A Curve and Surface Fitting Tool for *Octave*

Andreas Stahel

Version of 4th May 2022

Contents

List of Figures

List of Tables

1 Goal

Create a tool similar to the `cftool` in `MATLAB` seen in Figure 1. Essential features:

- Create a simple GUI, such that simple regression problems can be solved without the command line.
- The GUI should allow to define arbitrary functions, apply regression and display numerical and graphical results.
- Do not try to create an exact copy of `cftool()`, since there would be too many features to implement.
- Use the already available resources of the *Octave* package `optim`.

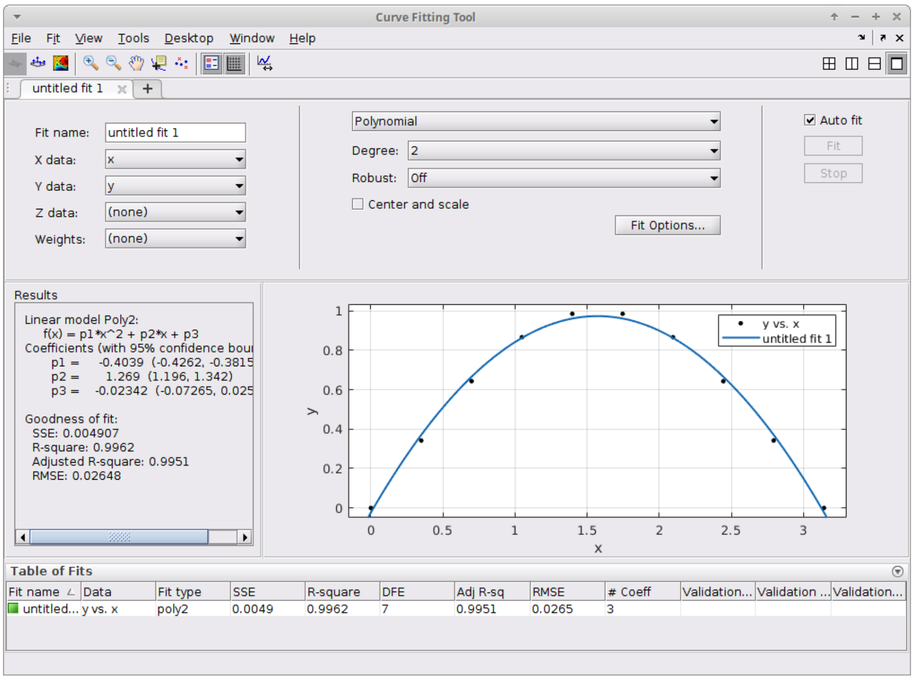

Figure 1: The very popular curve fitting tool `cftool` in `MATLAB`

2 The first versions of `FitTool.m` and `FitToolNL.m`

Currently the functions `LinearRegression()`, `nonlin_curvefit()` and `curvefit_stat()` are used to perform linear or nonlinear regression. The commands are part of the package `optim` in *Octave*. On Win* systems this is most often already installed and on Linux systems apply the commands `pkg install -forge optim`.

The code for `LinearRegression.m` can be downloaded from the web site of your author at web.sha1.bfh.science/Labs/PWF/Codes/regression/LinearRegression.m or from the new (and hopefully stable) site AndreasStahel.github.io/Octave.html .

2.1 General usage of the functions `FitTool()` and `FitToolNL()`

The tools will

`FitTool`: use linear regression to determine the best fitting linear combination of the selected basis functions to approximate the given data.

`FitToolNL`: use a nonlinear regression to attempt to find the best parameters to fit the given function to the given data.

2.1.1 Perform a linear regression, using `FitTool()`

- Generate some random data and start `FitTool()` .

Octave
```
x_in = 1.5 + 0.3*randn(10,1);  y_in = sin(x_in) + 0.01*randn(10,1);
FitTool("1D")
```

- Since the variables in the base context are called `x_in` and `y_in` their values are used inside the function `FitTool` and have to be entered in the boxes on the top left in Figure 2. Once the names of the variables are entered the raw data will be displayed.
- The expressions in the input boxes for x and y are evaluated in the base context. Thus one may apply simple operations, e.g. to read in the values of y use `sin(x_in)` in the y-box.
- The `help` button on the top right provides an elementary help description.
- The `1D/2D` button switches between the versions for one or two independent variables.
- The `export/import` button allows to export or import some of the fitting data to the base context. A dialog box will ask for the name of the variable to be exported or imported. The result is a cell array with three or four entries:
  1. The first entry is the string with the formula for the regression matrix $\mathbf{M}$ of the function $f(x)$
  2. The second entry is a column vector with the optimal values of the parameters p_i .
  3. The third entry is a column vector with the estimated values of the standard deviations of the parameters p_i .
  4. For `FitToolNL()` the forth entry is a row vector with the initial values of the parameters used.

  On import the formula for the regression matrix $\mathbf{M}$ or the function $f(x)$ is copied into the corresponding box in the GUI, but not evaluated. Hit the `Enter` key in the box to evaluate.
- For `FitToolNL()` the the optimal values of the parameters will used as new initial values for the parameters. This allows the iteration to be run with other initial values for the parameters.

2.1.2 Linear regression with one independent variable

- Within `FitTool` select a quadratic polynomial $f(x) = p_1\,x^0 + p_2\,x^1 + p_3\,x^2$ leading to the string[1]
$$\mathbf{M} = [\,x.\wedge 0,\ x.\wedge 1,\ x.\wedge 2]$$
in the box for the matrix $\mathbf{M}$ to be used for the linear regression.

[1] The silly trick of x^0 generates a vector of the correct size filled with numbers 1.

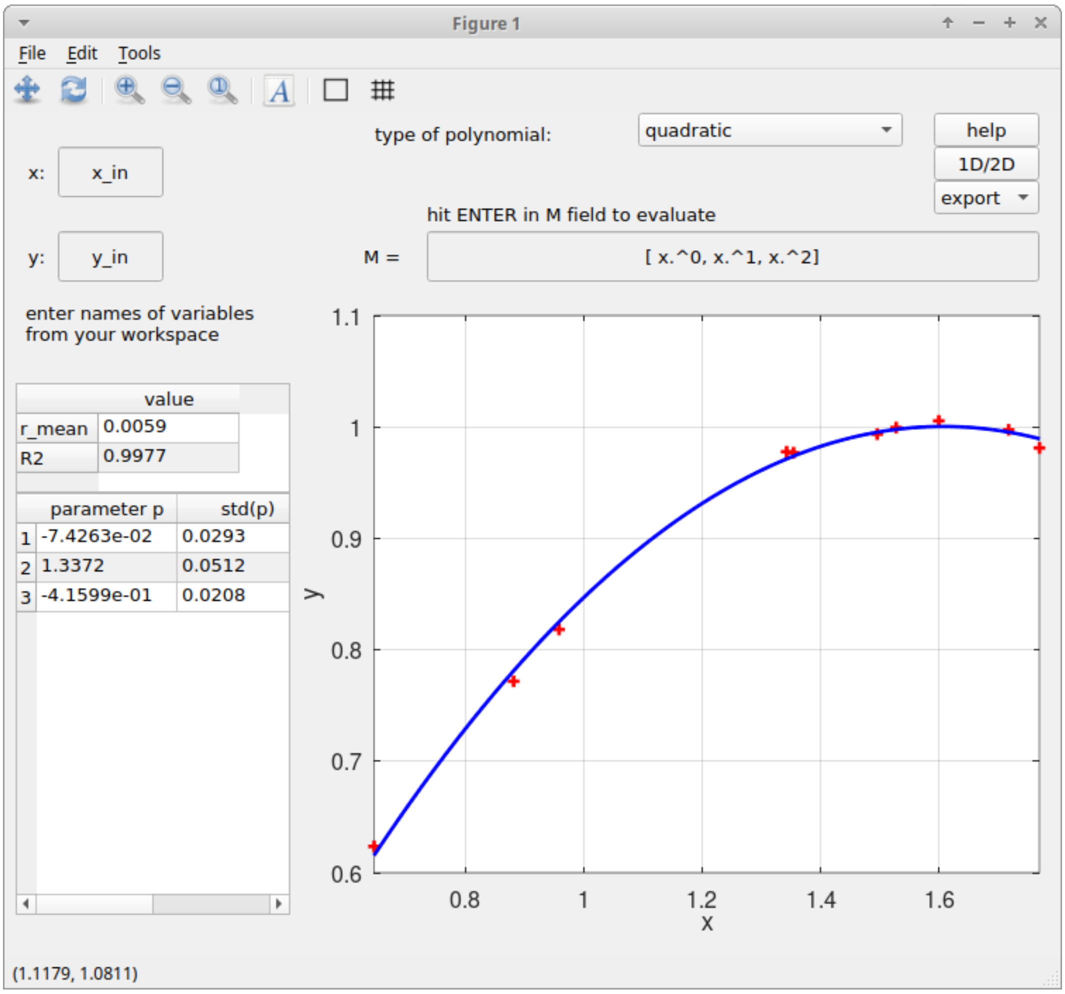

Figure 2: The curve fitting tool FitTool, using one independent variable

- You may edit the box with the regression matrix. Hit the Enter key in the box and the linear regression will be performed.
- In Figure 2 the raw data and the best fitting parabola are displayed.
- On the left find the numerical results:
  - The mean residual
$$r\text{mean} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}\ .$$
  - The square of the Pearson correlation coefficient $R2$.
  - The values of the optimal parameters and their estimated standard deviation. In the above example
$$\begin{aligned} p_1 &= -0.0743 \pm 0.0239 \\ p_2 &= +1.3372 \pm 0.0512 \\ p_3 &= -0.4160 \pm 0.0208 \end{aligned}\ .$$
- In the drop down selection of functions only polynomials are provided. The box with the regression matrix $\mathbf{M}$ can be edited and thus any linear regression can be performed. To fit a function $f(x) = p_1 \sin(3\,x) + p_2 \cos(3\,x) + p_3 \exp(x)$ use the string
$$\mathbf{M} = [\sin(3 * x),\ \cos(3 * x),\ \exp(x)\,]\ .$$
This string could also be used to import a cell structure into the GUI FitTool. In the base context create FitExpTrig.

Octave
```
FitExpTrig{1} = "[ sin(3*x), cos(3*x), exp(x) ]" ;
FitExpTrig{2} = [ 0, 0, 0] ;
FitExpTrig{3} = [ 0, 0, 0] ;
```
Then import FitExpTrig into the GUI, using export/import, and perform the regression by using the Enter key in the matrix box.

2.1.3 Linear regression with two independent variables

- By clicking on the 1D/2D button the GUI switches between one or two independent variables.
- Obviously values of x, y and z are required now. In the base context (i.e. the usual command window of *Octave*) create the data.

Octave
```
x = 2*randn(5); y = 2 + randn(5);
z = 1 + x - y + 0.5*x.*y;
```
- Then select a quadratic polynomial of two variables to obtain Figure 3. The numerical results confirm the best fit
$$f(x, y) = 1 + x - y + 0.5\,x \cdot y$$
with outstanding accuracy. This is no surprise, since exact data was used.
- For functions of two variables fewer functions are predefined, see Table 1.

  By editing the box with the regression matrix $\mathbf{M}$ any linear combination of functions can be used.

Constant	$f(x,y) = p_1$
Linear	$f(x,y) = p_1 + p_2\,x + p_3\,y$
Quadratic	$f(x,y) = p_1 + p_2\,x + p_3\,y + p_4\,x^2 + p_5\,x\,y + p_6\,y^2$
Cubic	$f(x,y) = p_1 + p_2 x + p_3 y + p_4 x^2 + p_5 x\,y + p_6 y^2 + p_7 x^3 + p_8 x^2 y + p_9 x\,y^2 + p_{10} y^3$

Table 1: Predefined functions for linear regression with two variables

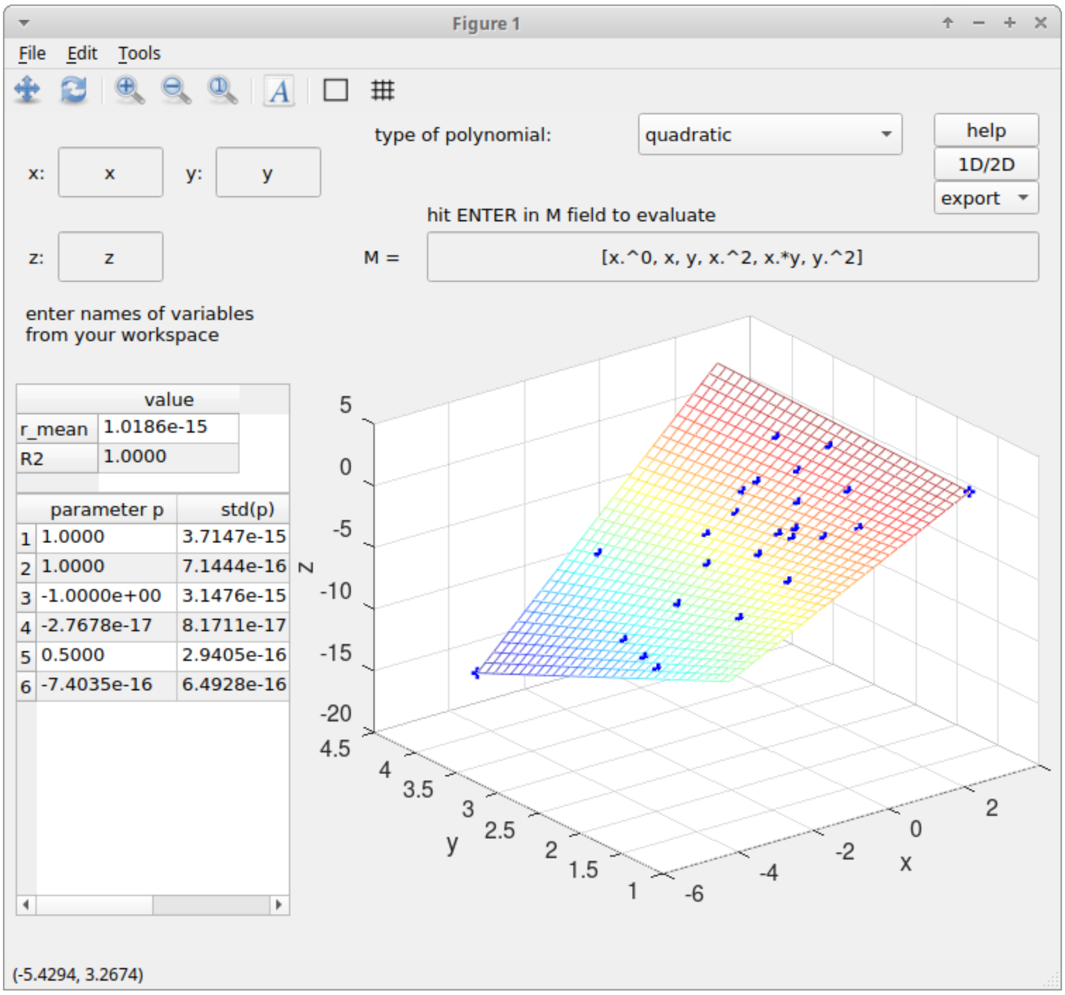

Figure 3: The curve fitting tool FitTool, using two independent variables

2.1.4 Nonlinear regression with one independent variable, using FitToolNL()

There is a second tool `FitToolNL.m` to perform a nonlinear regression with respect to one or two independent variables. It is using the commands `nonlin_curvefit()` and `curvefit_stat()` from the package `optim`. Thus the package has to be available.

The usage of `FitToolNL()` is very similar to the above `FitTool()`.

- First generate the data and then launch the tool.

Octave
```
x = linspace(0,1,20);
FitToolNL()
```

- To work with the generated data $y = \exp(-x^2)$ enter the string $\exp(-x. \wedge 2)$ into the y–box.
- Then select a rational function of the type

$$f(x) = \frac{p_1 + p_2\,x}{1 + p_3\,x + p_4\,x^2}\,.$$

  The parameters p_i are all initialized to 0. Using the box with the function and hitting the `Enter` key in the "$f(x) =$" field leads to Figure 4.

- Since you can edit the function $f(x)$ freely any type of nonlinear regression can be performed. The critical part is most often the choice of good initial values for the parameters p_i. The tool only provides naive initial guesses.
- The tool provides a few prototypes for functions for nonlinear regression with one independent variable, see Table 2.

Polynomial, linear	$f(x) = p_1 + p_2\,x$
Polynomial, quadratic	$f(x) = p_1 + p_2\,x + p_3\,x^2$
Polynomial, cubic	$f(x) = p_1 + p_2\,x + p_3\,x^2 + p_4\,x^3$
Trigonometric	$f(x) = p_1\,\cos(p_3\,x) + p_2\,\sin(p_3\,x)$
Trigonometric and linear	$f(x) = p_1\,\cos(p_3\,x) + p_2\,\sin(p_3\,x) + p_4 + p_5\,x$
Exponential	$f(x) = p_1\,\exp(p_2\,x)$
Exponential and constant	$f(x) = p_1 + p_2\,\exp(p_3\,x)$
Exponential and linear	$f(x) = p_1 + p_2\,x + p_3\,\exp(p_4\,x)$
Exponential and trigonometric	$f(x) = p_1\,\exp(p_2\,x) + p_3\cos(p_5\,x) + p_4\,\sin(p_5\,x)$
Double exponential	$f(x) = p_1\,\exp(p_2\,x) + p_3\,\exp(p_4\,x)$
Gaussian	$f(x) = p_1\,\exp(\frac{(x-p_2)^2}{2\,p_3^2})$
Rational 1/2	$f(x) = \frac{p_1+p_2\,x}{1+p_3\,x+p_4\,x^2}$
Rational 2/2	$f(x) = \frac{p_1+p_2\,x+p_3\,x^2}{1+p_4\,x+p_5\,x^2}$

Table 2: Predefined functions for nonlinear regression with one variable

2.1.5 Nonlinear regression with two independent variables

The procedure is similar to `FitToolNL()` with one independent variable. Create some data and launch the GUI.

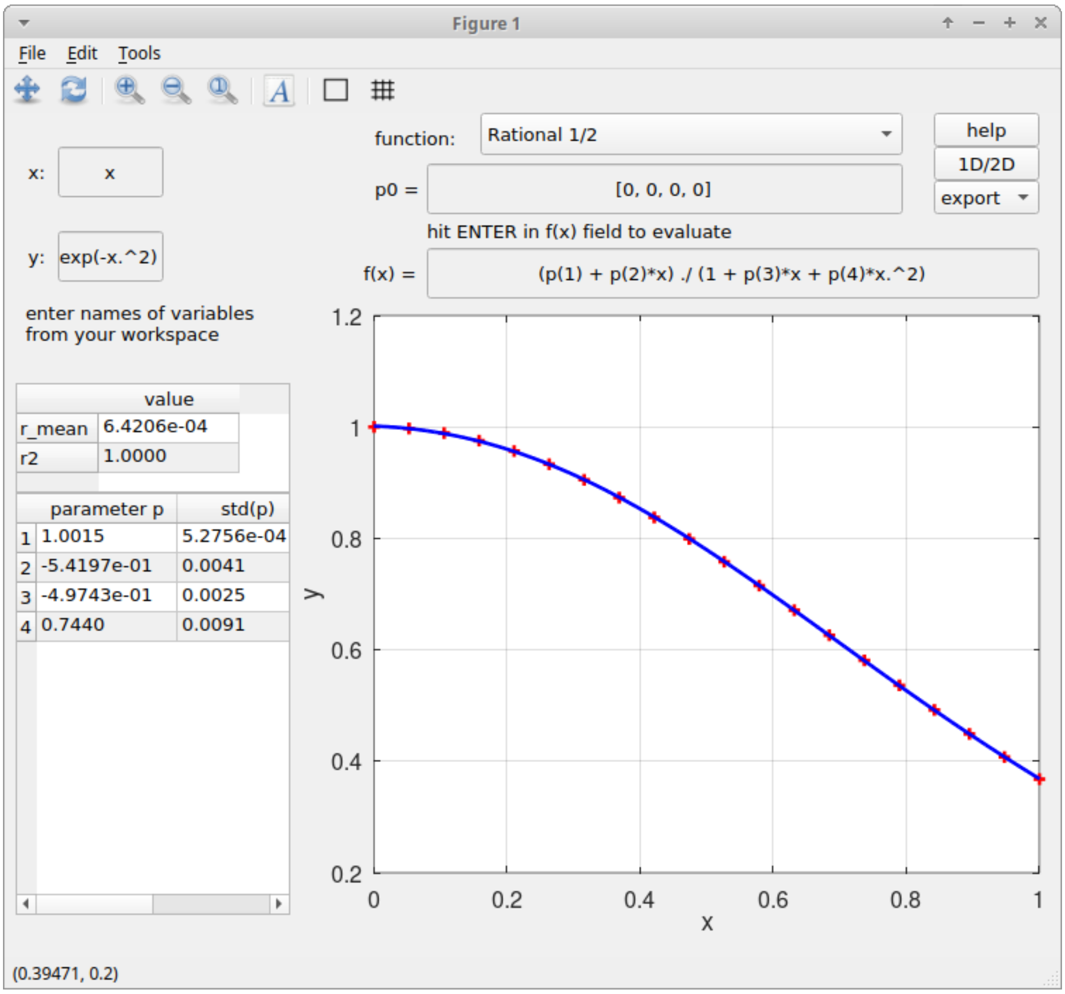

Figure 4: The nonlinear curve fitting tool `FitToolNL`, using one independent variable

Octave
```
x = 2 + randn(6); y = 2*randn(6);
z = 1 - x + x.^2 - 3*x.*y + 0.1*randn(6);
FitToolNL("2D")
```
Then select a quadratic polynomial of two variables to obtain Figure 5.

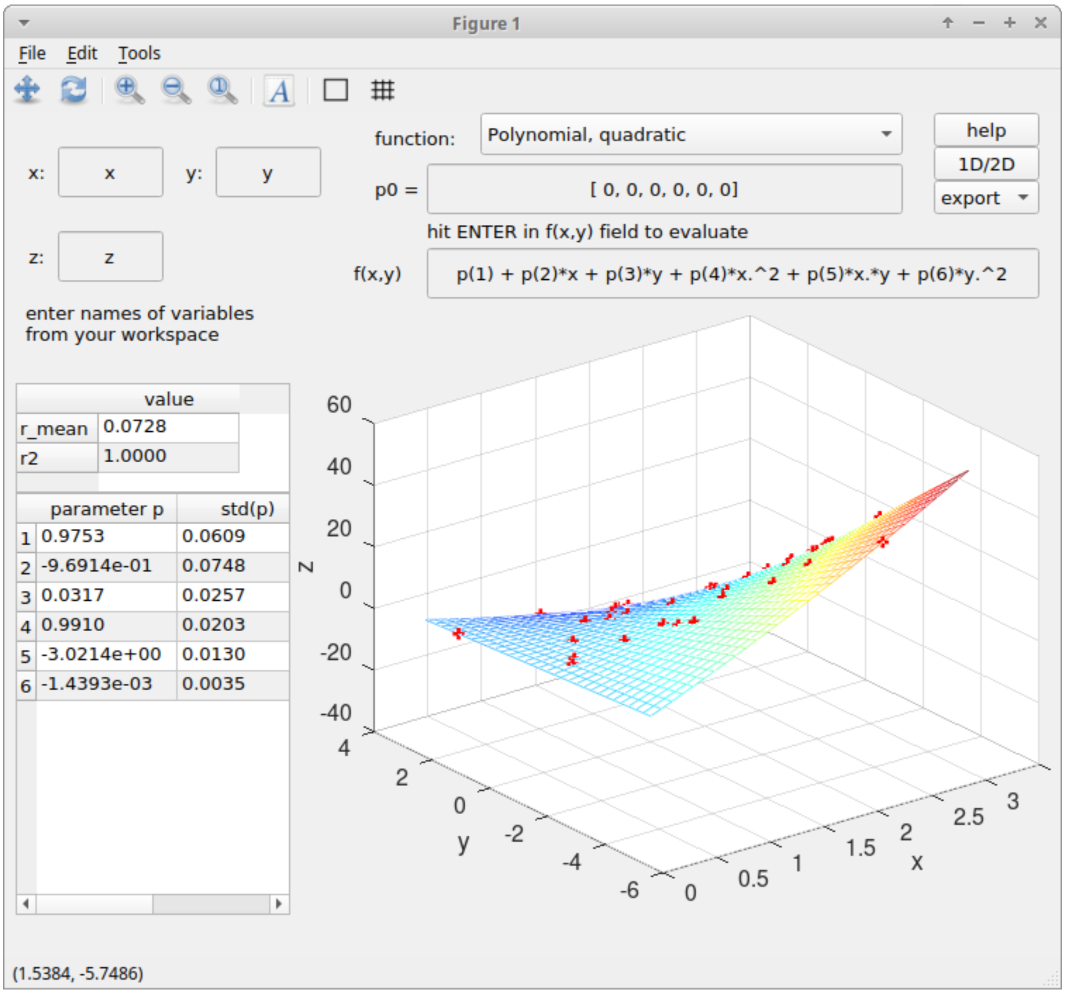

Figure 5: The nonlinear curve fitting tool FitToolNL, using two independent variables

For functions of two variables fewer functions are predefined, see Table 3. By editing the function $f(x, y)$ a nonlinear regression with any type of function can be attempted.

Polynomial, linear	$f(x,y) = p_1 + p_2\,x + p_3\,y$
Polynomial, quadratic	$f(x,y) = p_1 + p_2\,x + p_3 y + p_4\,x^2 + p_5\,x\,y + p_6\,y^2$
Trigonometric	$f(x,y) = p_1\,\cos(p_3\,x) + p_2\,\sin(p_3\,x) + p_4\,\cos(p_6\,y) + p_5\,\sin(p_6\,y)$
Exponential	$f(x,y) = p_1\,\exp(p_2\,x + p_3 y)$
Exponential and constant	$f(x,y) = p_1 + p_2\,\exp(p_3\,x + p_4 y)$
Gaussian	$f(x,y) = p_1\,\exp(-(p_4(x-p_2)^2 + p_5(x-p_2)(y-p_3) + p_6(y-p_3)^2))$

Table 3: Predefined functions for nonlinear regression with two variables