

PHYS 2030

Lab 5

October 27, 2014

Finish as many parts of the problem set as possible (and do not worry if you cannot finish all of them, since there is a lot to do).

Suggestions:

- Put lots of comments in your code. You might use some parts of the code for your next assignments (or projects), and one thing you do not want to do is to spend time on trying to recall after couple of months what does your code do and why does it do it in that particular way. In addition, commenting will help you track down any logical errors, which are very hard to find in general.
- Use meaningful names for your variables (even if they turn out to get somewhat long). The reason for this is the same as above.
- Do not just comment the particular lines of your code, but you can break your code into logical sections and give clear explanation of what is this section for.

You do not have to hand in anything and this problem set is not going to be graded.

1. Least-squares method is based on minimizing some quantity, that one chooses to reasonably well quantify the quality of the fit.

If a set of data, $\{x_i, y_i\}$ is given, such that $\{\sigma_i\}$ is the corresponding set of uncertainties, obtained from normally distributed data, then we define the quality of the fit in the following way

$$\chi^2 = \sum \frac{(f(x_i) - y_i)^2}{\sigma_i^2},$$

where $f(x)$ is the fit function. In general χ^2 is a function of fit parameters, $\{C_i\}$, ... etc that we want to find and the problem of finding these parameters can be written as,

$$\chi^2(\{C_i\}) = \sum \frac{(f(x_i, \{C_i\}) - y_i)^2}{\sigma_i^2}$$

This needs to be minimized with respect to each parameter:

$$\frac{d\chi^2(\{C_i\})}{dC_j} = 0, j = 1, \dots, M$$

In general this gives a set of non-linear equations of the following form

$$\sum \frac{(f(x_i, \{C_i\}) - y_i)}{\sigma_i^2} \frac{\partial f(x_i, \{C_i\})}{\partial C_j} = 0, j = 1, \dots, M$$

In principle one can bring these equation into matrix form and solve it for unknown parameters with Matlab backslash command.

- Construct the matrix equation described above for

$$f(x, C_1, C_2) = C_1 x + C_2$$

- Generate noisy set of data with linear relationship. Your $\{y_i\}$ data should have randomly generated set of uncertainties.
- Find the parameters that will minimize Chi-Squared for your data. For this you can simply use Matlab backslash command, which will spit out the parameters for you.¹
- Now use Matlab built-in function `polyfit` (read **Help** file for it) and find the fit parameters with it.
- Plot the data and the fit, and calculated Chi-Squared. How good is your fit?
- Now construct matrix equation described above for

$$f(x, C_1, C_2, C_3) = C_1 x^2 + C_2 x + C_3,$$

which is second-order polynomial. This should not be hard, especially if you understood how to find the parameters for the linear case (which is first-order polynomial).

- As before, construct noisy second-order polynomial data with random uncertainties for $\{y_i\}$ data.
 - Find the fit parameters
 - Plot your data, fit function and fit constructed from `polyfit`. Calculate Chi-Squared.
2. Sometimes fit function $f(x, \{C_i\})$ is nonlinear. In this case it is much harder (or impossible) to find the parameters that will minimize Chi-Squared. Below we will look at the nonlinear function whose parameters can be found in several ways.

¹You have already done this in your previous assignment, but now you will not use explicit formula for the parameters, but let MatLab to handle this simple, but tedious task.

- Consider $f(x, A, \lambda) = Ae^{\lambda x}$. We can take natural log of both sides and obtain

$$\ln(f(x, A, \lambda)) = \ln(A) + \lambda x \quad (1)$$

- Generate noisy data with exponential behavior. Find fit parameters for equation 1. From these parameters obtain $\{A, \lambda\}$. (Note that the generated data has to be converted into 'log scale' as well.)
- Now use 'brute-force' grid search method (as was described during the lectures) to find the parameters.
- Use `fminsearch` to find the parameters.
- Plot your data and fit functions obtained by methods outlined above.