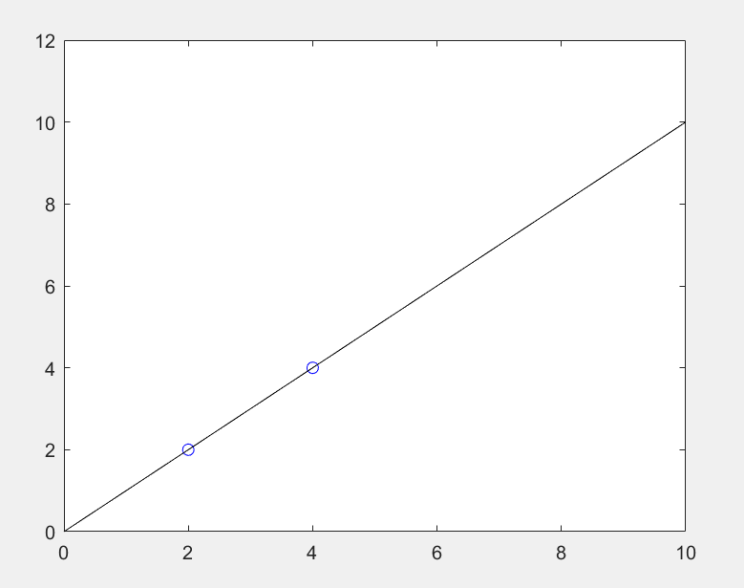
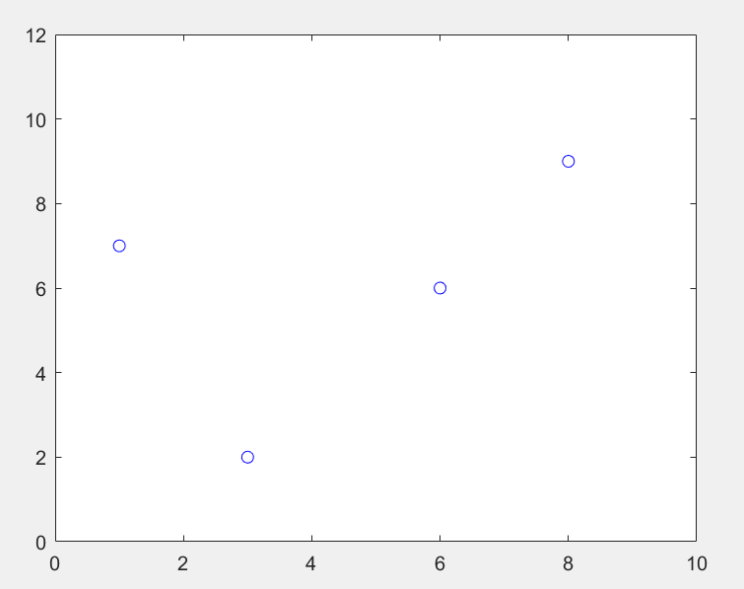
Brief Overview of Cubic Splines:

- Splines are a method used to interpolate between n data points.
    - What is interpolation? Interpolation is an estimation method that connects n data points using a polynomial optimized to the lowest possible degree
    - The simplest example is a linear interpolation:



The two points are connected using a linear equation: $y = x$

    - So what if we have a more complex dataset?

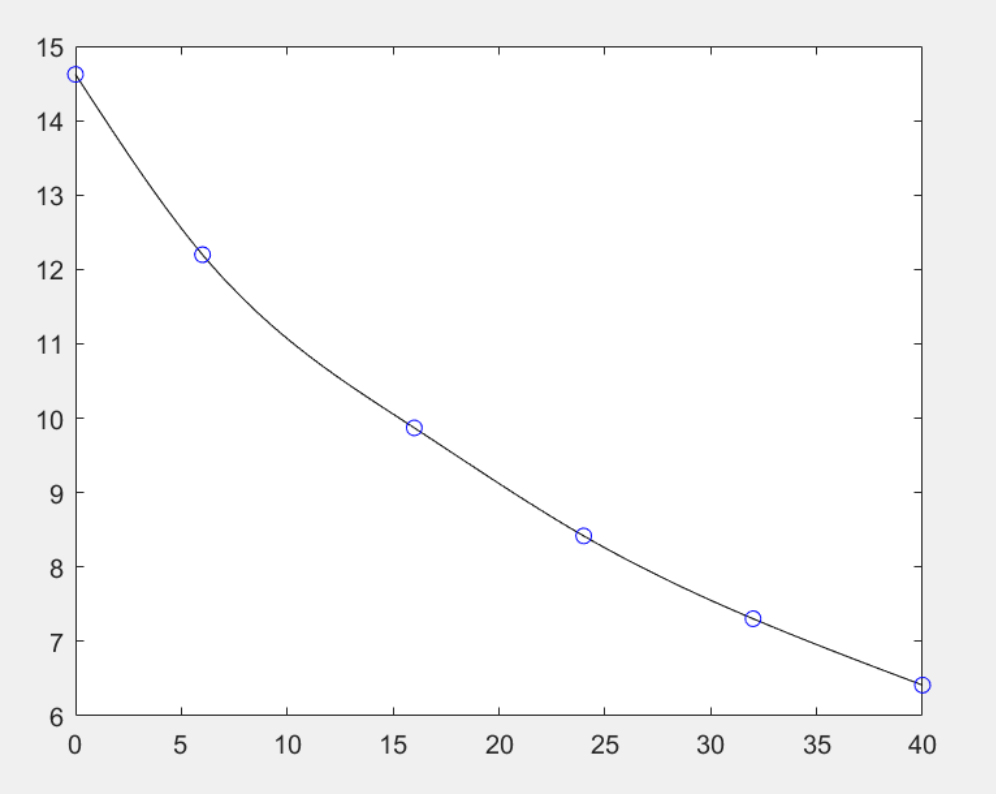


One approach to interpolate this dataset is using cubic splines

- Splines are an application of using lower-order polynomials in a piecewise fashion to subsets of data points
- Cubic splines are generally used in place other splines, such as quadratic splines and splines of higher order, because they provide the simplest representation that exhibits the smooth appearance expected of an equation

Below is an example graph applying a cubic spline to the dataset:

```
T = [0 6 16 24 32 40];
o = [14.621 12.198 9.870 8.418 7.305 6.413];
```



Between each data point is an individual polynomial. In total, 5 equations are used to piecewise the overall dataset

MATLAB Code:

```
n = length(T);

h = T(2:n) - T(1:n - 1);
Df = (o(2:n) - o(1:n - 1))./h;

C = eye(n); r = zeros(n, 1);
for i = 2:n - 1
     C(i, i - 1:i + 1) = [h(i - 1) 2*(h(i - 1) + h(i)) h(i)];
     r(i) = 3*(Df(i) - Df(i - 1));
end
c = C\r;

a = o;
for j = 1:n - 1
     d(j) = (c(j + 1) - c(j))/(3*h(j));
     b(j) = ((o(j + 1) - o(j))/h(j)) - (h(j)/3)*(2*c(j) + c(j +
     1));
end

xi1 = linspace(T(1), T(2), 1000);
spline1 = @(xi1) a(1) + b(1)*(xi1 - T(1)) + c(1)*(xi1 - T(1)).^2
+ d(1)*(xi1 - T(1)).^3;

xi2 = linspace(T(2), T(3), 1000);
spline2 = @(xi2) a(2) + b(2)*(xi2 - T(2)) + c(2)*(xi2 - T(2)).^2
+ d(2)*(xi2 - T(2)).^3;

xi3 = linspace(T(3), T(4), 1000);
spline3 = @(xi3) a(3) + b(3)*(xi3 - T(3)) + c(3)*(xi3 - T(3)).^2
+ d(3)*(xi3 - T(3)).^3;

xi4 = linspace(T(4), T(5), 1000);
spline4 = @(xi4) a(4) + b(4)*(xi4 - T(4)) + c(4)*(xi4 - T(4)).^2
+ d(4)*(xi4 - T(4)).^3;

xi5 = linspace(T(5), T(6), 1000);
spline5 = @(xi5) a(5) + b(5)*(xi5 - T(5)) + c(5)*(xi5 - T(5)).^2
+ d(5)*(xi5 - T(5)).^3;

plot(xi1, spline1(xi1), 'k-');
hold on
plot(xi2, spline2(xi2), 'k-');
plot(xi3, spline3(xi3), 'k-');
plot(xi4, spline4(xi4), 'k-');
plot(xi5, spline5(xi5), 'k-');
plot(T, o, 'bo')
hold off
```

This MATLAB code is based on Problem 18.9 from the Fourth Edition Applied Numerical Methods with MATLAB® textbook.

Sources:

Chapra, S. C. (2017). Chapter 18. In *Applied numerical methods with Matlab for engineers and scientists* (4th ed., pp. 453–483). McGraw-Hill.