

# Curve fitting toolkit

Curve fitting toolkit is a powerful plug-in that consolidates the functions for curve fitting in an easy to use graphical interface. The toolbox provided the various methods of regression and interpolation combined with statistical functions that allow you to perform data analysis, process and visualize data, compare methods, optimize parameters...

To start using toolkit, first create a variable, call the function **curvefitresults** and place the data in it. The variable that we have just created will be used to display the fitting results on a graph and it will be linked between toolkit and graph.

```
a := [ 0.1  1  2  3  4  5  6
      0  0.6325  0.7402  0.1274  -0.8293  -0.9768  -0.1133 ]

f := curvefitform(0 , "form 1")

y := curvefitresult(f , a)
```

The next step is to place a curve fitting form in the widget using the combination **embed widget(f)**. More about GUI elements and widgets can be read about in GUIUserManual.pdf from the Help tab.

embed widget(f)



Fitting method

Interpolations

Method: Cubic Spline Points: 1

Fit

Interpolation/spline equations

[0.1,1]

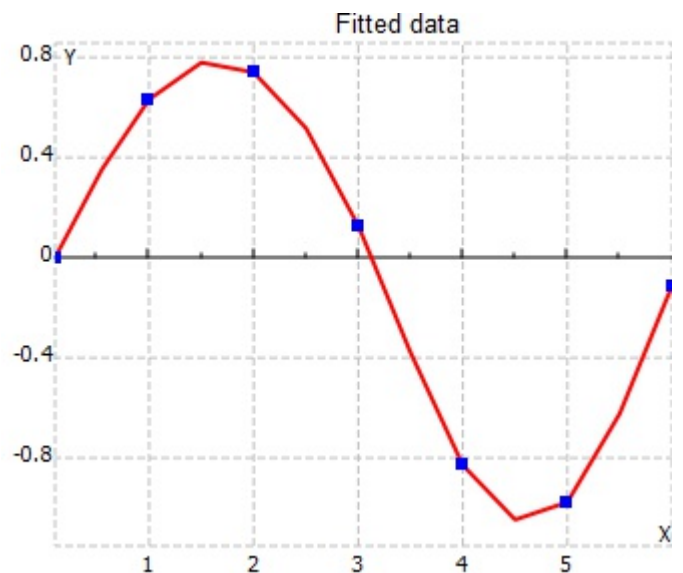
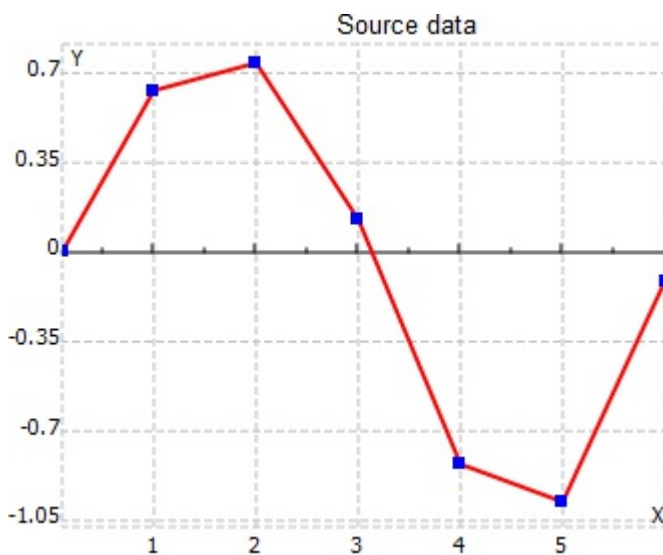
[0. 0.8101 x-0.0813-0.1376 x^3+0.0413 x^2]

Goodnes of fit

R-Square: 1 Adj. R-Square:

RMSE: SSE: 0

The final step is to insert the graph and Add variable y onto it. Variable y is the final output of the toolkit and it stores a matrix with data calculated regarding the parameters that were chosen in toolkit.



From the **Interpolation** menu you can choose the method of interpolation, number of points between two interpolation points, interval for which you want to create the equation. The list of intervals will fill out automatically with intervals which have the start and end values as two consecutive nodes.

Methods that you have at your disposal are:

linear interpolation, polynomial interpolation, ration interpolation, Cubic spline, Akima spline, Hermite spline, Cubic B spline and Bezier interpolation.

After you select the options you prefer, press the Fit button to initiate a calculation. If you change the source data, stored in variable a in our example, the toolkit values will change automatically.

From the **Regression** menu you can choose the method of regression, and degree of polynomial regression (parameter that appears only if the polynomial regression is chosen).

Outputs of the regression part of toolkit are:

- Regression equation
- R-squared statistics (in Interpolation this value is always 1)
- Adjusted R-squared statistic (not defined in Interpolation)
- RMSE: Root mean square error (not defined in Interpolation)
- SSE: Sum of squares due to error - least square error of the fit (in Interpolation this value is always 0)

Lets calculate and display the best regression method for inserted data stored in variable, a.

Choose Regressions from the first list and press the button **Fit All**. A window from the canvas bellow will appear.

Method	R-square	Adj. R-sq	RMSE	SSE	Coeff Num.
"lin"	0.369	0.2428	0.4835	1.6366	2
"exp"					0
"log"	0.1295	-0.0447	0.568	2.258	2
"pow"					0
"poly2"	0.3699	0.2439	0.4832	1.6342	3
"poly3"	0.9771	0.9725	0.0921	0.0594	4
"poly4"	0.9892	0.9871	0.0632	0.028	5
"poly5"	0.9984	0.998	0.0246	0.0042	6
"poly6"	1	1	0.	0.	7

The value from the last row, polynomial regression of sixth degree, we will not consider because there are seven points in our source data and it would become an interpolation instead of a regression (the curve goes through all the points).

So the method with the biggest R-squared and Adjusted R-squared values, and the smallest amount of errors is the polynomial regression of fifth degree. Close this window and select **Polynomial** as the Method of regression and then set the degree to 5 and press the **Fit** button.

```
r:=curvefitform(0, "form2")
```

```
xy:=curvefitresult(r, a)
```

Fitting method

Regressions ▾

Method: Polynomial ▾    Degree: 5    Equation:  $y = a_1 * x^n + a_2 * x^{(n-1)} + \dots + a_{n-1} * x + a_n$

Fit    Fit All

---

Output equation

Equation:

$y = -0.0062 x^5 + 0.1027 x^4 - 0.5169 x^3 + 0.6697 x^2 + 0.4144 x - 0.045$

---

Goodnes of fit

R-Square: 0.9984    Adj. R-Square: 0.998

RMSE: 0.0246    SSE: 0.0042

The result of our toolkit we will present on a graph.

We included variable **xy** as a result of the toolkit and variable **a** to present the source data points (variable **a** curve style is set to invisible).

```
x:=row2vec(xy, 0)
```

```
up:=row2vec(xy, 1) + 2 esterror(a, ["polynomial"  
5])
```

```
down:=row2vec(xy, 1) - 2 esterror(a, ["polynomial"  
5])
```

```
xy1:=join mat rows(x, up)
```

```
xy2:=join mat rows(x, down)
```

