PHY340 Data Analysis Feedback:

Group P04 doing Problem P5

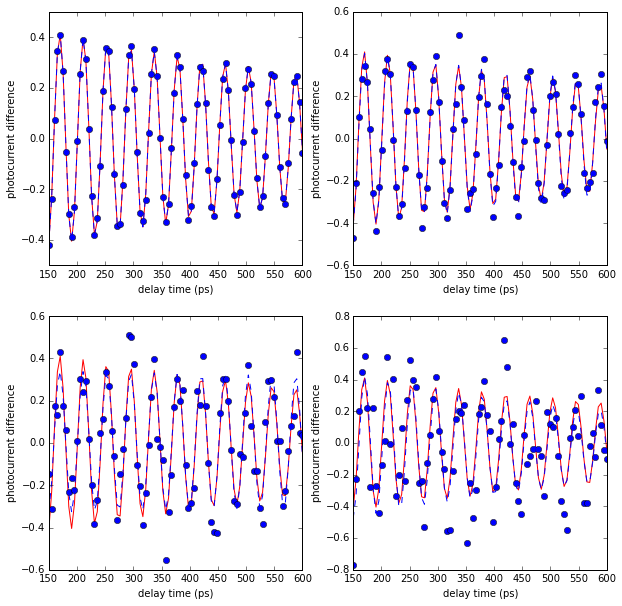
# Data Analysis

Your data analysis seems to be broadly correct and your fitted parameters seem reasonable. The main problem is the lack of meaningful uncertainties. Your statement that “there is no individual error associated with these values” is simply untrue: curve\_fit returns a full covariance matrix pcov, and the individual errors on parameters are extracted from this by writing perr = np.sqrt(np.diag(pcov)). There is no excuse for not knowing this: it is clearly documented in the SciPy reference manual[[1]](#footnote-1), which states “The diagonals [of pcov] provide the variance of the parameter estimate. To compute one standard deviation errors on the parameters use perr = np.sqrt(np.diag(pcov))” and examples were given in the lectures (see the fits2 and hypothesis\_test notebooks).

Instead of using the errors provided by curve\_fit, you seem to have done a linear regression compare­son between the data and the fit. This returns a correlation coefficient and something described as “the standard error on the estimate”. The SciPy manual is not at all helpful about what these actually are, but there is a simple example at <http://davidmlane.com/hyperstat/A134205.html>: it’s actually the RMS residual,   
where *yi* is the *i*th measurement, *y*fit is the corresponding value produced by the fit, and *N* is the total number of measurements. **This tells you nothing at all about the errors on the parameters**: unless your function is the wrong shape, it is probably telling you about the errors on the measurements.

The statement that the need for good initial values of the parameters “was due to the programme having to find the values of five unknown parameters at once using a single fit function” is complete nonsense, as you should have known from the lectures. Provided that your fit function is linear *in the parameters* (**not** in the variables: is an example of a function with five unknown parameters that is linear in the parameters), a least-squares fit should converge to the global minimum regardless of the initial parameter values used, because the set of simultaneous equa­tions that the fitting routine has to solve are all linear. If the fit function is *not* linear in its parameters, e.g. *y* = *axb* + *c*, then convergence is *not* guaranteed and you may have to be fairly precise in your initial guess. The function you are trying to fit, , is very non-linear in its para­me­ters, so it is not at all surprising that it needs a good initial guess to converge.

The assertion that “all errors in the *g*-factor were due to the devised fitting method” is also complete nonsense. To demonstrate this, I simulated data using the above equation, added Gaussian random errors to the calculated *y*-values, and fitted the resulting fake data using curve\_fit. The results are shown in figure 1 below. Clearly (and as expected) the errors on the fit parameters (apart from *τ*, which is badly affected by correlations, see below) scale with the er­rors on the input data: your claim that they do not is totally false. Note that the RMS residual, or “standard error of the estimate”, reflects the error on the data, not the errors on the fit parameters. Also note that the parameter you are in­terested in, *ω*, is *by far* the best constrained parameter in the fit: even for the noisiest data its relative uncertainty is only 0.5%. In contrast, *A* and *τ* are poorly constrained, because they are highly anti­correlated: because the fit does not start at *t* = 0, a small value of *A* can be compensated for by a larger value of *τ*, and vice versa. This can be clearly seen in the fits below, especially the third one. Crucially, it does not seem to affect the accuracy of *ω*.

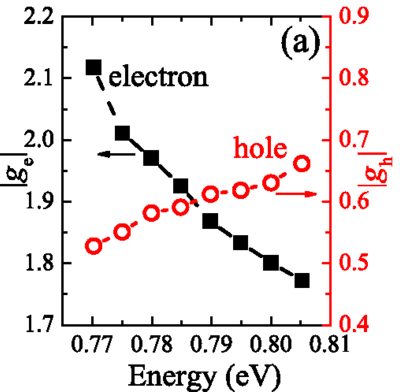


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Error | *A* | *τ* | *ω* | *ϕ* | *C* | *RMS* |
| (truth) | 0.500 | 900 | 0.1500 | 1.300 | 0.0 |  |
| 0.01 | 0.502±0.007 | 882±29 | 0.15000±0.00004 | 1.297±0.013 | 0.0014±0.0011 | 0.011 |
| 0.05 | 0.478±0.030 | 1080±199 | 0.15015±0.00017 | 1.245±0.062 | −0.0029±0.0055 | 0.050 |
| 0.10 | 0.346±0.050 | 5530±11451 | 0.15018±0.00037 | 1.246±0.146 | 0.0067±0.0112 | 0.107 |
| 0.20 | 0.570±0.138 | 657±303 | 0.15044±0.00070 | 1.043±0.246 | −0.0212±0.0210 | 0.209 |

*Figure 1: Fits to simulated data with different amounts of noise: upper left, ±0.01; upper right, ±0.05; lower left, ±0.10; lower right, ±0.20. The red curve represents the true values, and the blue dashed curve is the fit. The fitted parameters and RMS are shown in the table.*

It should have been obvious to you from your figure 6 that your estimate of the uncertainty on *ω* was wildly wrong: error bars should reflect the scatter of the points about the true line, so the fact that all four of your points lie dead on the line, despite their alleged huge errors, is a clear pointer that some­thing is badly wrong. Your procedure for fitting your data is, of course, nonsense: the way to fit points with errors is to input them into a suitable fitting program. For a simple straight line fit, curve\_fit might be considered overkill, but it does work: with your (silly) errors, it gives a gradient of 0.050± 0.019 THz T−1, which corresponds to a *g*-factor of 0.57±0.22. To demonstrate just how silly your er­rors really are, if we allow curve\_fit to scale them according to the real scatter, the gradient comes out as 0.05035±0.00008 (and the *g*-factor as 0.5725±0.0009).

Having calculated your wildly overestimated error, you then proceed to ignore it! This is the only ex­planation for your statement that your *g*-factor “is 6 errors away from the accepted value 0.39±0.03.” Clearly 0.57±0.32 is perfectly compatible with 0.39±0.03: the difference is 0.18±0.32, which is self-evidently consistent with zero. (With correctly calculated errors, you *would* have an issue with the “accepted value” that you quote; however, the *g*-factors of individual quantum dots are quite variable: Belykh et al. (2015) quote |*gh*| = 0.64, and show a dependence on energy (i.e. quantum dot size) which easily accommodates your 0.57. (I have no idea why you think that the difference between 0.64 and 0.57±0.32 is “much greater than a single error bar.”) In fact, Mark Fox told me that the QD whose data you were given is not a very typical specimen, so some discrepancy with the literature might be expec­ted—but, given your inflated error estimate, you have no evidence for any discrepancy with anything.



*Figure 2: g-factor of holes (red) and electrons (black) in InAs/InAlGaAs quantum dots, from VV Belykh et al., Phys. Rev.* ***B 92*** *(2015) 165307.*

Average mark for this section: 30.5/50

# Data Presentation

Your plots are generally well presented, although in the wrong order: your figure 2, showing a com­plete dataset and demonstrating the need to neglect the first 50 ps or so, belongs before your figure 1, where this cut has already been made. It is not obvious why figure 6 has the units of precession fre­quency as (ps)−1 and figure 7—which shows the same data—as THz: these are equivalent units, so pick one (probably THz is better) and stick to it.

Your consistently quote far too many significant figures in numerical results: this is partly because of your failure to make use of the uncertainties that curve\_fit provides, but that does not excuse quoting 4 significant figures in the RMS residuals, or “*g* = 0.573301±0.32367” on the next-to-last page. Your pre­sentation is also unnecessarily repetitive: the analysis is the same for all four magnetic fields, so des­cribe it *once*, make any comments about minor adjustments for individual datasets (e.g. not using the last 200 ps of the 4 T dataset because you didn’t like the look of it—which wasn’t necessary, by the way, as the fit handles it perfectly well), and present a table of fitted parameters similar to that in­cluded in figure 1 above.

Comparison of your results with the literature is good practice, but I have no idea why you claim that 0.39±0.03 is “the accepted value”: the Prechtel et al. paper you cite gives this as the ***electron*** (not hole) *g*-factor “in our sample”, and comment that the hole *g*-factor depends strongly on the applied electric field (they do not quote a value, but their figure 3 shows a smaller hole *g*-factor, around 0.2 to 0.3 for their QD1). They certainly do not claim that 0.39±0.03 is a generally applicable value, and there seems no reason to prefer it to Belykh et al’s 0.64 “with a spread of 0.05”.

Your discussions of your individual fits are very muddled. The statement that “the resolution of the data collected is reduced at higher magnetic field strengths” is completely unjustified and, as far as I can see, completely untrue. It *is* true that the number of data points per oscillation period is lower because the oscillation frequency is higher, but this says *nothing* about the resolution of the data. In fact, it is possible to fit data that are sampled at a frequency that is lower than the frequency you want to fit, although you may need to use more subtle techniques than curve\_fit to do so. It’s *definitely* not true that goodness of fit will be reduced “for any model” because of sparser sampling: clearly, if your data were free of noise, the goodness of fit would be perfect even if the sampling were very sparse, since it’s defined by the difference between the fitted points and the data. And I have noidea why you think the data at higher fields are “more difficult to plot accurately”—they are just pairs of (*x*, *y*) points; it doesn’t matter to the plotting program whether they are on a nice smooth curve or not. It is also not true, I don’t think, that the experimental data at 4 T are “more scattered”, even though I agree that they look as though they are: I found that the RMS residual was not significantly different for this dataset, but the signal was smaller (so the signal-to-noise ratio was worse). As for comments like “the peaks of the plotted data are close in time to the peaks of the modelled fit”—well, *of course* they are, that’s what a fit does! “[T]he peaks in this [2 T] set are much less clustered” (than the 1 T data)—I have *no* idea what this is supposed to mean: with the exception of the transient below 100 ps, the peaks in all the data sets are equally spaced—there *is* no “clus­tering”—and the 1 T peaks have wider spacing than the 2 T peaks. I am equally baffled by the asser­tion that the peaks in the 1 T data are “ambiguous”: there’s nothing ambiguous about them, they are as obvious as peaks get. This is all extremely confused and difficult to follow: *you* may know what you mean, but short of telepathy nobody else does.

Average mark for this section: 19.5/30

# Style

Your report has a reasonably sound overall structure, although the fitting of the individual datasets and the final linear fit of the *ω* against *B* plot probably deserved to have separate sections, rather than being subsumed under “Method”, and the comparison of your final result with literature values does not belong in the “Method” section at all. The English is generally not bad, apart from a tendency to put capital letters after every centred equation—that’s probably Word being “helpful”, but these are *not* new sentences and should *not* start with capitals, so if Word puts one in, take it away again.

The report doesn’t read well, because it’s too repetitive (see above) and because there are too many statements that are grammatically correct but don’t actually make any sense, such as those quoted above: that’s not really a stylistic issue; it’s a product of muddled thinking. You need to think about what you are trying to say and then how to say it so it is comprehensible to someone else!

Average mark for this section: 12.5/20

Overall average mark: 62.5%

1. <http://docs.scipy.org/doc/scipy-0.17.0/reference/generated/scipy.optimize.curve_fit.html> [↑](#footnote-ref-1)