Task 4 – Interpolation & Curve Fitting

Introduction: Curve fitting and interpolation are among the most usable tool in signal processing and data science. In curve fitting, the given points can be treated as one chunk that fits one polynomial, or small chunks, each to be fit into a different polynomial. The latter is often called spline interpolation. In any interpolation problem, a compromise is often needed between the order of the polynomial (usually, the higher order, the better accuracy) and the computational needs (usually, the higher order, the more exhaustive).

Description: Develop an application that illustrates the efficacy of different curve fitting and interpolation models where:

- The user can open and display an arbitrary signal (of reasonable length of 1000 points).
- The user can choose if the fitting to be done as one chunk or multiple ones. If multiple, choose how many chunks the curve should be divided into. Propose the suitable UI controls to handle these options in a user-friendly way.
- The user can choose the order of the fitting polynomial (either for the big chunk or the multiple ones).
- Upon any change in the selections, the fitting result should display in dotted line on the same graph that displays the original signal.
- The fitted equation and percentage error should show in its mathematical form above the main graph using Latex format. You need to search on how to display Latex equation in your application.
- The user can generate an error map for the fitting process via clicking a button. The map should be displayed somewhere in the same window (i.e., not as a separate window). The user can choose the x- and y- axis of the error map among number of chunks, order of the fitting polynomial, and overlapping between consecutive chunks (0-25% of the chunk size). The error values in the map should be normalized and show up in percentage. Note that the user should have the flexibility to put any controlling factor on any axis of the map.
- Generating the error map might take some time. So, this process should be moved to a lower-priority thread. When the user clicks a button to initiate the process, a progress bar should show the status of the generation process, and the clicked button should turn into a "cancel" button where the user can click it to stop the process and return to the normal status.
- To illustrate the extrapolation efficacy, there should be an option to clip the curve fitting process into only portion of the open signal. For example, apply the fitting process on only 50% or 60%,…or 90% of the signal. Then, the fitted curve should be plotted for the whole signal. i.e., the last portion is practically extrapolated. Propose your own UI elements that make this process easy to handle, convenient and visually obvious for the user.

Dataset: Same You need to obtain your own non-synthetic data/signals/curves/time-series. Each group should obtain at least three different signals, each should result in different optimized settings. Signals that have the same settings would be discarded. You can obtain your data from open-access medical signals, stock market data, national or international statistics data of any type, …etc. But you are the one who will pick them to be of different features and thus, their fitting is optimized at different settings.

Code practice:

- Same practices from Task 3 (i.e., proper variable names & No code repetition, logging) will continue with task 4. No new practices for this task.