PHYS 250 Homework 09

Due: 14 April 2025

Problem 1. (10 points) Write a python routine that performs a least squares fit to the line

$$y = a_1 + a_2 x.$$

Your function should be defined so that it optionally accepts uncertainties; if not provided, the uncertainties are assumed to be one. To get you started, the function should be of the following form.

```
def linear_leastsq (x, y, sigma=None) :
    if sigma is None :
        sigma = np.ones_like(y) # This sets sigma=1 for each y
    ... # fill in the rest!
```

Your function should return a_1, a_2, σ_{a_1}, and σ_{a_2}. Include the code as your solution. **No loops are required in this function.** It is much easier to write the code without loops. You will use this code in the following problem. You can test it by comparing the output to that generated from `scipy.optimize.curve_fit`. (This will also show why `absolute_sigma=True` is required when using `curve_fit`.)

Problem 2. (10 points) Consider the data

x	-2	-1	0	1	2
y	-4	-7	-3	-3	-2

We will fit this data to a line using our routine from the previous problem.

(i) Fit the data to a line using your function from the previous problem. Determine and print the best fit parameters a_1 and a_2, along with their uncertainties, the χ^2 and the reduced chi-squared, χ^2_ν.

(ii) One of the points will be far from the line you fit in the previous part. Which point is the outlier?

(iii) Remove the outlier from the data set and then repeat part (i). [*Note*: In practice we *cannot* just throw away data because it does not fit well. We would need some other reason, unrelated to the model we are trying to fit, to reject the data. Without a compelling reason, the data must be used as provided and without modification!]

Problem 3. (10 points) In the previous homework we encountered the Hilbert matrix and saw that it is ill-conditioned. This is not just a matrix invented by a mathematician to create problems[†] but instead can appear in a minimization problem. Suppose we are given a known function $g(x)$ and wish to expand it in a finite power series so that

$$g(x) \approx \sum_{i=0}^{n} a_i x^i.$$

To find the coefficients, a_i, we could minimize a "χ^2-like" quantity we define as

$$X^2 \equiv \int_0^1 \left[g(x) - \sum_{i=0}^{n} a_i x^i \right]^2 \mathrm{d}x.$$

[†] Not to say that a mathematician would not create such a matrix for just such a purpose.

Notice that if the integral were replaced by a sum over a finite number of points this would just be the χ^2 with equal weight for each point. When we minimize X^2 with respect to the coefficients a_i we end up with a system of linear equations that can be written in the familiar form

$$\mathsf{A}\boldsymbol{a} = \boldsymbol{b},$$

where now $\boldsymbol{a}$ is a vector with components given by the coefficients a_i. This system of linear equations can then be solved.

(i) Perform the minimization and find the expression for the components of $\boldsymbol{b}$. These will depend on $g(x)$, but, when this method is applied, a particular functional form for $g(x)$ would be given and actual numerical values for the components of $\boldsymbol{b}$ would be known.

(ii) Again from the minimization determine the components of the matrix A. You should find that the A_{ij} are precisely the components of the Hilbert matrix. [*Note*: It can be useful to consider a small n case, such as $n = 2$, to more directly see the structure of the matrix. The results can be generalized to arbitrary n from there.]