How to approximate 2D data with nonlinear multiparametric function using symbolic C# framework

Sergey L. Gladkiy

Introduction

This short tip explains how to implement nonlinear multiparametric approximation of 2D data using symbolic C# framework **ANALYTICS**.

Approximation is widely used in many scientific applications: economical, statistics, physics and other. The most popular method of approximation is the least squares method (https://en.wikipedia.org/wiki/Least_squares). This method is simply formulated and can be used for approximating functions of any dimension. The least squares approximation can implemented for both linear and nonlinear problems. As the linear least squares is rather simple and straightforward (https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics)), the nonlinear formula can be very complicated and require special nonlinear solution methods to handle it (https://en.wikipedia.org/wiki/Non-linear_least_squares).

The symbolic C# framework **ANALYTICS** includes special numerical tools for implementing as linear as nonlinear least squares approximation. These numerical tools are totally integrated with the symbolic capabilities of the framework and so, the approximation task can be solved with simple, minimal code.

Example

Let us consider the example of surface data fitting (approximation) with the 2D Gaussian function. The Gaussian distribution of 2 variables **x** and **y** (https://en.wikipedia.org/wiki/Gaussian_function) has the following math expression:

$$f(x,y) = A\exp\left(-\frac{(x-x_0)}{2\sigma_x^2} - \frac{(y-y_0)}{2\sigma_y^2}\right)$$

Except the amplitude **A**, the distribution nonlinearly depends on four parameters: center coordinates $\mathbf{x_0}$, $\mathbf{y_0}$ and deviations $\boldsymbol{\sigma_x}$, $\boldsymbol{\sigma_y}$. The task of fitting data with the Gaussian function is to find optimal values of the four parameters, those minimize the error of the distribution and some experimental data $\mathbf{z_i(x_i,y_i)}$**, i=1..N**.

Let we have in our program some experimental data for Gaussian distribution as two arrays:

```
double[][] xyData;
double[] zData;
```

where the '**xyData**' is the array of 2D points $\mathbf{(x_i,y_i)}$ (xyData[i] = [x_i, y_i], i=0..N-1) and the '**zData**' is the array of function values (zData[i] = f(x_i,y_i), i=0..N-1).

Then the approximation task can be solved by the numerical tool with the following code:

```
// Basis 2D function nonlineary depending on 4 coefficients.
string function = "e^-((x-xₒ)^2/(2*σₓ^2)+(y-yₒ)^2/(2*σᵧ^2))";
string[] variables = new string[] { "x", "y" };
string[] coefficients = new string[] { "xₒ", "yₒ", "σₓ", "σᵧ" };
// Creating nonlinear 2D (two variables) basis for approximation.
```

```
NonlinearBasis basis = new NonlinearScalarBasis(variables, coefficients, null, function);

// Using Gauss-Newton approximator.
NonlinearApproximator appr = new GaussNewtonLeastSquares();
// Set initial guess for coefficient values to solve nonlinear problem.
appr.C0 = new double[] { 0.1, 0.1, 0.1, 0.1 };
// Define the nonlinear solution options.
appr.Options = new SolverOptions() { Precision = 1e-8, Norm = new EuclidianNorm() };
// Approximate the data (calculating optimal coefficient values).
double[] cValues = appr.Approximate(basis, xyData, zData);

// Use found coefficients cValues with the basis instance...
```

The solution consists of the following steps.

Step 1: Set up the approximation function expression (Gaussian distribution).

Step 2: Create array of variable names (**x** and **y**).

Step 3: Create array of four distribution coefficient names ($\mathbf{x_0}$, $\mathbf{y_0}$, $\boldsymbol{\sigma_x}$, $\boldsymbol{\sigma_y}$).

Step 4: Create the nonlinear basis instance with specified data.

Step 5: Create the appropriate nonlinear approximator instance (Gauss-Newton least squares).

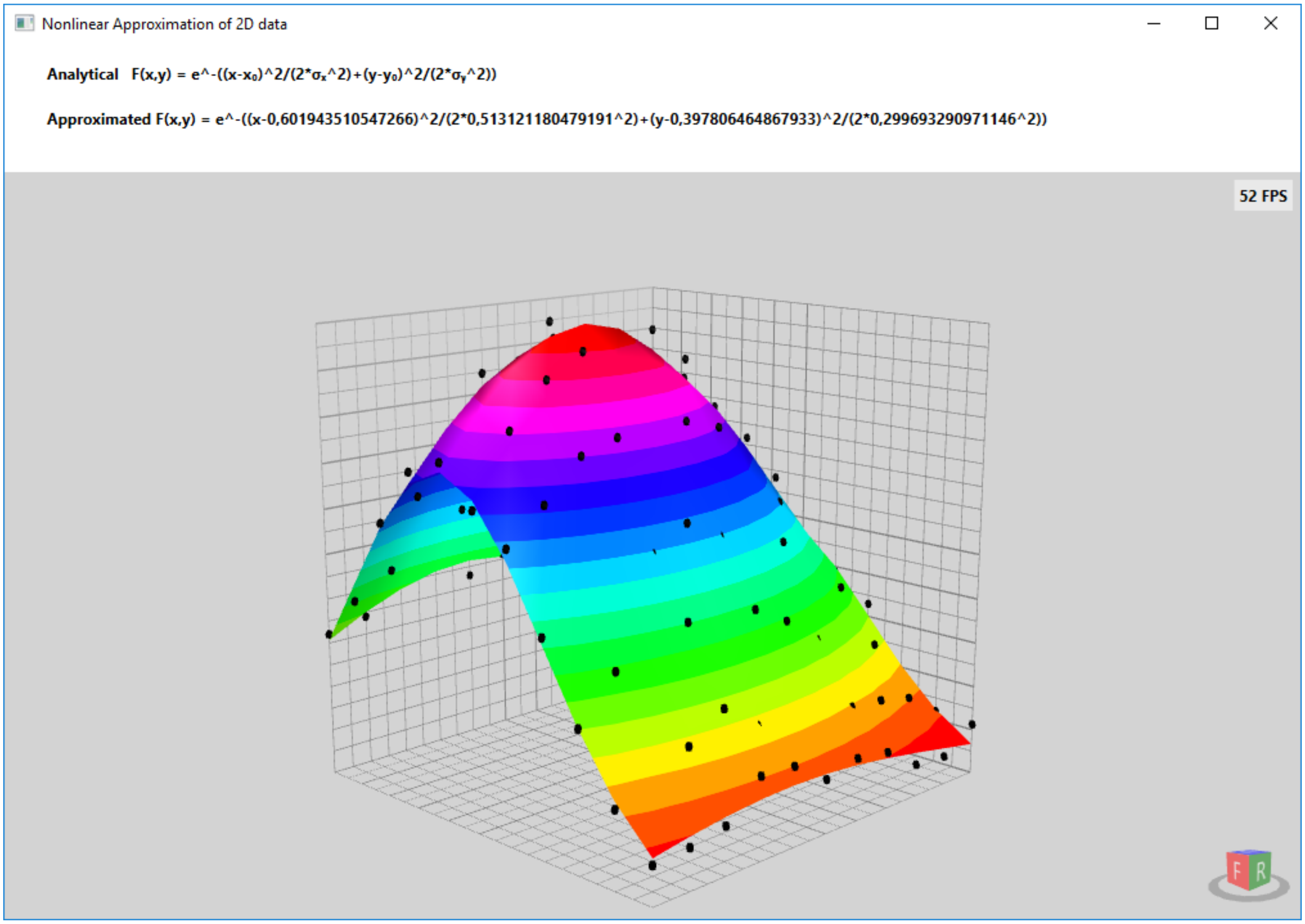
Step 6: Set up initial guess of coefficient values for nonlinear solution.

Step 7: Set up appropriate nonlinear solution options.

Step 8: Solve the problem (finding optimal coefficient values).

When the optimal values of the Gauss distribution parameters found, they can be used with the ***basis*** instance to calculate the distribution function in any specified point **(x,y)** or for other analysis methods, like derivative calculations and so on.

On the pictures below there are results of the approximation made with some data: points – experimental data (generated example data with random noise); color surface – found Gaussian function.



As can be seen from the code above, the nonlinear approximation problem can be solved with simple and straightforward code, using the symbolic representation of basis (approximation) function and other parameters. The symbolic (string) data can be then easily used for other symbolic manipulations, like derivative calculation, transferring data by the network, serializing data as text or XML, show the data to the user in readable form (as math expression).

The source code of the examples (VS 2010 solution) can be found in the download for the tip. The symbolic framework **ANALYTICS** can be found here http://sergey-l-gladkiy.narod.ru/index/analytics/0-13.