BIL 108E
Introduction to Scientific and Engineering Computing

ASST. PROF. DR. İPEK AKIN

POLYNOMIALS

INTRODUCTION

Polynomials are mathematical expressions that are frequently used for problem solving and modeling in science and engineering.

Polynomials

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

- a_n, a_{n-1}, ….., a_1, a_0 → real numbers

- n → degree or order of polynomial, nonnegative integer

Degree of Polynomials

$f(x) = 5x^5 + 6x^2 + 7x + 3$ polynomial of degree 5.

$f(x) = 2x^2 - 4x + 10$ polynomial of degree 2.

$f(x) = 11x - 5$ polynomial of degree 1.

A constant (e.g., $f(x) = 6$) is a polynomial of degree 0.

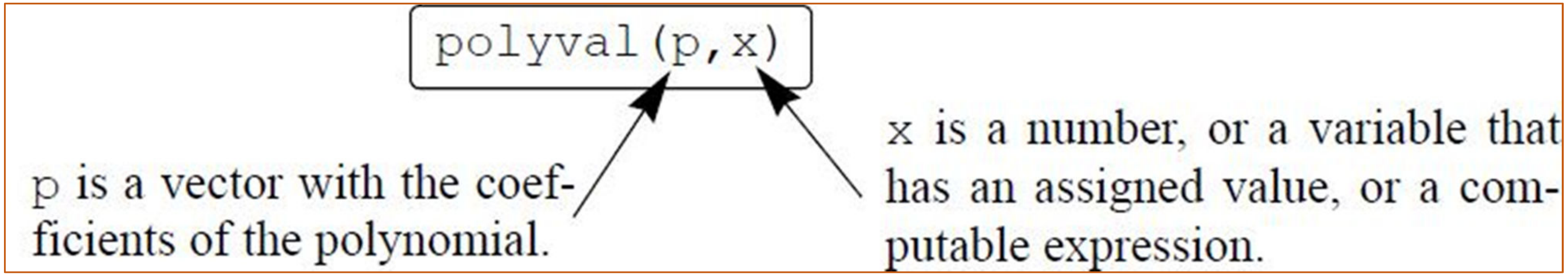
In MATLAB, polynomials are represented by a row vector in which the elements are the coefficients a_n, a_{n-1}, ….., a_1, a_0

The first element is the coefficient of the *x* with the highest power. The vector has to include all the coefficients, including the ones that are equal to 0.

Polynomial	**MATLAB representation**
$8x + 5$	p = [8 5]
$2x^2 - 4x + 10$	d = [2 –4 10]
$6x^2 - 150$, MATLAB form: $6x^2 + 0x - 150$	h = [6 0 –150]
$5x^5 + 6x^2 - 7x$, MATLAB form: $5x^5 + 0x^4 + 0x^3 + 6x^2 - 7x + 0$	c = [5 0 0 6 –7 0]

Value of a Polynomial

The value of a polynomial at a point *x* can be calculated with the function **polyval** which has the form:



x can also be a vector or a matrix → polynomial is calculated for each element (element-by-element), and the answer is a vector, or a matrix, with the corresponding values of the polynomial.

Example

For the polynomial

$$f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17.015x^2 - 71.95x + 35.88$$

(a) Calculate f(9)

(b) Plot the polynomial for $-1.5 \le x \le 6.7$

Solution

(a) Coefficients of the polynomials are assigned to vector p. The function **polyval** is then used to calculate the value at $x = 9$.

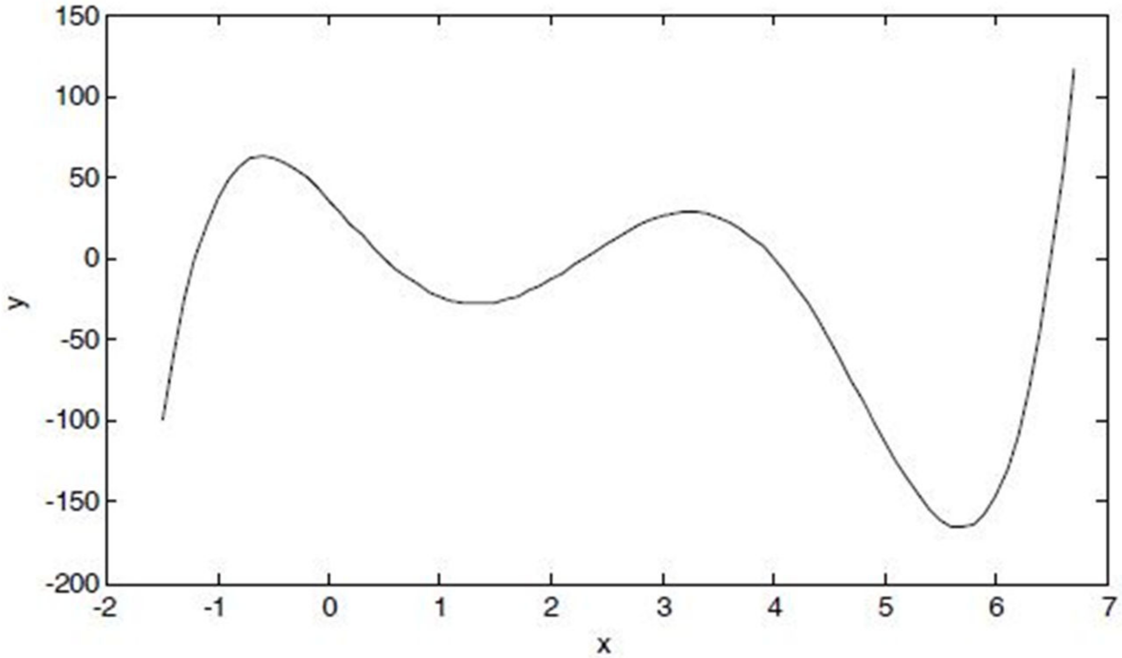
```
>> p = [1 -12.1 40.59 -17.015 -71.95 35.88];
>> polyval(p,9)
ans =
  7.2611e+003
```

(b) To plot the polynomial, a vector x is first defined with elements ranging from –1.5 to 6.7.

- Then a vector y is created with the values of the polynomial for every element of x.
- A plot of y vs. x is made.

```
>> x=-1.5:0.1:6.7;
>> y=polyval(p,x);
>> plot(x,y)
```

Calculating the value of the polynomial for each element of the vector x.



Roots of Polynomial

- The roots of a polynomial are the values of the argument for which the value of the polynomial is equal to zero.

- For example, the roots of the polynomial

$f(x) = x^2 - 2x - 3 = 0$ are the values of *x* for which

$x^2 - 2x - 3 = 0$, which are $x = -1$ and $x = 3$.

Roots of Polynomial

- Command is **roots**
- Determines the root, or roots of a polynomial

$$\boxed{r = roots\ (p)}$$

r is a column vector with the roots of the polynomial

p is a row vector with the coefficients of the polynomial

Example

For the polynomial

$$f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17.015x^2 - 71.95x + 35.88$$

Determine the root.

```
>> p= 1 -12.1 40.59 -17.015 -71.95 35.88];
>> r=roots(p)
r =
    6.5000
    4.0000
    2.3000
   -1.2000
    0.5000
```

When the roots are known, the polynomial can actually be written as:

$$f(x) = (x+1.2)(x-0.5)(x-2.3)(x-4)(x-6.5)$$

The **roots** command is very useful for finding the roots of a quadratic equation.

For example, to find the roots of $f(x) = 4x^2 + 10x - 8$

```
>> roots([4 10 -8])

ans =
   -3.1375
    0.6375
```

When the roots of a polynomial are known, the poly command can be used for determining the coefficients of the polynomial.

```
p = poly(r)
```

p is a row vector with the coefficients of the polynomial.

r is a vector (row or column) with the roots of the polynomial.

Addition, Multiplication, and Division of Polynomials

Addition and Subtraction

- Two polynomials can be added (or subtracted) by adding (subtracting) the vectors of the coefficients.

- If the polynomials are not of the same order (which means that the vectors of the coefficients are not of the same length), the shorter vector has to be modified to be of the same length as the longer vector by **adding zeros** (called padding) in front.

Addition and Subtraction

$$f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$$

$$f_2(x) = 3x^3 - 2x - 6$$

```
>> p1=[3 15 0 -10 -3 15 -40];
>> p2=[3 0 -2 -6];
>> p=p1+[0 0 0 p2]
p =
     3    15     0    -7    -3    13   -46
```

Three 0s are added in front of p2, since the order of p1 is 6 and the order of p2 is 3.

Multiplication **conv**

```
c = conv(a,b)
```

c is a vector of the coefficients of the polynomial that is the product of the multiplication.

a and b are the vectors of the coefficients of the polynomials that are being multiplied.

- The two polynomials do not have to be of the same order.
- Multiplication of three or more polynomials is done by using the **conv** function repeatedly.

Multiplication

$$f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$$

$$f_2(x) = 3x^3 - 2x - 6$$

```
>> p1=[3 15 0 -10 -3 15 -40];
>> p2=[3 0 -2 -6];
```

```
>> pm=conv(p1,p2)
pm =
     9    45    -6   -78   -99    65   -54   -12   -10   240
```

The answer is

$$9x^9 + 45x^8 - 6x^7 - 78x^6 - 99x^5 + 65x^4 - 54x^3 - 12x^2 - 10x + 240$$

Division **deconv**

```
[q,r] = deconv(u,v)
```

q is a vector with the coefficients of the quotient polynomial.
r is a vector with the coefficients of the remainder polynomial.

u is a vector with the coefficients of the numerator polynomial.
v is a vector with the coefficients of the denominator polynomial.

Dividing $\mathbf{2x^3 + 9x^2 + 7x - 6}$ by $\mathbf{x + 3}$

Example

Divide $\mathbf{2x^3 + 9x^2 + 7x - 6}$ by $\mathbf{x + 3}$

Solution

```
>> u = [2  9  7  -6];
>> v = [1  3];
```

```
>> [a b]=deconv(u,v)

a =
     2     3    -2

b =
     0     0     0     0
```

The answer is: $2x^2 + 3x - 2$.

Remainder is zero.

Example (with a remainder)

Divide $\mathbf{2x^6 - 13x^5 + 75x^3 + 2x^2 - 60}$ by $\mathbf{x^2 - 5}$

Solution

```
>> w=[2 -13 0 75 2 0 -60];
>> z=[1 0 -5];
>> [g h]=deconv(w,z)
g =
     2   -13    10    10    52
h =
     0     0     0     0     0    50    200
```

The quotient is: $2x^4 - 13x^3 + 10x^2 + 10x + 52$.

The remainder is: $50x + 200$.

The answer is:

$$2x^4 - 13x^3 + 10x^2 + 10x + 52 + \frac{50x + 200}{x^2 - 5}$$

Derivatives of Polynomials **polyder**

polyder can be used to calculate the derivative of a single polynomial, a product of two polynomials, or a quotient of two polynomials.

Derivatives of Polynomials

`k = polyder(p)`	Derivative of a single polynomial. `p` is a vector with the coefficients of the polynomial. `k` is a vector with the coefficients of the polynomial that is the derivative.
`k = polyder(a,b)`	Derivative of a product of two polynomials. `a` and `b` are vectors with the coefficients of the polynomials that are multiplied. `k` is a vector with the coefficients of the polynomial that is the derivative of the product.
`[n d] = polyder(u,v)`	Derivative of a quotient of two polynomials. `u` and `v` are vectors with the coefficients of the numerator and denominator polynomials. `n` and `d` are vectors with the coefficients of the numerator and denominator polynomials in the quotient that is the derivative.

Number of output arguments are different in 2 and 3.

With two output arguments MATLAB calculates the derivative of the quotient of two polynomials. With one output argument the derivative is of the product.

Example

$$f_1(x) = 3x^2 - 2x + 4$$

$$f_2(x) = x^2 + 5$$

Determine the derivatives of

$$3x^2 - 2x + 4$$

$$(3x^2 - 2x + 4)\,(x^2 + 5)$$

$$\frac{3x^2 - 2x + 4}{x^2 + 5}$$

Solution

```
>> f1 = [3  -2  4];
>> f2 = [1  0  5];
```

```
>> k = polyder (f1)
```

```
k =
        6       -2
```

The derivative of f_1 is: $6x - 2$.

```
>> k = polyder (f1,f2)
```

```
>> d=polyder(f1,f2)
d =
    12      -6      38      -10
```

The derivative of $f_1 * f_2$ is: $12x^3 - 6x^2 + 38x - 10$

```
>> [n d] = polyder (f1,f2)
```

```
n =
      2      22     -10
d =
      1       0      10       0      25
```

The derivative of $\dfrac{3x^2 - 2x + 4}{x^2 + 5}$ is: $\dfrac{2x^2 + 22x - 10}{x^4 + 10x^2 + 25}$

Curve Fitting (Regression Analysis)

- Process of fitting a function to a set of data points.

- Many times one has some idea of the type of function that might fit the given data and will need only to determine the coefficients of the function.

- In other situations, where nothing is known about the data, it is possible to make different types of plots that provide information about possible forms of functions that might fit the data well.

Curve Fitting with Polynomials; The polyfit Function

- Polynomials can be used to fit data points in two ways:
- In one the **polynomial passes through all the data points**, and in the other the **polynomial does not necessarily pass through any of the points, but overall gives a good approximation of the data**.

Polynomials that pass through all the points:

Polynomials that do not necessarily pass through any of the points:

Polynomials that pass through all the points:

When *n* points (x_i, y_i) are given, it is possible to write a polynomial of degree n-1 that passes through all the points.
For example, **if two points are given** it is possible to write a linear equation in the form of **$y = mx + b$** that passes through the points.

With 3 points the equation has the form of $y = ax^2 + bx + c$.

With n points the polynomial has the form

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$$

The coefficients of the polynomial are determined by substituting each point in the polynomial and then solving the *n* equations for the coefficients.

Polynomials of high degree might give a large error if they are used to estimate values between data points.

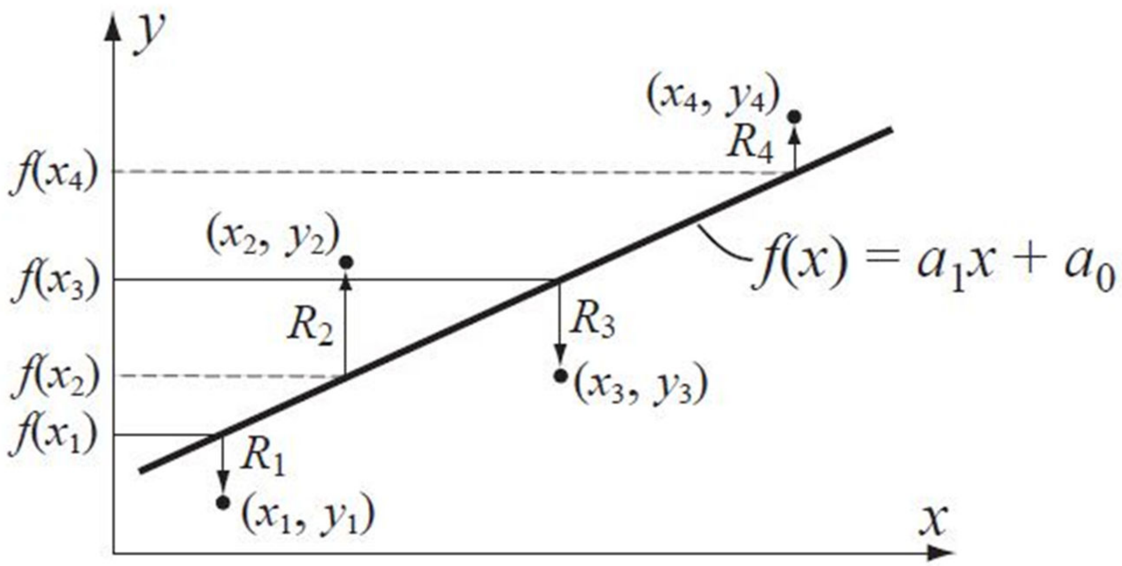
Polynomials that do not necessarily pass through any of the points:

When *n* points are given, it is possible to write a polynomial of degree less than n-1 that does not necessarily pass through any of the points, but overall approximates the data.

The most common method of finding the best fit to data points is the method of least squares.

In this method the coefficients of the polynomial are determined by minimizing the sum of the squares of the residuals at all the data points.

The residual at each point is defined as the difference between the value of the polynomial and the value of the data.



$$(x_1,y_1),\ (x_2,y_2),\ (x_3,y_3),\ \text{and}\ (x_4,y_4)$$

- The polynomial of the first degree can be written as $f(x) = a_1x + a_0$
- The residual (R_i) at each point is the difference between the value of the function at x_i and y_i , $R_i = f(x_i)-y_i$
- An equation for the sum of the squares of the residuals (R_i) of all the points:

$$R = [f(x_1)-y_1]^2 + [f(x_2)-y_2]^2 + [f(x_3)-y_3]^2 + [f(x_4)-y_4]^2$$

After substituting the equation of the polynomial at each point:

$$R = [a_1x_1 + a_0 - y_1]^2 + [a_1x_2 + a_0 - y_2]^2 + [a_1x_3 + a_0 - y_3]^2 + [a_1x_4 + a_0 - y_4]^2$$

At this stage R is a function of a_1 and a_0. The minimum of R can be determined by taking the partial derivative of R with respect to a_1 and a_0 (two equations) and equating them to zero.

$$\frac{\partial R}{\partial a_1} = 0 \quad \text{and} \quad \frac{\partial R}{\partial a_0} = 0$$

This results in a system of two equations with two unknowns, a_1 and a_0.

The solution of these equations gives the values of the coefficients of the polynomial that best fits the data.

The same procedure can be followed with more points and higher-order polynomials.

```
p = polyfit(x,y,n)
```

`p` is the vector of the coefficients of the polynomial that fits the data.

`x` is a vector with the horizontal coordinates of the data points (independent variable).
`y` is a vector with the vertical coordinates of the data points (dependent variable).
`n` is the degree of the polynomial.

Interpolation

Estimation of values between data points

- In one-dimensional interpolation each point has one independent variable (x) and one dependent variable (y).

- In two-dimensional interpolation each point has two independent variables (x and y) and one dependent variable (z).

One-Dimensional Interpolation

If only two data points exist, the points can be connected with a straight line is and a linear equation (polynomial of first order) can be used to estimate values between the points.

```
yi = interp1(x,y,xi,'method')
```

`yi` is the interpolated value.

`x` is a vector with the horizontal coordinates of the input data points (independent variable).
`y` is a vector with the vertical coordinates of the input data points (dependent variable).
`xi` is the horizontal coordinate of the interpolation point (independent variable).

Method of interpolation, typed as a string (optional).

The vector `x` must be monotonic (with elements in ascending or descending order).

`xi` can be a scalar (interpolation of one point) or a vector (interpolation of many points). `yi` is a scalar or a vector with the corresponding interpolated values.

MATLAB can do the interpolation using one of several methods that can be specified. These methods include:

`'nearest'`	returns the value of the data point that is nearest to the interpolated point.
`'linear'`	uses linear spline interpolation.
`'spline'`	uses cubic spline interpolation.
`'pchip'`	uses piecewise cubic Hermite interpolation, also called `'cubic'`

When the `'nearest'` and the `'linear'` methods are used, the value(s) of `xi` must be within the domain of `x`. If the `'spline'` or the `'pchip'` methods are used, `xi` can have values outside the domain of `x` and the function `interp1` performs extrapolation.

The `'spline'` method can give large errors if the input data points are nonuniform such that some points are much closer together than others.

Specification of the method is optional. If no method is specified, the default is `'linear'`.

Example

The following data points, which are points of the function

$$f(x) = 1.5^{x} \cos(2x)$$

x	0	1	2	3	4	5
y	1.0	–0.6242	–1.4707	3.2406	–0.7366	–6.3717

Use **linear**, **spline**, and **pchip** interpolation methods to calculate the value of y between the points. Make a figure for each of the interpolation methods.

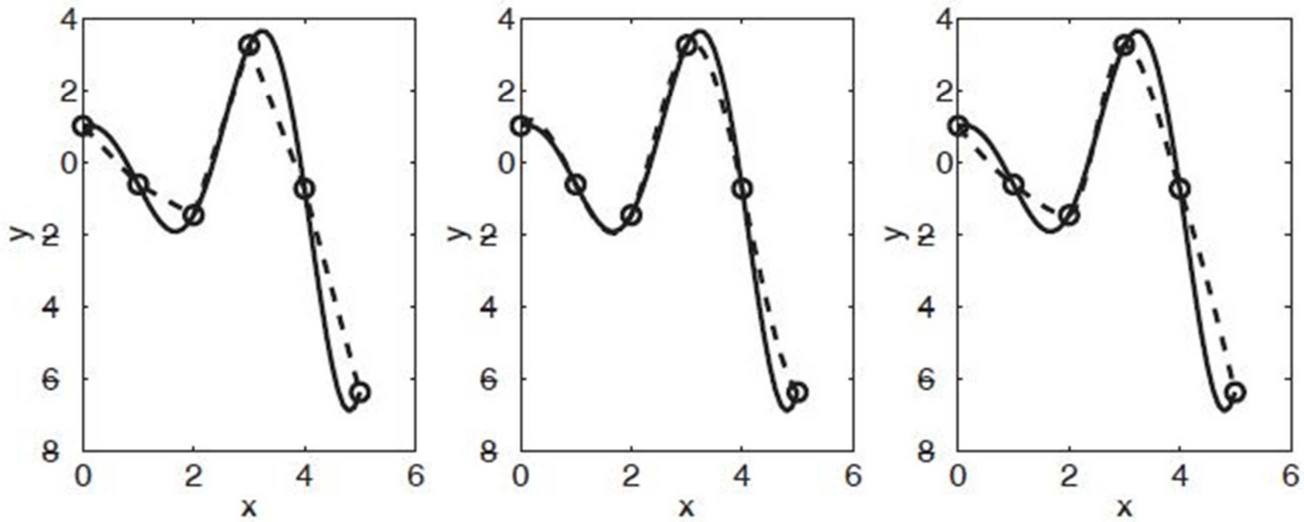
In the figure show the points, a plot of the function, and a curve that corresponds to the interpolation method.

The following is a program written in a script file that solves the problem:

```
x=0:1.0:5;                                          Create vectors x and y with coordinates of the data points.
y=[1.0 -0.6242 -1.4707 3.2406 -0.7366 -6.3717];
xi=0:0.1:5;                                         Create vector xi with points for interpolation.
yilin=interp1(x,y,xi,'linear');                     Calculate y points from linear interpolation.
yispl=interp1(x,y,xi,'spline');                     Calculate y points from spline interpolation.
yipch=interp1(x,y,xi,'pchip');                      Calculate y points from pchip interpolation.
yfun=1.5.^xi.*cos(2*xi);                            Calculate y points from the function.
```

```
subplot(1,3,1)
plot(x,y,'o',xi,yfun,xi,yilin,'--');
subplot(1,3,2)
plot(x,y,'o',xi,yfun,xi,yispl,'--');
subplot(1,3,3)
plot(x,y,'o',xi,yfun,xi,yipch,'--');
```

The three figures generated by the program are shown below (axes labels were added with the Plot Editor). The data points are marked with circles, the interpolation curves are plotted with dashed lines, and the function is shown with a solid line. The left figure shows the linear interpolation, the middle is the spline, and the figure on the right shows the pchip interpolation.



The Basic Fitting Interface

The basic fitting interface is a tool that can be used to perform curve fitting and interpolation interactively.

By using the interface the user can:

Curve-fit the data points with polynomials of various degrees up to 10, and with spline and Hermite interpolation methods.

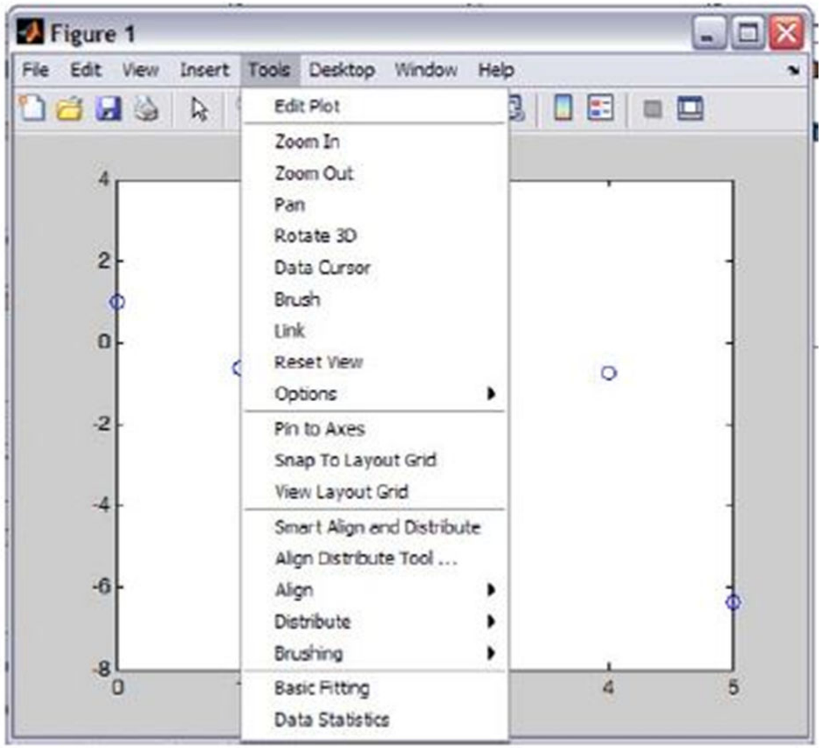
Plot the various fits on the same graph so that they can be compared.

Plot the residuals of the various polynomial fits and compare the norms of the residuals.

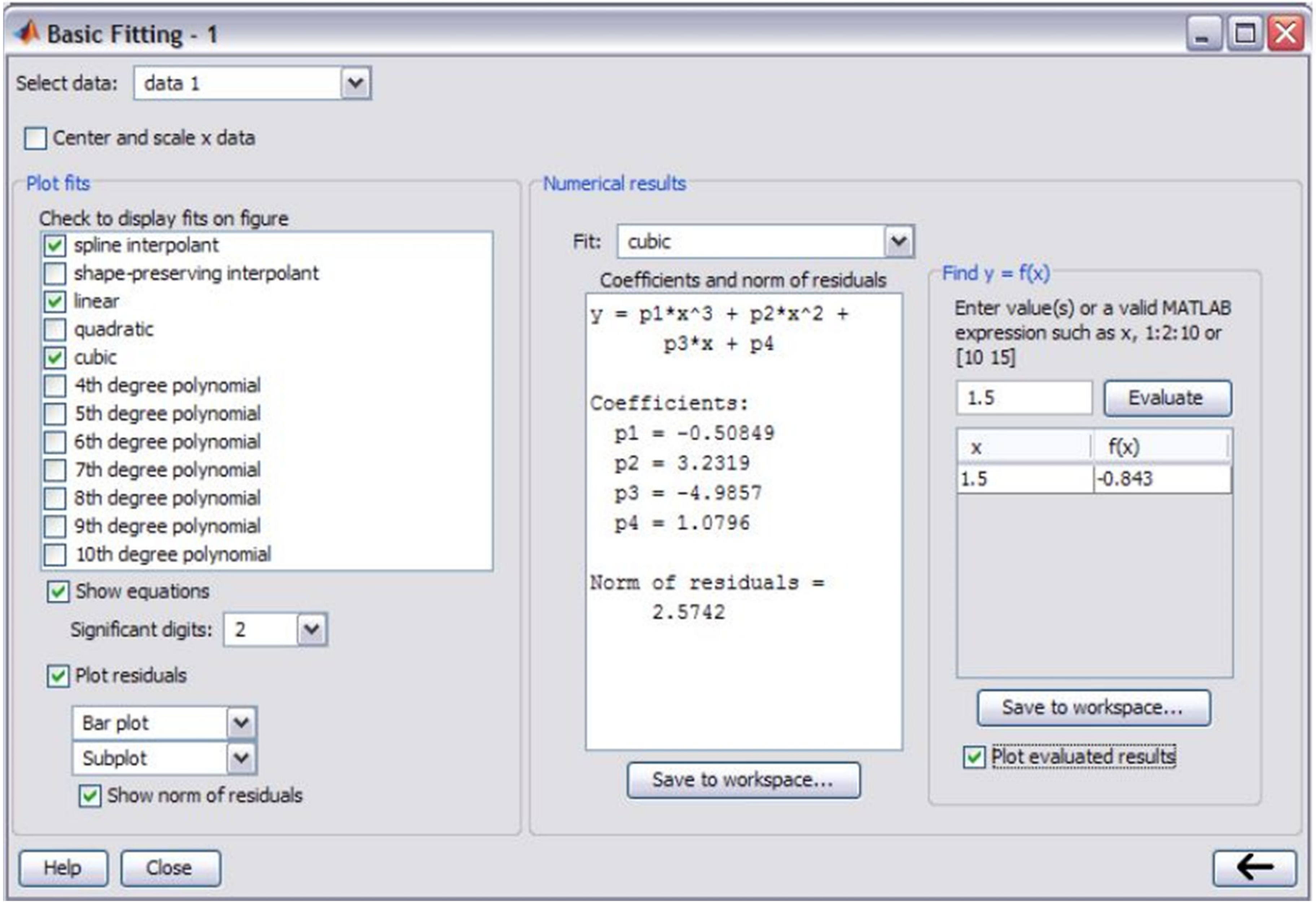
Calculate the values of specific points with the various fits.

Add the equations of the polynomials to the plot.

To activate the basic fitting interface, the user first has to make a plot of the data points. Then the interface is activated by selecting **Basic Fitting** in the **Tools** menu, as shown on the right. This opens the Basic Fitting Window.



When the window first opens, only one panel (the **Plot fits** panel) is visible. The window can be extended to show a second panel (the **Numerical results** panel) by clicking on the → button. One click adds the first section of the panel, and a second click makes the window look as:



window can be
reduced back