Curve Fitting

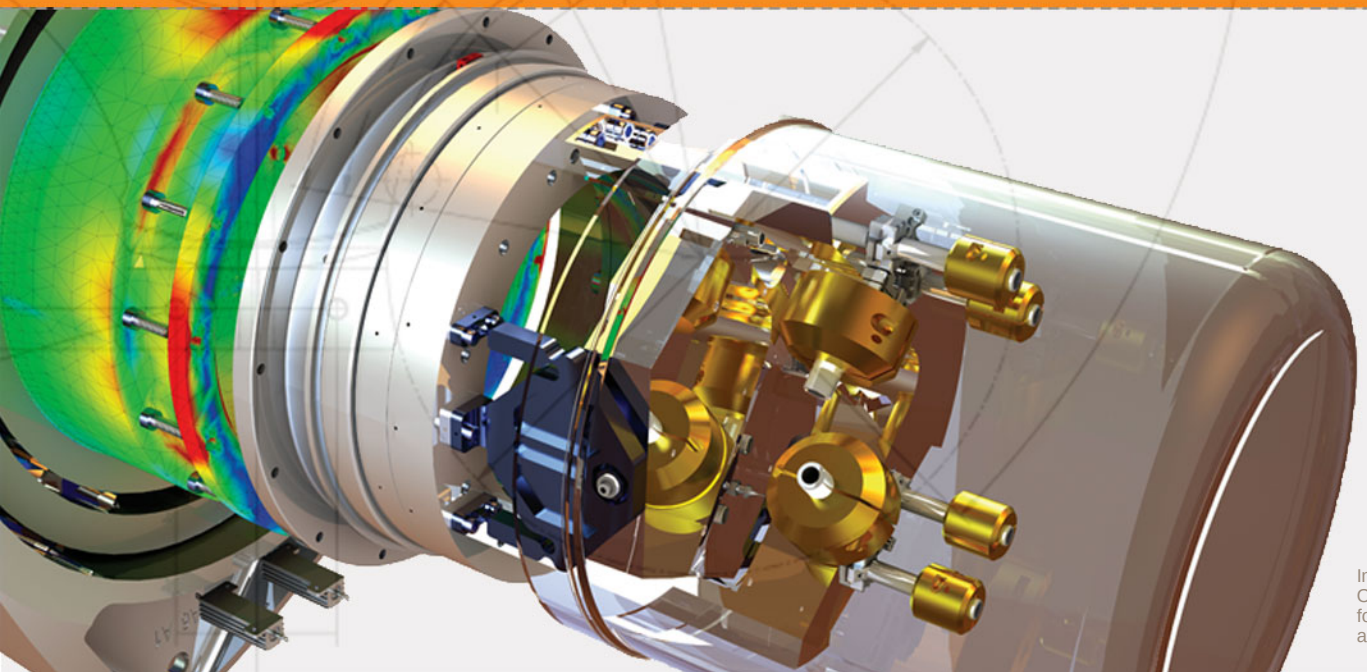


Image courtesy of National Optical Astronomy Observatory, operated by the Association of Universities for Research in Astronomy, under cooperative agreement with the National Science Foundation.

Curve Fitting

The process of finding a mathematical equation that adequately fits a given set of data points based on some constraint(s).

Examples of Applications:

1. Mathematically model a process or system based on measurement data.
2. Predict trends (future performance) based on existing data.

CURVE FITTING USING MATLAB

Basic Curve Fitting in MATLAB

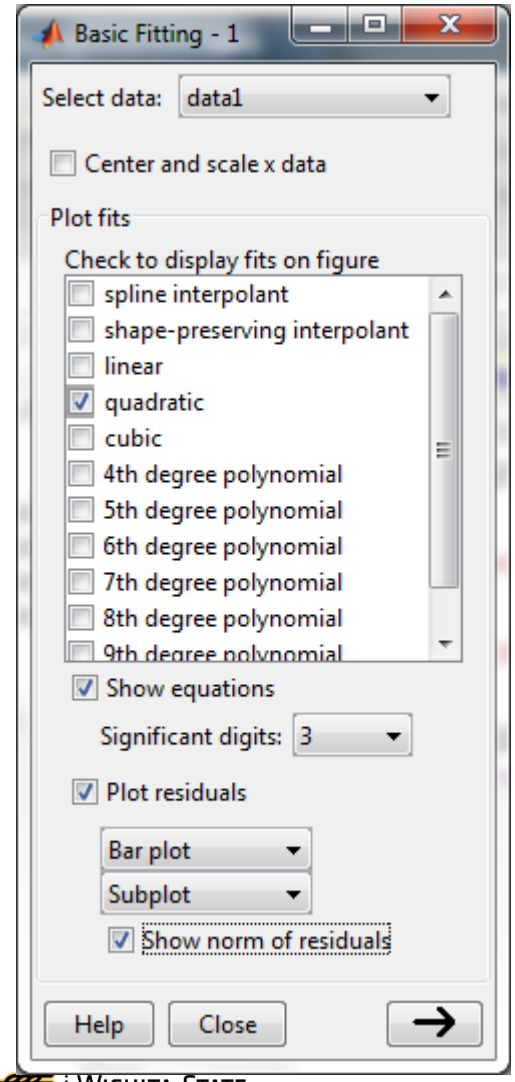
Plot the data in MATLAB. In the plot window, choose Tools → Basic Fitting to launch the Basic Fitting GUI.

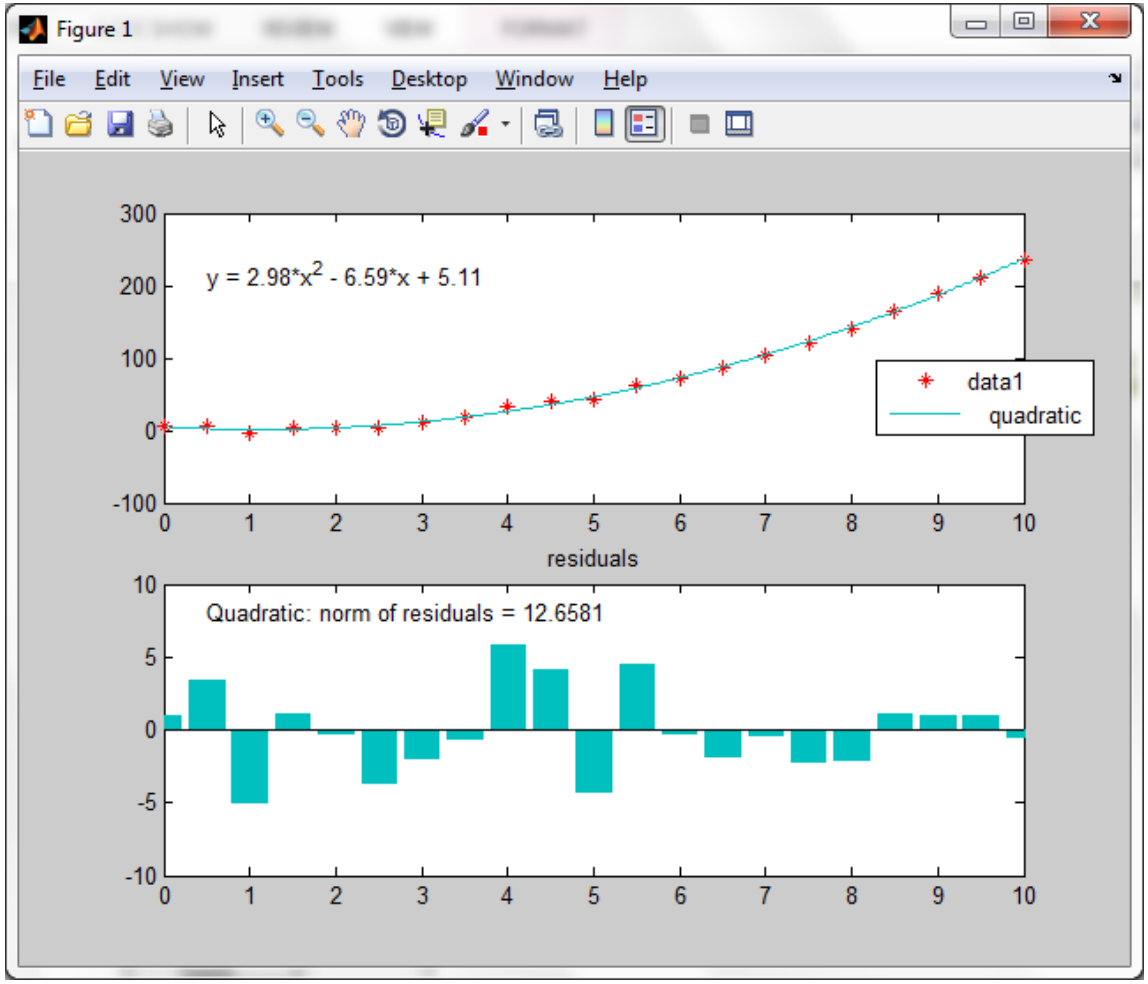
Example:

Create some noisy data

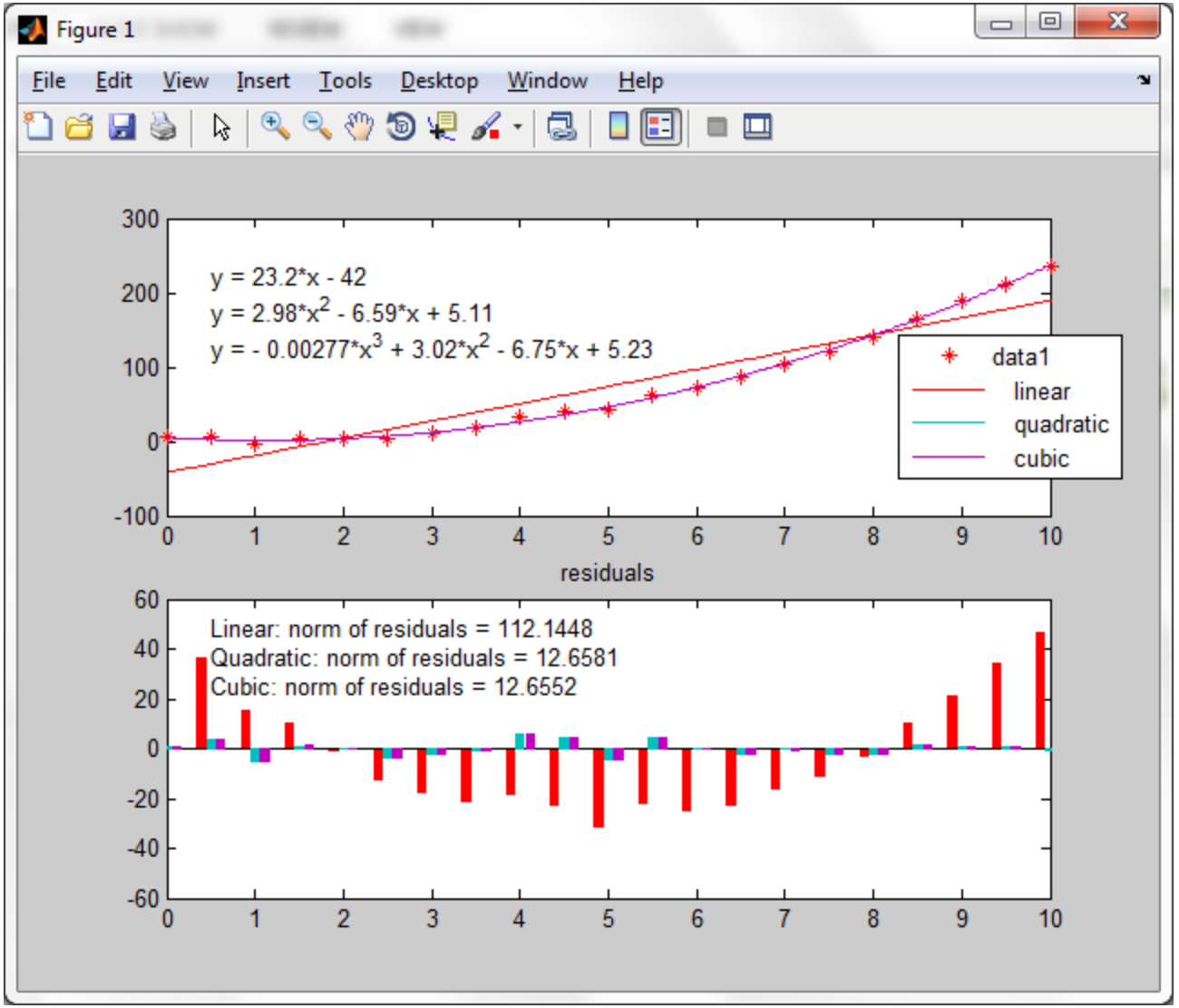
```
>> x = 0:0.5:10;
>> y = 3*x.^2-7*x+5+2*randn(1,length(x));
>> plot(x,y,'r*')
```

Basic Curve Fitting in MATLAB





Basic Curve Fitting in MATLAB



Basic Curve Fitting in MATLAB

The norm of the residuals is calculated as:

$$\|residual\| = \sqrt{\sum_{i=1}^{N} (y_i - f(x_i))^2}$$

Where x_i and y_i are the original data points, f(x) is the generated polynomial function, and N is the number of data points.

A norm of zero indicates a perfect match at the data points.

Curve Fitting in MATLAB using polyfit

MATLAB has a couple of useful functions for curve fitting:

polyfit(x, y, N) *will fit an Nth order polynomials to the set of data points (x,y). The output of* ***polyfit*** *is a vector of the numerical coefficients of the Nth order polynomial in descending order.*

polyval(polynomial, xvalues) *will plug the xvalues into a given polynomial to compute the corresponding yvalues. The polynomial argument is simply a vector of the numerical coefficients of the polynomial in descending order.*

Curve Fitting in MATLAB using polyfit

```
% Try 1st order polynomial (3rd argument in polyfit)

>> poly1st = polyfit(x,y,1)

poly1st = 23.1668  -42.0043
```

$$Equation\ of\ 1st\ order\ fitted\ polynomial\text{:}\quad y = 23.1668x - 42.0043$$

```
% Plug x Values into fitted polynomial to get new y
values
>> yfit1 = polyval(poly1st,x);

% Calculate residual norm
>> residual_1 = sqrt(sum((y-yfit1).^2))

    residual_1 = 112.1448
```

Curve Fitting in MATLAB using polyfit

```
% Try 2nd order polynomial (3rd argument in polyfit)

>> poly2nd = polyfit(x,y,2)

poly2nd =   2.9759   -6.5919    5.1136
```

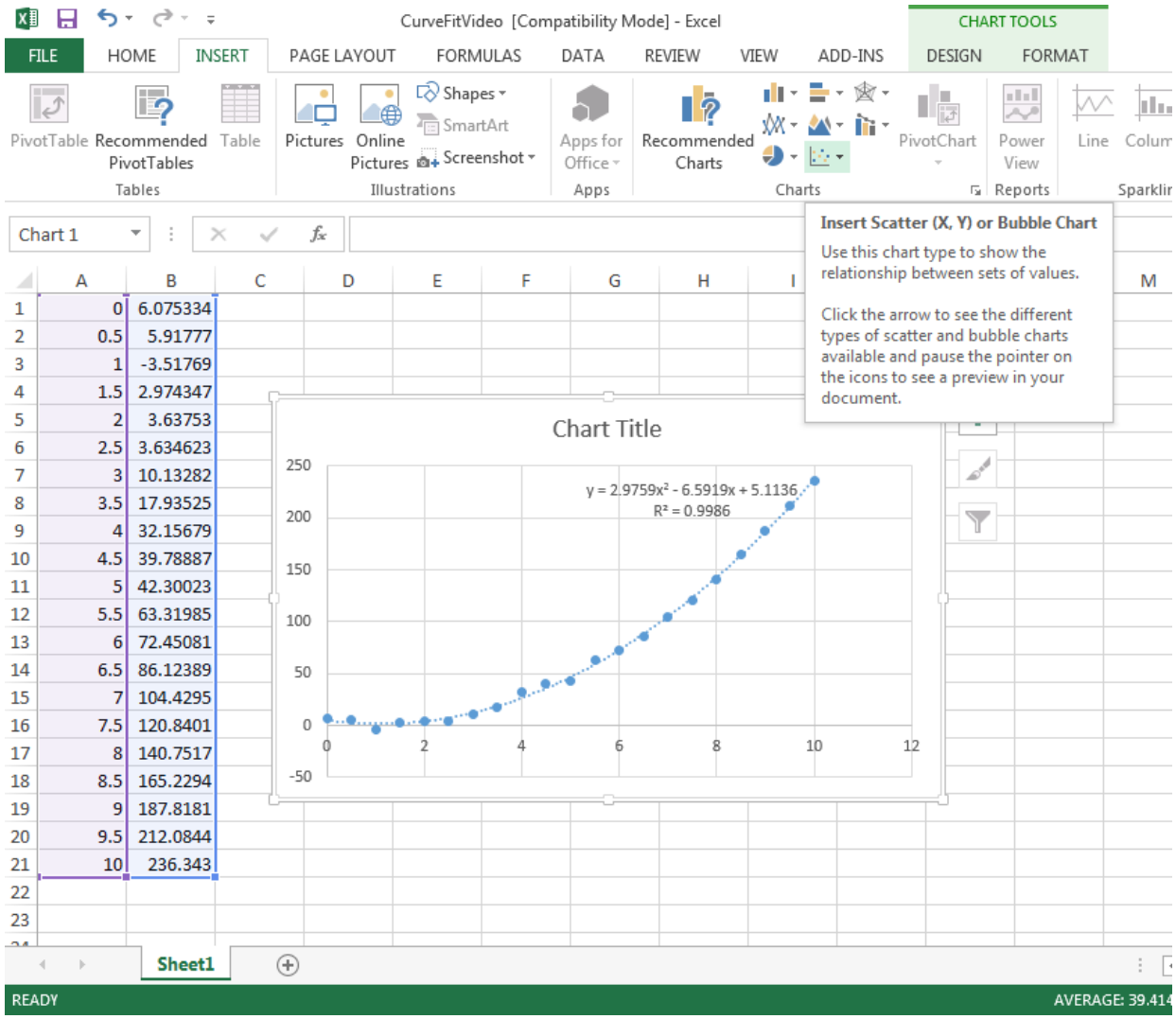
$Equation\ of\ 2nd\ order\ fitted\ polynomial\colon\quad y = 2.9759x^2 - 6.5919x + 5.1136$

```
% Plug x Values into fitted polynomial to get new y
values
>> yfit2 = polyval(poly2nd,x);

% Calculate residual norm
>> residual_2 = sqrt(sum((y-yfit2).^2))

      residual_2 = 12.6581
```

CURVE FITTING USING EXCEL

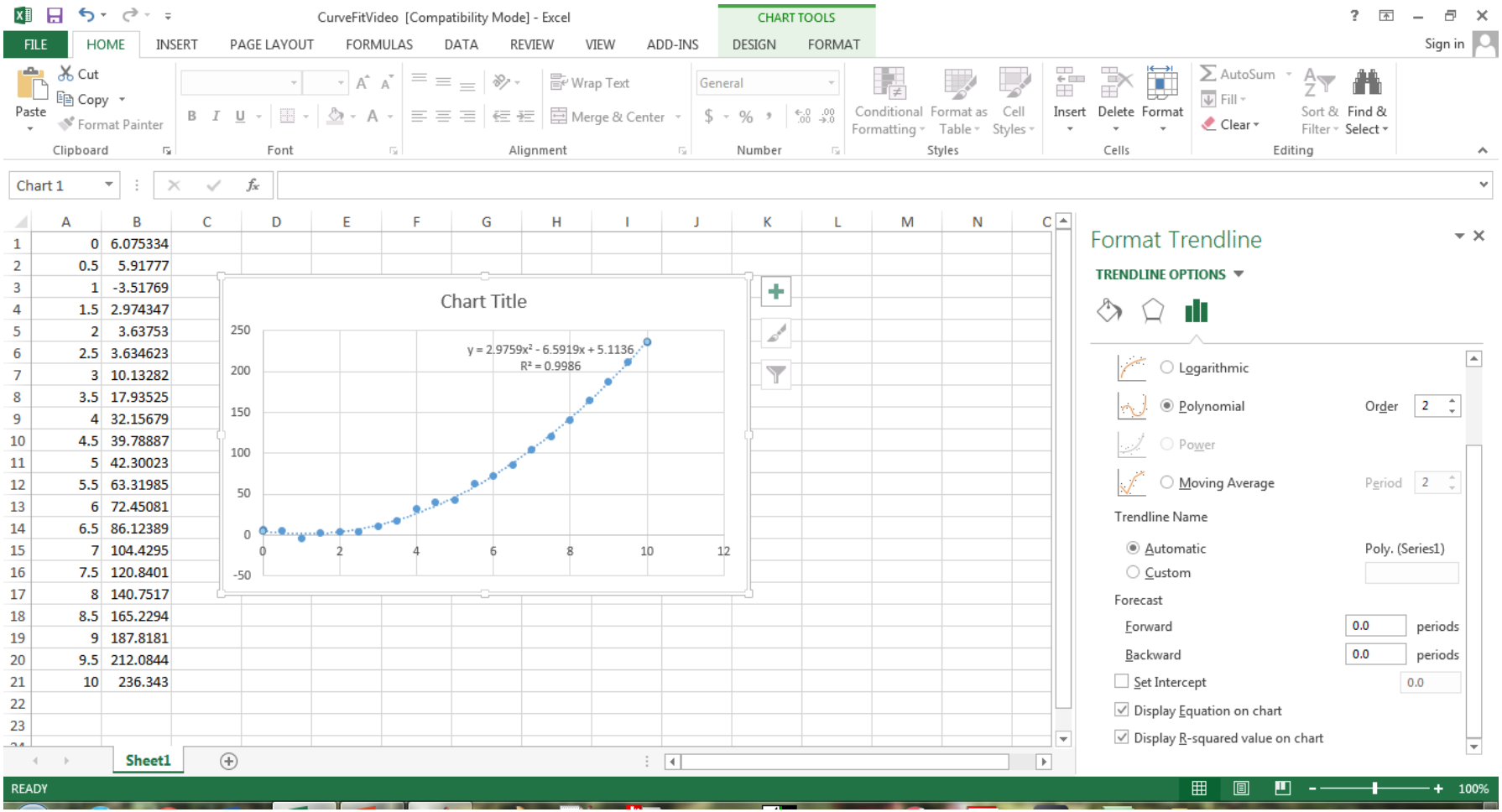
Basic Curve Fitting in Excel

1. Select the data in Excel.
2. Click on the Insert Tab.
3. Choose the Scatter plot.
4. Right click on a data point and choose Add Trendline or Click on the ChartTools Tab and select Trendline and Options



	A	B
1	0	6.075334
2	0.5	5.91777
3	1	-3.51769
4	1.5	2.974347
5	2	3.63753
6	2.5	3.634623
7	3	10.13282
8	3.5	17.93525
9	4	32.15679
10	4.5	39.78887
11	5	42.30023
12	5.5	63.31985
13	6	72.45081
14	6.5	86.12389
15	7	104.4295
16	7.5	120.8401
17	8	140.7517
18	8.5	165.2294
19	9	187.8181
20	9.5	212.0844
21	10	236.343
22		
23		

Basic Curve Fitting in Excel



	A	B
1	0	6.075334
2	0.5	5.91777
3	1	-3.51769
4	1.5	2.974347
5	2	3.63753
6	2.5	3.634623
7	3	10.13282
8	3.5	17.93525
9	4	32.15679
10	4.5	39.78887
11	5	42.30023
12	5.5	63.31985
13	6	72.45081
14	6.5	86.12389
15	7	104.4295
16	7.5	120.8401
17	8	140.7517
18	8.5	165.2294
19	9	187.8181
20	9.5	212.0844
21	10	236.343
22		
23		

Basic Curve Fitting in Excel

The R^2 value is the cross-correlation coefficient (squared) between the actual y-values and the y-values predicted by the function. It is calculated as:

$$R^2 = \frac{[\sum(y - \bar{y}) \cdot (y_{est} - \bar{y}_{est})]^2}{\sum(y - \bar{y})^2 \cdot \sum(y_{est} - \bar{y}_{est})^2}$$

where y is the vector of y-values from the original data set and y_{est} is the vector of values of the curve fitting function evaluated using the x-values of the original data set.

R^2 will be in the interval [0 1]. A value of 1 indicates a perfect match between the function and the original data set.

IMPORTING EXCEL DATA
&
EXPORTING DATA TO EXCEL

Importing Data from Excel

- Data is easily imported from Excel to MATLAB by using the xlsread function.

>> ArrayName = xlsread('*filename',CellsToBeRead)*

- The Import tool can also be used to import data from Excel.

Exporting Data to Excel

- Data is easily exported to Excel from MATLAB by using the xlswrite function. If the excel file doesn't already exist, it will be automatically created.

>> xlswrite('*filename', ArrayName, CellsToBeWrittenTo)*

- It is also possible to simply copy the data from the variable editor window and paste it into excel.

Try These Commands in MATLAB

```
>> poly1st = [-2 1]
% Numerical Coefficients of the Polynomial -2x + 1

>> polyval(poly1st,6)
% Evaluate the Polynomial at x = 6

>> poly2nd = [1 -5 2]
% Numerical Coefficients of the Polynomial x^2 - 5x + 2

>> polyval(poly2nd,4)
% Evaluate the Polynomial at x = 4
```