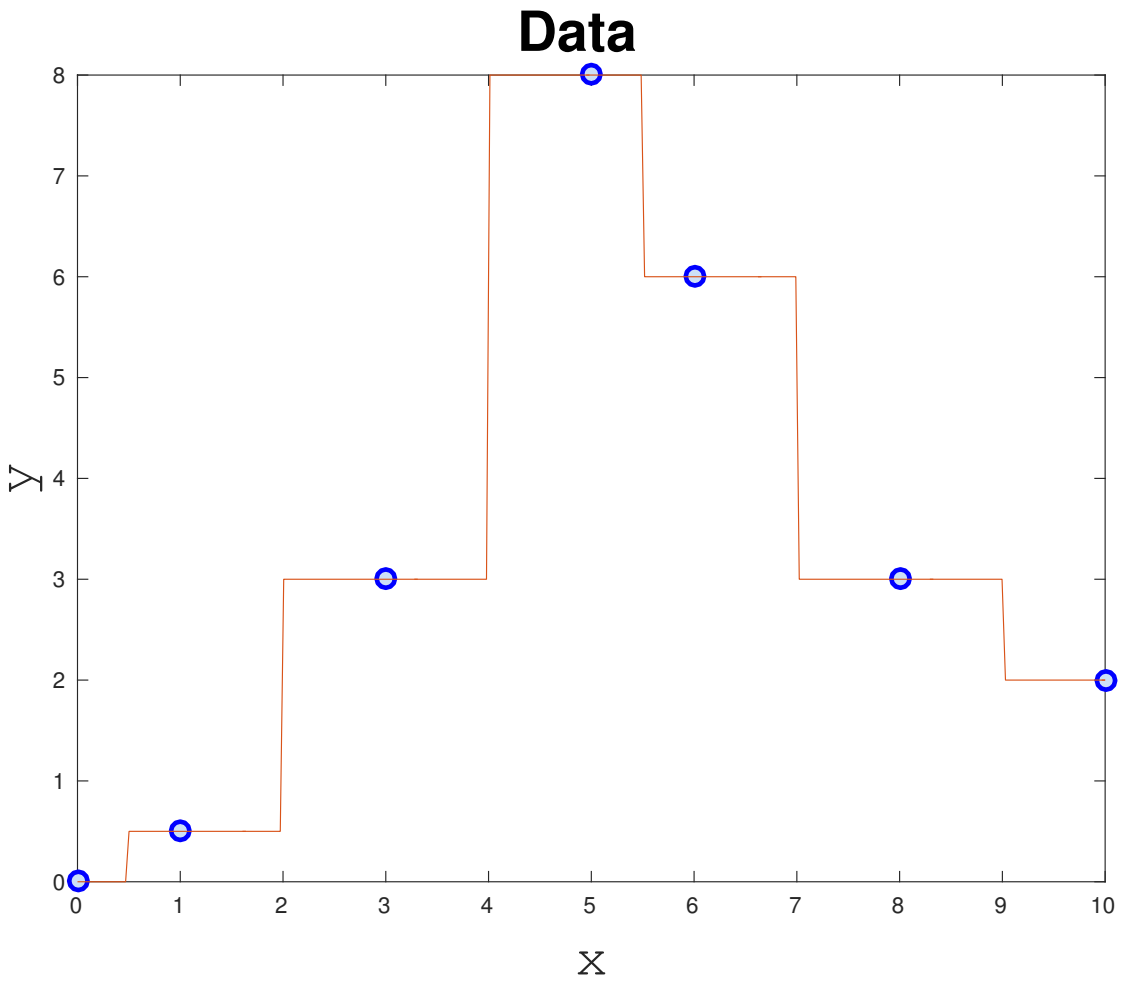
Data interpolation - filling the voids

There is rarely enough data. It often takes a lot of time to get a data point. It might be expensive. Nevertheless, we would like to have some representation of the system in the voids.



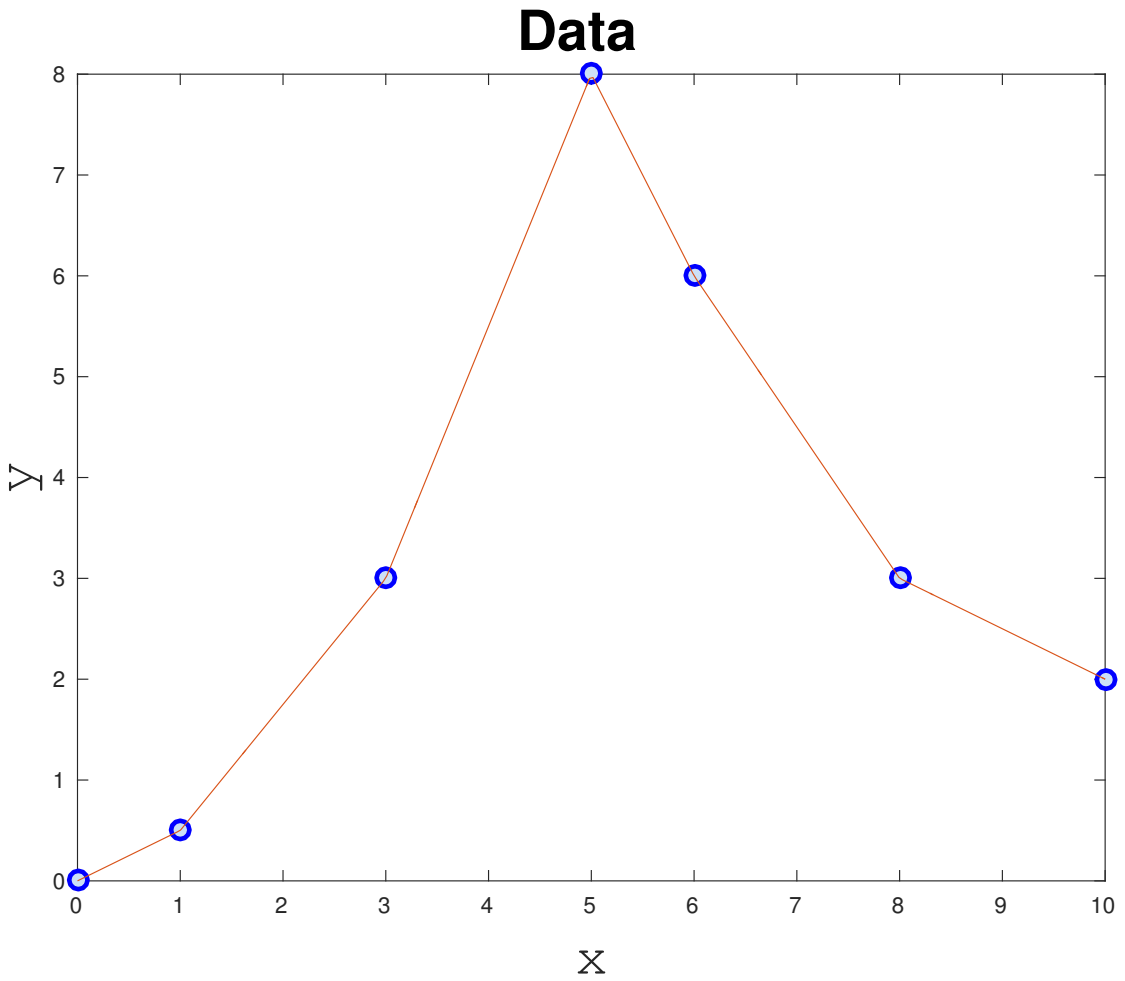
The nearest neighbor interpolation

The name says it all. For each interpolated point ($x_{interpolated}$), find the nearest neighbor along the x_i axis in the data set and use its y_i value.



Linear interpolation

We will split our data set with N points to $N-1$ intervals and interpolate the values in the given interval as a line passing through the border points (x_i, y_i) and (x_{i+1}, y_{i+1})



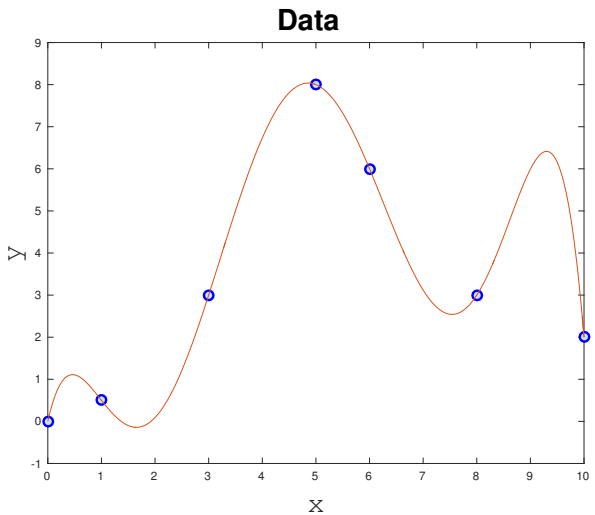
Polynomial fit

You can always find a polynomial of $N-1$ degree passing through N data points.

$$P_N(x) = p_1 x^N + p_2 x^{N-1} + \cdots + p_N x + p_{N+1}$$

Matlab has the 'polyfit' function which returns the polynomial coefficient.

```
% calculate coefficients
p=polyfit(xdata, ydata, (length(xdata)-1) );
%  interpolate
yi=polyval(p,xi);
```

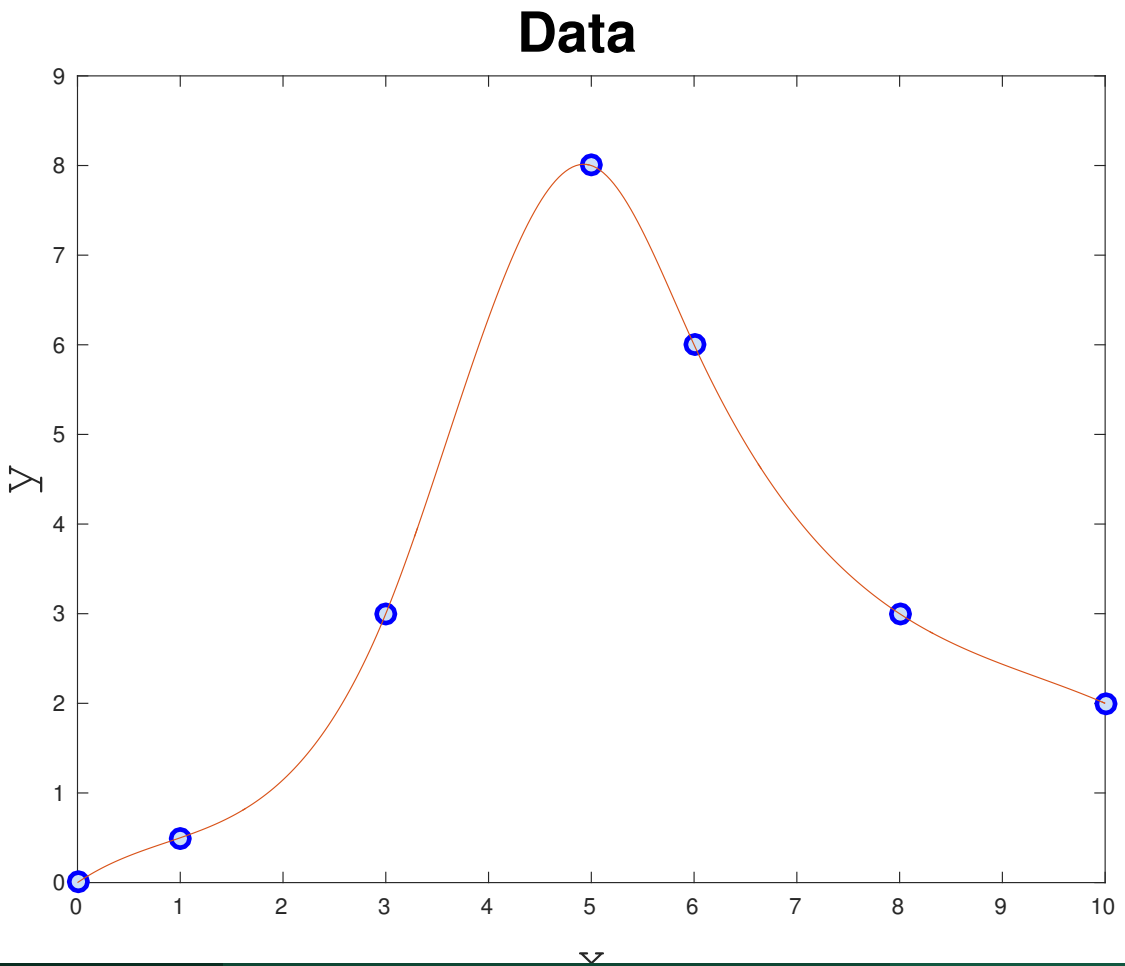


Interpolated values tend to oscillate for a polynomial of a high degree.
Do not use it unless you know what you are doing!

Cubic spline interpolation

We will interpolate N data points by a polynomial of 3rd degree for each i_{th} interval between data point

$$f_i(x) = p_{1_i} x^3 + p_{2_i} x^2 + p_{3_i} x + p_{4_i}, x \in [x_i, x_{i+1}]$$



Cubic spline interpolation demystified

We will interpolate N data points by a polynomial of 3rd degree for each i_{th} interval between data point

$$f_i(x) = p_{1_i} x^3 + p_{2_i} x^2 + p_{3_i} x + p_{4_i}, x \in [x_i, x_{i+1}]$$

Interpolation must pass through data points

$$\begin{aligned} f_i(x_i) &= y_i \\ f_i(x_{i+1}) &= y_{i+1} \end{aligned}$$

The two above equations are not sufficient to constrain the four polynomial coefficients.

We request $f_i(x)$ to have continuous 1st derivative at the borders

$$\begin{aligned} f_i'(x_{i+1}) &= f_{i+1}'(x_{i+1}) \\ f_i'(x_i) &= f_{i-1}''(x_i) \end{aligned}$$

Additionally, we specify the 2nd derivatives at end points. Common choice is to set it to 0. This is the, so-called, natural cubic spline.

$$\begin{aligned} f_1''(x_1) &= 0 \\ f_{N-1}''(x_N) &= 0 \end{aligned}$$

Matlab built in interpolation

Use matlab `interp1(xdata, ydata, xi, method)` for some of above methods
Where `method` could be

'nearest'	Nearest neighbor interpolation
'linear'	Linear interpolation (default)
'spline'	Cubic spline interpolation
other	see more in help

Extrapolation - voids outside the measurements region

Do not extrapolate unless you have a physical model of the process!