PCSC Project Report
Data Approximation

Haojun Zhu

I. Introduction

In this project, we implement numerical methods to solve data approximation problems. The data we are going to approximate is a sequence of 2 dimensional real-valued points (x_i, y_i) of length N. We assume the data is generated by an implicit intrinsic function $f : \mathbb{R} \to \mathbb{R}$.

By our implemented algorithms, we try to find an approximator function $\widetilde{f} : \mathbb{R} \to \mathbb{R}$ such that for every pair of (x_i, y_i), we have $\widetilde{f}(\mathrm{x}_i) = y_i$. And we try to reconstruct the intrinsic function f by this approximator.

II. How to compile

First, clone our project from our Github repository and open a terminal in the top directory of the project. In order to successfully compile our project, should type this following commands in the terminal:
mkdir build
cd build
cmake ..
make
If one has entered these commends, an executable $proj$ will be generated in the folder $build$.
Note that, the required minimum version of Cmake is 3.17.

III. The Flow

The class $UserInterface$ will display the textual information and read the keyboard input for the other classes which have different functionalities.
According to the keyboard input received earlier, the class $Data$ will read the data files and parse them into two vectors. This class can also be used separately to generate data files and that's how we prepared the four pre-defined data sample files in this project.
The class $Approximator$ will receive the data vectors read by the class $Data$ and use the keyboard input which defines the details of the method that is going to be applied to generate the approximation data. The number of the data points in the approximation data is 10 times the number of data points in the original data. For the specific approximation methods which require some linear algebra solution, some function members of the class $Matrix$ will be called.

IV. How to use

When compiling, linking and generating is done, the program is executive. User can choose two different ways to use this program. Each of them has some standards for the input that the user may need to respect.

A. Using the new data

One can choose to using their own new data to approximate. In this case, the user should choose the mode 1 when the user interface displays the option of working modes of the execution. The user also will be asked to give the valid path to their data file and the path of the output approximation data file. Note that, the data file should be put into the folder $data$ under the top directory of the project. In order to make the program process successfully, one should respect the form of the keyboard input of the data path. More specifically, the user should enter $../data/NameOfTheDataFile.dat$ for the path of the data file and enter $../data/NameOfTheOutput.dat$ for the output approximation data file path.
After that, user should specify the exact type of the approximation they want to use from the 3 implemented options: polynomial approximation, piecewise linear polynomial interpolation and piecewise cubic polynomial interpolation. Note that, if the user chooses to use the polynomial approximation method, should mention the degree of polynomials and the user interface will display the maximum degree can be optimized. If the user chooses the maximum degreem the normal polynomial approximation method will be applied; otherwise, the least squares method for data fitting will be applied.

B. Testing the prepared test cases

The uses can also choose to see the result of the test cases of our pre-defined data samples. In this project,

there are 4 data samples that are already present in the folder $data$ under the top directory of the project. User can directly use these 4 cases to see how this program works.
In this case, uses doesn't have to specify the path of the data file and the output approximation data file. The process of choosing the approximation methods is the same as in the case when user chooses to use their own new data file.

V. Algorithms and methods

There are 3 data approximation methods implemented in this project:
-Polynomial approximation
-Piecewise linear polynomial interpolation
-Piecewise cubic polynomial interpolation

VI. Test and result

As mentioned earlier, 4 data samples were already prepared. The values of x_i are the same 101 equally spaced points ranging from -20 to 20.
data sample 1: The values of y_i are generated from the function $f(x) = e^x$.
data sample 2: The values of y_i are generated from the function $f(x) = x^3 + x^2 + x + 1$.
data sample 3: The values of y_i are generated from the function $f(x) = 1/(1 + 25 \times x^2)$.
data sample 4: The values of y_i are generated from the function $f(x) = cos(|x|^{0.5}\pi)$.

These 4 cases are all tested and proved to be usable in this project. In order to show the performance of our algorithm, we display the approximation data of the data sample 4 below.
We first apply the polynomial approximation and choose 100 as the degree of polynomials, which is also the maximum degree according to the data sample. The comparison between the original data and the approximation is show below in the Figure 1.

Then we still apply the polynomial approximation methods but choose 95 as the degree of polynomial, the least squared method for data fitting will be applied this time. The result is shown in the Figure 2.

We also test the piecewise linear polynomial interpolation and the piecewise cubic polynomial interpolation and the results are shown respectively in Figure 3 and Figure4.

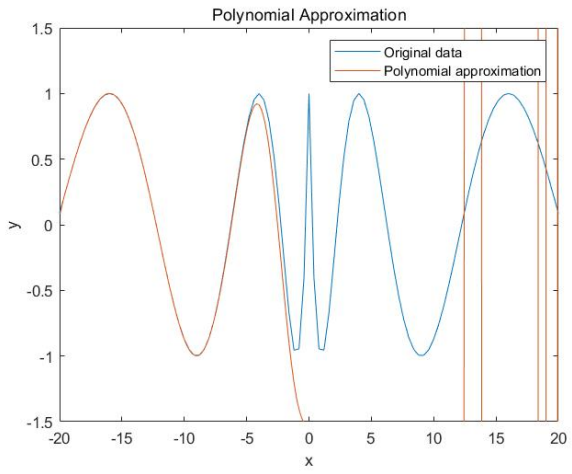

Fig. 1: Comparison between the original data and the polynomial approximation for the data sample 4

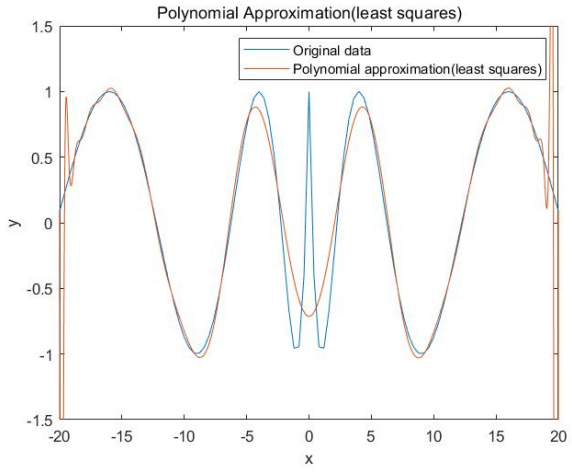

Fig. 2: Comparison between the original data and the least squares approximation for the data sample 4

VII. TODOs and perspective

A. Other approximation methods

In this project, we only implement 3 basic approximation algorithms. There are still many other different approximation methods which can give better results under some particular circumstances, for example, the Fourier approximation for periodic data.

B. A more delicate user interface

As mentioned, we have designed a user interface in this project, which is still somehow simple and crude. In the future, we may be able to extend its funcitonalities to make it more general for the users.

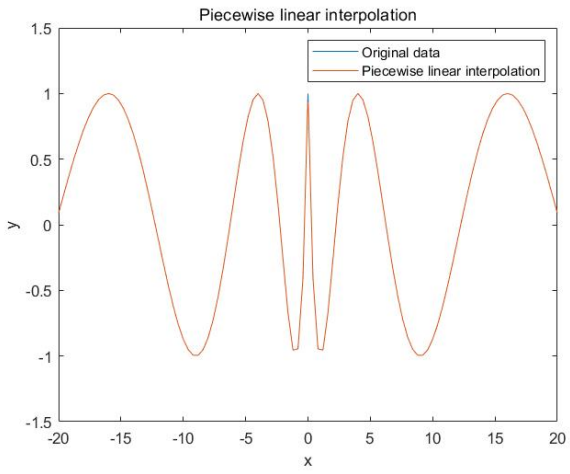

Fig. 3: Comparison between the original data and the piecewise linear polynomial interpolation for the data sample 4

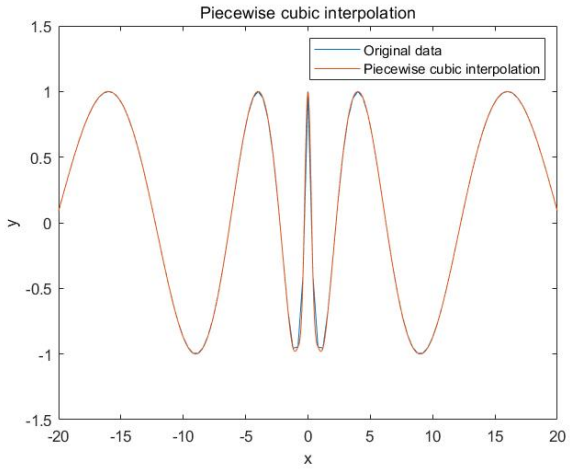

Fig. 4: Comparison between the original data and the piecewise cubic polynomial interpolation for the data sample 4

C. Graphic representation

We didn't implement an automatic method to compare the original data and the approximation data once the computation of the approximation is done by the machine. To see the comparison between the two, we still need to use Matlab to get the graphic comparison manually. In the future, we can add new classes to enable the automatic production of the comparison graphs.

D. A metric for the comparison

To evaluate the quality of the approximation, we should find some appropriate numerical metric that can be applied to the project. So far, we can only have an intuitive evaluation by the human eyes.