PH3205-Computational Physics

SPRING 2022

Bipradeep Saha (19MS135)

Indian Institute of Science Education and Research, Kolkata,
Mohanpur, West Bengal, 741246, India.

March 16, 2022

Assignment 1

Aim

In this assignment we need to fit a given 2D surface.

Approach

I first implemented it in *Matlab* and the code can be found in the file *Plot_a_fit_surface.m*
The code fits the given surface using a polynomial function whose degree in x and y is determined by the user. Given a surface, one can always approximate it using a polynomial of sufficiently higher degree. Thus polynomial model is a good choice.

However, it is observed that the curve fitting toolbox in *Matlab* doesn't work well enough for custom fitting surfaces specifically for trigonometric functions. Thus we shift to Python when we need to work with large complex functions.

For fitting with python, a Jupyter notebook is provided *surface_fitting.ipynb* along with python file containing the same thing: *surface_fitting.py*
In Python we generate a surface from the set of data point. The Surface is given by the equation:

$$f(x, y) = e^x \sin(y) + e^y \sin(x) + \cos(x + y)$$

Then some noise is added to the data points and it was fitted to the surface. where the fitting function is

$$g(x, y) = A \times e^x \sin(y) + B \times e^y \sin(x) + C \times \cos(x + y)$$

After fitting we get the following values of the parameters

- $A = 0.9992599657273723$
- $B = 1.0005971885083516$
- $C = 1.0040030208579842$

We can clearly see that the best fit surface approximates the given surface reasonably well as $A, B, C \approx 1$. Below we attach the contour plot($Contour\_plot.jpg$)

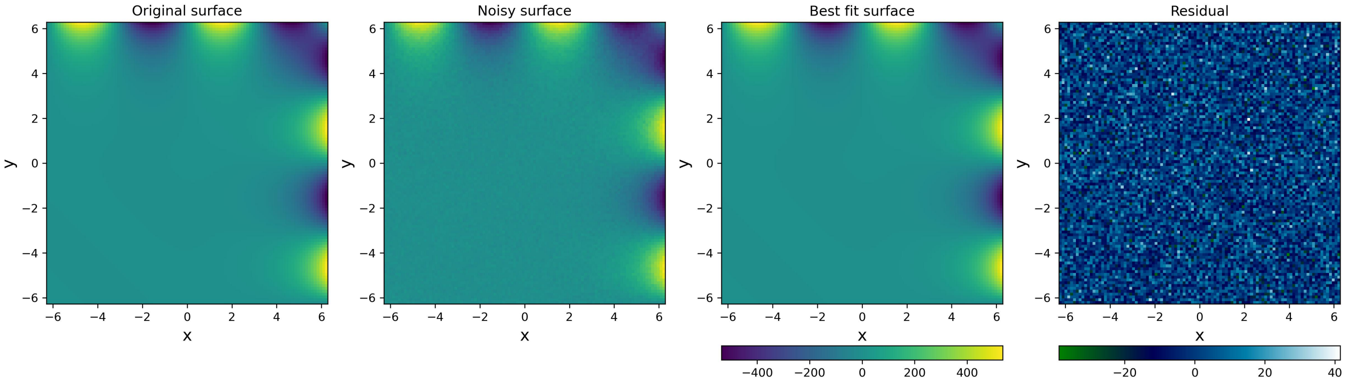

Figure 1: contour plot of the fitting process

One can also look at the 3D surface plot($3D\_surface\_plot.jpg$)

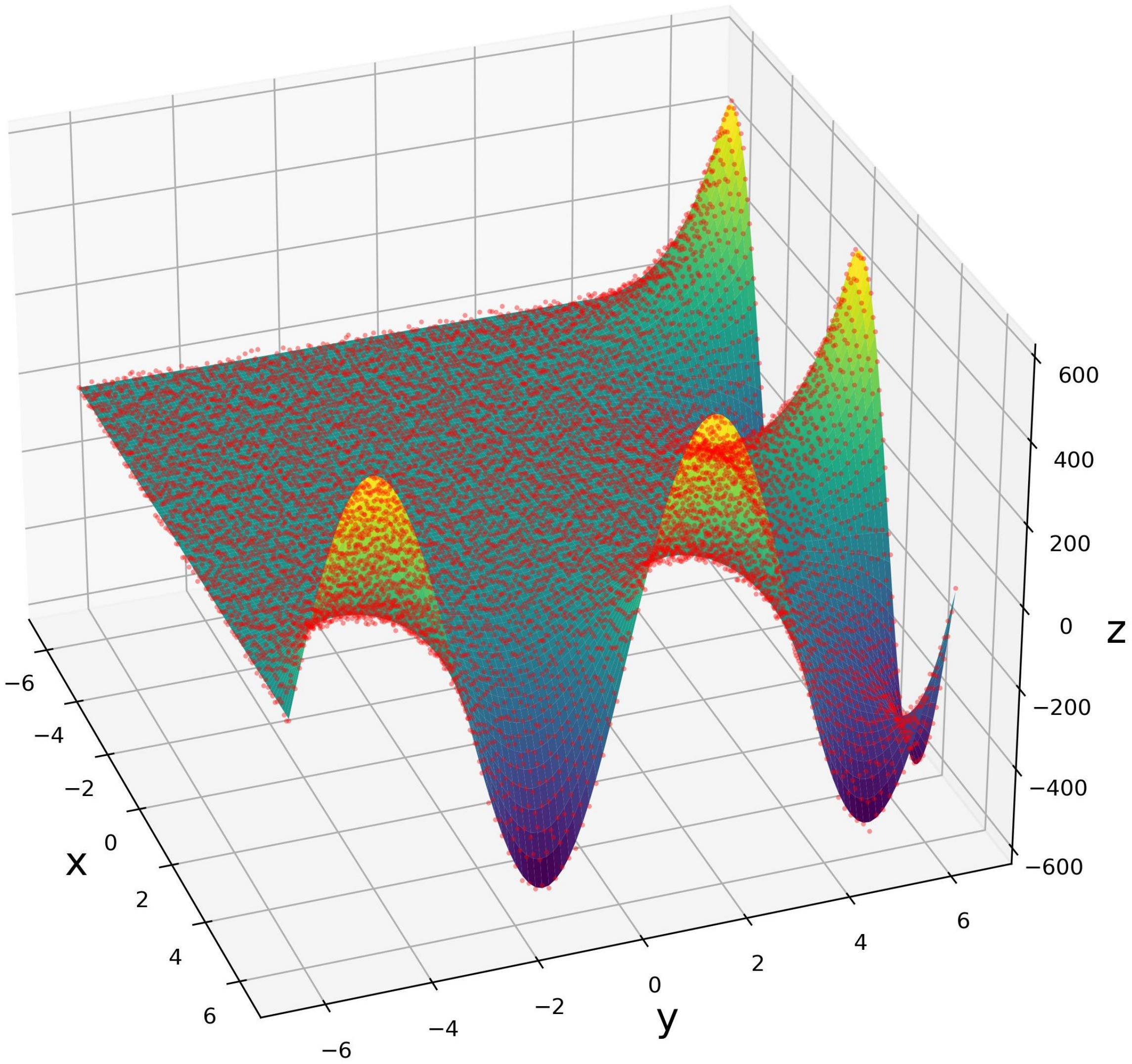

Figure 2: 3D surface plot of best fitted surface along with input data points