Monte Carlo Estimation of Parameter Uncertainty

28 May 2020

PHY2004W KDSMIL001

1 Introduction and Aim

In this activity, we investigate methods of estimating uncertainties of parameters found when fitting a curve to positional data of a damped oscillator.

2 Activity

We chose to implement the Jackknife method of estimating uncertainties. This involves taking our original data set, DampedData.txt, removing one value, and then using `scipy.optimize.curve_fit` to fit a curve to those points. We then have a set of optimal fitting parameters. If we do this N times, where N is the number of values in the set, then we'll have N sets of optimal fitting parameters. We can then do some analysis to find a value and uncertainty for those parameters. The mean $\bar{x}$ is found in the usual manner, but the variance is found by

$$\sigma_x^2 = \frac{N-1}{N}\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \tag{1}$$

We can then find the standard uncertainty with

$$u(\bar{x}) = \sqrt{\frac{\sigma_x^2}{N-1}} \tag{2}$$

Where Monte Carlo methods come in is in choosing which value to remove each time. Usually the jackknife method requires you to remove sequential points, as in the first modified set is missing the first value, the second set is missing the second value, and so on. In our case, we will randomly choose a number between 1 and N, inclusive, and remove the value at that position in the array. We were sceptical that this approach is any better than doing it the usual way so we did both. Below are the results.

	Monte Carlo Method	Standard Method
A:	$0.283379291 \pm 4.09 \times 10^{-6}$	$0.283379121 \pm 4.1 \times 10^{-6}$
B:	$0.0281486 \pm 2.53 \times 10^{-5}$	$0.0281489 \pm 2.34 \times 10^{-5}$
γ:	$0.28119 \pm 3.55 \times 10^{-3}$	$0.28121 \pm 3.4 \times 10^{-3}$
ω:	$21.462285 \pm 3.08 \times 10^{-4}$	$21.462277 \pm 2.91 \times 10^{-4}$
α:	$0.329399 \pm 6.88 \times 10^{-4}$	$0.329386 \pm 6.36 \times 10^{-4}$

It seems that the uncertainties obtained using the standard method are actually smaller than those obtained using Monte Carlo while the values obtained definitely agree with each other.
Comparing this to the results obtained using the covariance matrix from `curve_fit` in CP2, which were

$u(A)$	$u(B)$	$u(\gamma)$	$u(\omega)$	$u(\alpha)$
6.33×10^{-5}	2.43×10^{-4}	4.54×10^{-3}	4.59×10^{-3}	8.73×10^{-3}

which shows that this method, Monte Carlo or not, gives us uncertainties of an order of magnitude smaller than using the covariance matrix from the fitting program.