EP 501 Homework 3: Least Squares and Interpolation

October 22, 2019

Instructions:

- Submit all source code and publish Matlab results in .pdf form via Canvas. Please zip all contents of your solution into single file and then submit in a single zip file.
- Discussing the assignment with others is fine, but you must not copy anyone's code.
- I must be able to run your code and produce all results by executing a single top-level Matlab script, e.g. `assignment1.m` or similar.
- You may use any of the example codes from our course repository: `https://github.com/mattzett/EP501/`.
- Do not copy verbatim any other codes (i.e. any source codes other than from our course repository). You may use other examples as a reference but you must write you own programs (except for those I give you).
- For demonstrating that your code is correct when you turn in the assignment, you must use the test problems given in the assignment text below.

Purpose of this assignment:

- Learn principles behind data fitting and polynomial approximation.
- Develop good coding and documentation practices, such that your programs are easily understood by others.
- Hone skills of developing, debugging, and testing your own software
- Learn how to build programs on top of existing codes

1. Least squares and data fitting: this problem requires use of the example dataset from the repository which provides data for x_i, y_i, σ_i.

   (a) Write a program that performs a linear least squares fit of a set of data to a polynomial of arbitrary order n.

   (b) Use this program to fit the data to a line and a quadratic form. Plot your results and the data on the same axis so they can be easily compared. Which fit looks better (linear or quadratic)?

   (c) A more rigorous way of deciding between preferred function forms in fits (e.g. linear, quadratic, cubic, quartic, etc.) is to define a *goodness-of-fit* statistic that quantifies how effective a particular form is a describing a given data set. The most commonly used goodness of fit statistic is the *reduced Chi-squared statistic* defined by:

   $$\chi^2_\nu = \frac{1}{\nu} \sum_i \frac{(y_i - f(x_i))^2}{\sigma_i^2}, \tag{1}$$

   where x_i are the points of the independent variable at which data are sampled, y_i are the data, $f(\cdot)$ is the function which is being fitted to the data (linear, quadratic, cubic, etc.), σ_i is the uncertainty of measurement y_i and ν is the number of degrees of freedom in your fit (the number of unknown quantities involved in the fit - the coefficients of whatever order polynomial is being fitted). Write a function that evaluates χ^2_ν for a polynomial fit that has been performed.

   (d) What order polynomial ($1 \leq n \leq 10$) best describes the test data?

2. Bilinear interpolation: this problem requires use of the grid and data from the repository.

   (a) Write a function that takes in a grid of points describing some independent variable (say x_i), and a point to which the data are to be interpolated x' and finds the index i into the array x_i such that: $x_i \leq x' \leq x_{i+1}$.

   (b) Use the function from part (a) to construct an additional function that works over a 2D grid x, y. I.e. given two grids x_i, y_j find the indices i, j such that: $x_i \leq x' \leq x_{i+1}$ and $y_j \leq y' \leq y_{j+1}$.

   (c) Use your results from parts a and b to create a bilinear interpolation function that takes in a sequence of data points $\{x'_k, y'_k\}$ to which data are being interpolated, a grid x_i, y_j, and a dataset f_{ij} that is defined over this grid and produces bilinearly interpolated values of f at the points $\{x'_k, y'_k\}$. Write your program so that the input points are simply a flat list and no necessarily a 2D grid of points.

   (d) Test your results against Matlab's bilinear interpolation function and show that you get the same result.