Recitation – 4: Polynomial Interpolation

```
In this recitation, an application of polynomial interpolation is shown. In the code below,
several ways of polynomial interpolation in matlab are given. Please try it yourself with
different input values and check the values of the variables after execution.
For example; See V, after V=vander(x); statement executed.

A common choice of model functions for interpolation is polynomials, which are easy to evaluate
and smooth, i.e., infinitely differentiable.
```

Polynomial interpolation with monomial basis	$\Phi_j(x) = x^j$
For example; for the data points	$(xi, yi) = (2, 14), (6, 24), (4, 25), (7, 15)$
Requires four basis functions	$\Phi(x) = 1, x, x^2, x^3.$
The interpolant will be	$p(x) = c_0 1 + c_1 x + c_2 x^2 + c_3 x^3.$
Construct linear system and solve	c = V \ y

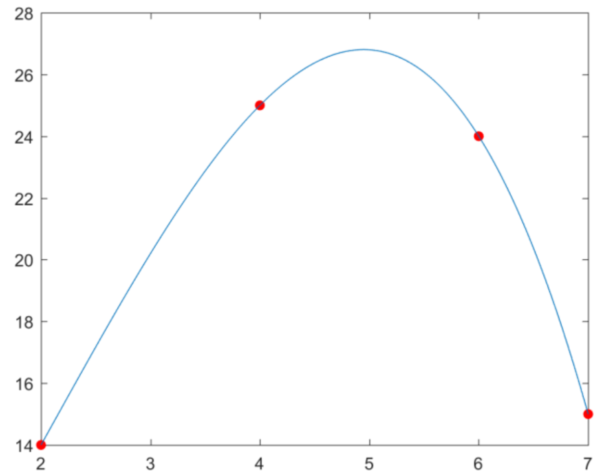
$$\underbrace{\begin{bmatrix} 1 & x_0 & \dots & x_0^{n-1} & x_0^n \\ 1 & x_1 & \dots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & \dots & x_n^{n-1} & x_n^n \end{bmatrix}}_{V} \underbrace{\begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix}}_{\mathbf{c}} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\mathbf{y}}$$

Construct a linear system to find the polynomial function that fit to data points and solve Example 1

```
% Data points
x = [2 6 4 7];
y = [14 24 25 15];

V=vander(x); % V is called a Vandermonde matrix. Its elements are the powers of the nodes.
c=V\y'; % Solve system of linear equations V c = y for c ==> V^-1 * y;

% Plot
xx = [2:.02:7]; % define an interval of points containing x points to give to the fitted function
yy = polyval(c,xx);
figure(); % Plot for (xx, yy) points in the interval along with the (x, y) points
plot(x,y,'r.','MarkerSize',20); hold on;
plot(xx, yy); hold off;
```

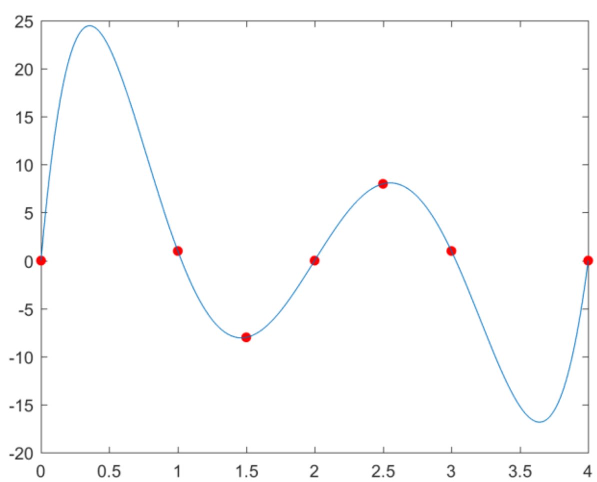


Example 2

```
% A different set of points
x2 = [0 1 1.5 2 2.5 3 4];
y2 = [0 1 -8 0 8 1 0];

V2 = vander(x2);
c2 = V2\y2';

xx2 = [0:.02:4];
yy2 = polyval(c2,xx2);
figure();
plot(x2,y2,'r.','MarkerSize',20); hold on;
plot(xx2, yy2); hold off;
```



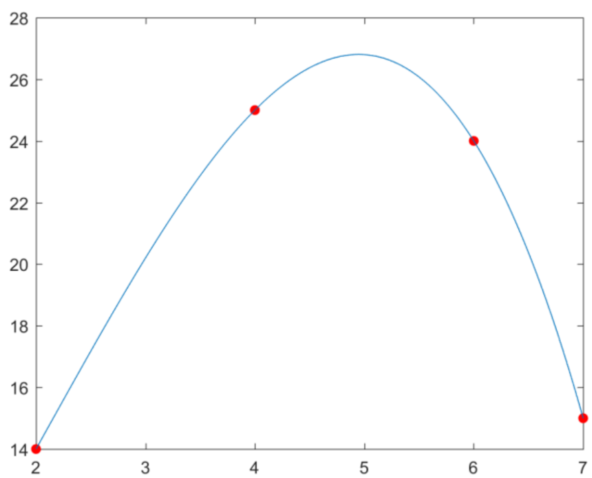
You can also calculate vandermonde matrix as below.

```
% Data points
x = [2 6 4 7];
y = [14 24 25 15];

% Vandermonde matrix
A = zeros(length(x));
for i=1:length(x)
    A(:,i)= x.^(length(x)-i);
end

% Solve the linear system
c = A^-1 * y'; % A * c = y -> A^-1 * A * c = A^-1 * y -> c = A^-1 * y

xx = [2:.02:7]; % define an interval of points containing x points to give to the fitted function
yy = polyval(c,xx);
figure();
plot(x,y,'r.','MarkerSize',20); hold on;
plot(xx, yy); hold off;
```



You can use polyfit function to fit a polynomial of degree n for the given data points

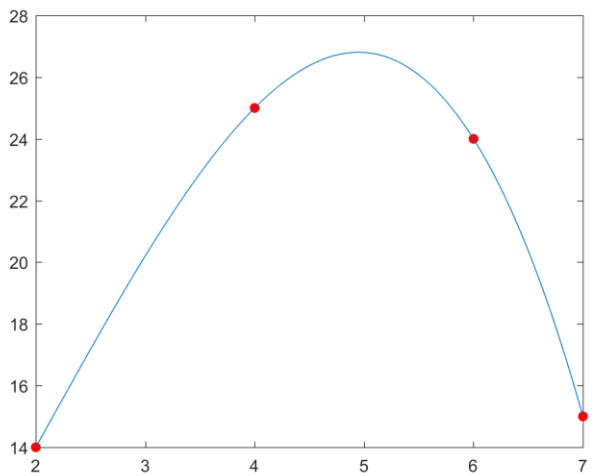
polyfit function solve the linear system and returns the coefficient vector

```
% Data points
x = [2 6 4 7];
y = [14 24 25 15];

c = polyfit(x,y,3);

xx = [2:.02:7]; % define an interval of points containing x points to give to the fitted function
yy = polyval(c,xx);
figure();
```

```
plot(x,y,'r.','MarkerSize',20); hold on;
plot(xx, yy); hold off;
```



Arş. Gör. İsmail Bilgen