

# Dissertation

Francesco Tamburi

October 2023

# Todo list

expand paragraph . . . . .	7
reference to justification, placed earlier . . . . .	15
possibly expand . . . . .	15

# Introduction and Overview

What is PALS? - Potential uses - Experimental setup - Intro to PALSfit - Positron lifetime (in some order)

Positron Annihilation Spectroscopy is an experimental technique for analyzing materials that uses positrons (duh).

Many types of analysis from positron experiments, lifetime analysis is particularly useful.

Source, usually radioactive source material, produces positrons. High energy positrons slow down in "thermalization" phase, mechanism differs slightly for different materials (e.g conductors vs semiconductors vs insulators). Thermalization is overall not significant from a lifetime standpoint [cite the study cited in Krause]. Once slowed, positron wavefunction diffuses through the material (something something Bloch state).

During diffusion, lattice structure affects lifetime due to electron density. Deviations from regular lattice structure affect lifetimes due to changes to electron density, e.g atomic vacancies (but also dislocations, bigger voids and precipitates). Longer lifetimes when defects present.

A little bit about PALSfit...?

## 0.1 Positron Annihilation Lifetime Spectroscopy

Positron annihilation as a means of investigating lattice imperfections traces its roots back to the late 1960s, when the link between positron lifetimes and lattice vacancies was first proposed. [?] As the lifetime of a positron is a function of the electron density at the annihilation site, and lattice defects (e.g. open vacancies) represent a local decrease in electron density, longer lifetimes are observed for positrons trapped in these defects. [?] The presence of multiple positron lifetimes within a sample material, thus, indicates the presence of lattice defects within that material.

Positron Annihilation Lifetime Spectroscopy (PALS) is the study of these lifetimes, with the aim of determining the number and types of defects present in a sample material. While other techniques to study lattice defects exist, the main advantage of PALS is the high sensitivity of the technique to open-volume defects.<sup>1</sup> This property, along with the non-destructive nature of the technique, has led to its application in various fields of study.[?] [?]

Positron lifetimes can be determined experimentally by measuring the time interval between the emission of gamma photons that signal the production and annihilation of the positron. Commonly used positron sources are weakly radioactive isotopes such as  $^{22}\text{Na}$ , which simultaneously emits a 1.27 MeV  $\gamma$ -photon and a positron as a part of its  $\beta^+$  decay. As the positron encounters the sample material, it quickly loses its kinetic energy in a process known as "thermalization" as it penetrates into the sample. The positron's wavefunction then diffuses into the material and finally annihilates, producing two 511 keV  $\gamma$ -photons.

A scintillator-photomultiplier setup converts the  $\gamma$ -rays into analog electrical pulses, which are then processed by discriminators, distinguishing the 1.27 MeV from the 511keV pulses. The former is used as a "start" signal to begin charging a capacitor, and the latter stops the charging process. The capacitor acts as a sort of "electronic stopwatch", converting the time delay into an amplitude signal. To ensure a linear time-amplitude relationship, the stop signal is delayed by a fixed amount. Each signal is stored in the memory of a multichannel analyzer, where the channel numbers represent time and every count represents a single annihilation event. The resulting lifetime spectrum contains more than  $10^6$  annihilation events.

## 0.2 The lifetime spectrum

The resulting spectrum contains  $k + 1$  lifetime components, corresponding to  $k$  different defect types and the positron lifetime in the defect-free bulk ( $\tau_b$ ). From these we get the time-dependent positron decay spectrum  $D(t)$  – the probability for a positron to still be alive at a given time  $t$  [?] – given by the expression

$$D(t) = \sum_{i=1}^{k+1} I_i \exp\left(-\frac{t}{\tau_i}\right), \quad (1)$$

where  $\tau_i$  and  $I_i$  represent the lifetime and intensity of each component, respectively.<sup>2</sup> The resulting positron lifetime spectrum  $N(t)$  is given by the absolute value of the time derivative of the positron decay spectrum  $D(t)$ <sup>3</sup>:

$$N(t) = \left| \frac{dD(t)}{dt} \right| = \sum_{i=1}^{k+1} \frac{I_i}{\tau_i} \exp\left(-\frac{t + t_0}{\tau_i}\right). \quad (2)$$

---

<sup>1</sup>Thorough comparisons of defect sensitive techniques, as well as their respective advantages and disadvantages, are available in literature, but detailed analysis of these is beyond the scope of this project.

<sup>2</sup>When  $k = 0$  (i.e. there are no defect components),  $\tau_i = \tau_b$  (the bulk lifetime) and  $I_i = 100\%$

<sup>3</sup>Due to the delay introduced in the time-amplitude conversion,  $t = t + t_0$

The experimental spectrum is the convolution of the lifetime spectrum analytically described above and an Instrument Resolution Function (IRF). This resolution function is mainly determined by the scintillator+photomultiplier and takes the form of a sum of several Gaussian functions,

$$F(t) = \sum G_i(t), \quad (3)$$

with variations in the peak position, relative height and width of each Gaussian,  $G_i(t)$ . This gives us an experimental spectrum of the form

$$D_{exp}(t) = \int_{-\infty}^{+\infty} D(t-t')F(t')dt'. \quad (4)$$

### 0.3 PALSfit

Analysis of experimental spectra to extract physically meaningful parameters can be done using various computer programs. Most software does this by fitting a model function to the experimental spectrum, using some sort of least-squares method. PALSfit is a software package developed for this purpose. Using the POSITRONFIT module, it can determine lifetimes and intensities from lifetime spectra.

PALSfit Version 3, or more simply PALSfit3, is the latest version of this package, which this project is aimed at evaluating. Any subsequent references to PALSfit refer to this version of the software.

# Chapter 1

## Reading the spectrum

### 1.1 The positron lifetime

The lifetime of a positron, from production to annihilation, tends to be quite short, usually measured in the picosecond to nanosecond time scale. Nevertheless, during this brief lifetime, we can identify a few key stages and different paths that the positron may take.

A significant fraction of the positrons produced (up to 10-15%) never penetrates the sample material, annihilating instead in the source or source-sample interface. This can occur, for example, due to the backscattering of the positrons as they encounter positively charged atomic nuclei.

Thermalization, the process that slows down positrons before they diffuse into the sample, occurs via different mechanisms, depending on both the type of material and the energy of the positron. For example, at high energies, atomic ionization is dominant, while at low energies ( $< 1$  eV in metals and  $\sim 1$  eV in semiconductors) scattering off phonons dominates. Regardless of the specific mechanisms, however, the thermalization process occurs within the first few picoseconds.

Once thermalization occurs, the positron is free to diffuse through the sample lattice. This is the longest stage and also the one where defect trapping predominantly occurs. During diffusion, the wavefunction of the positively charged positron is primarily concentrated in the interstitial space between atoms due to the repulsion of positively charged atomic nuclei. In a perfect lattice, the positron delocalizes into a Bloch state with  $k_+ = 0$ . [mention of band structure?]

### 1.2 Trapping

Trapping occurs when a vacancy-type defect is encountered and a localized positron state forms at the defect. The rate at which this occurs, called the positron trapping rate  $\kappa$  is proportional to the defect concentration  $C$ . Specifically

$$\kappa = \mu C, \tag{1.1}$$

where  $\mu$  is termed the positron trapping coefficient, and is constant for a given defect.

Prob of trapping depends on vacancy concentration. — > Single defect trapping model

# Results

## 1.3 First analysis

Using PALSSIM, a series of spectra containing two positron lifetimes,  $\tau_1$  and  $\tau_2$ , were generated with a three gaussian instrument resolution function (see Table 1.1 and Figure 1.1).  $\tau_1$  was kept fixed at 180ps and  $\tau_2$  varied from 220-280ps, in 10ps intervals. For each value of  $\tau_2$ , three spectra were generated with the relative intensities of  $\tau_1$  and  $\tau_2$  set to 20%-80%, 50%-50% and 80%-20%. All the resulting spectra were then analyzed using PALSFIT, in order to evaluate how well the program could extract the two lifetimes, and their respective intensities, from each simulated spectrum.

In Figure 1.2a, we can observe how the values of  $\tau_1$  outputted by the program change, as the simulated value of  $\tau_2$  (on the horizontal axis) increases and the relative intensities (indicated by color and shape of marker) vary. Zooming in to the  $\tau_2 = 250$ -270ps range, we can see in Figure 1.2b that, for these values in particular, PALSFIT struggles to determine  $\tau_1$  in the 50-50 and 80-20 case.

Unlike for  $\tau_1$ , where the simulated value of the lifetime is fixed, the simulated value of  $\tau_2$  (our point of comparison) changes in between spectra. To make the data easier to visualize, rather than the fitted value of  $\tau_2$ , the difference between the result and the original is plotted instead (see Figure 1.2c). In the figure we can see that, aside from the 20-80 spectrum for  $\tau_2 = 220$ , the software performs better than for  $\tau_1$ .

The error bars represent the standard deviation of – and thus the confidence of the program in – the lifetimes. From them, we see two factors that affect the size of the bars. The first is the relative intensities of the two intensities. In fact, in Figure 1.2a, which plots  $\tau_1$ , the error bars are the smallest for the 80-20 data, where the shorter lifetime is more intense, and in Figure 1.2c, tracking  $\tau_2$ , the opposite is the case and we have the 20-80 data is most precise. Observing all the figures, we see the second factor: as the time interval between the two lifetimes increases, the size of the error bars decrease.

In Figures 1.2d - 1.2f the fitted intensities of the two lifetimes are plotted against simulated  $\tau_2$ , for each combination of simulated intensities. In the figures, we see that when the relative intensity of  $\tau_2$  was greater or equal to  $\tau_1$ , then PALSFIT was able to calculate all the appropriate lifetime intensities. However, when the intensity of  $\tau_2$  was set to 80%, shown in Figure 1.2d,

Table 1.1: Instrument resolution function

FWHM (ps)	Shift (ps)	Intensity (%)
213.3	0	80
150	-5	10
267	17	10

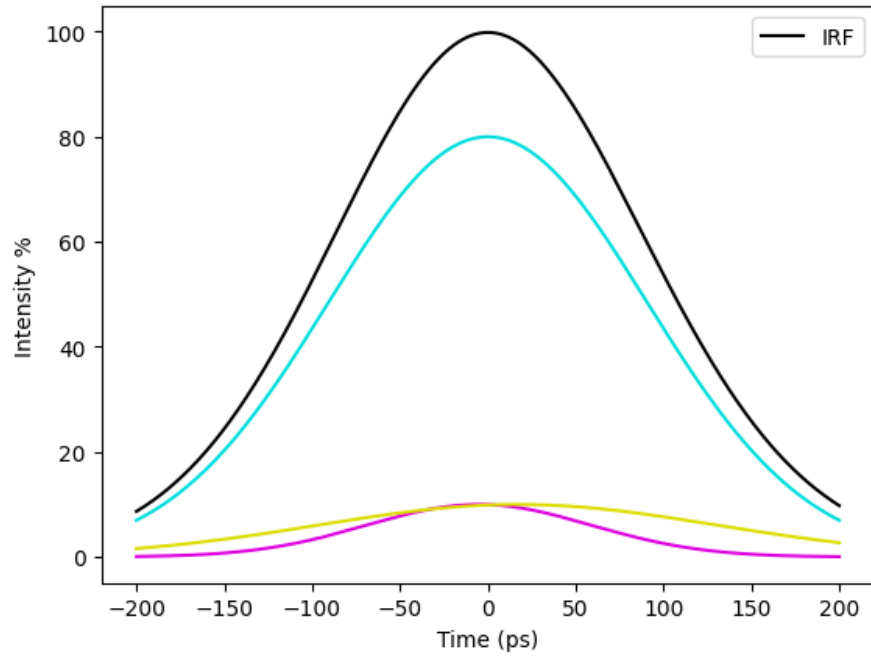


Figure 1.1: instrument resolution function

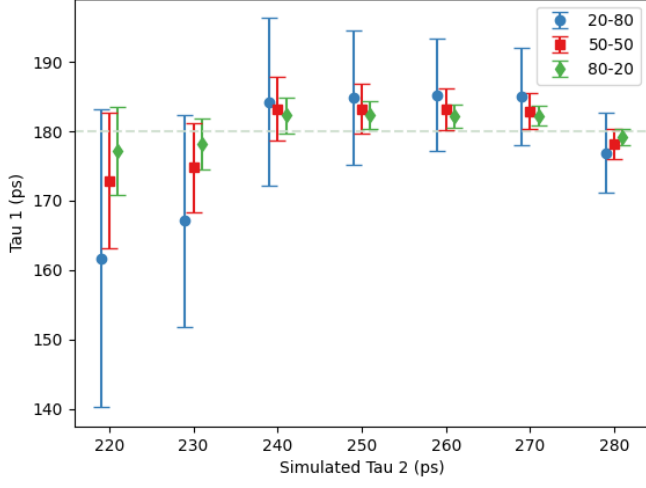
the software was unable to output the correct intensities for the first two simulated values of  $\tau_2$ . Looking at our error bars, we can see the same relationship between their size and the lifetime separation mentioned earlier.

On the whole, PALSFIT seems to have performed well, but not perfectly. As the first lifetime,  $\tau_1$ , was kept constant at 180ps, the next step would be to change  $\tau_1$  and see how that affects our results.

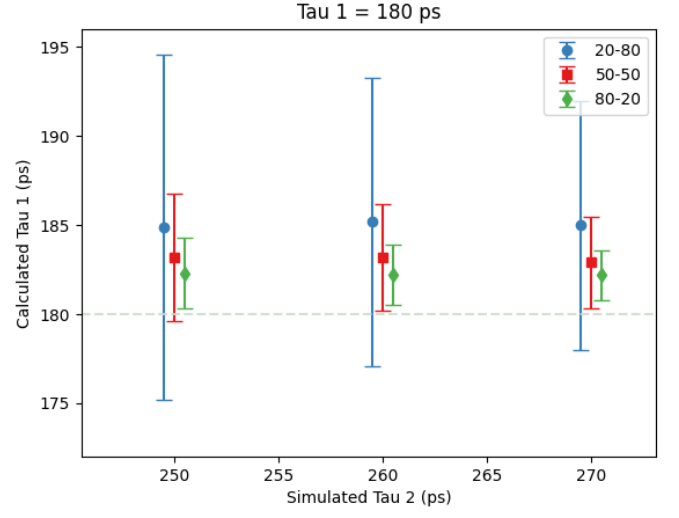
expand  
paragraph



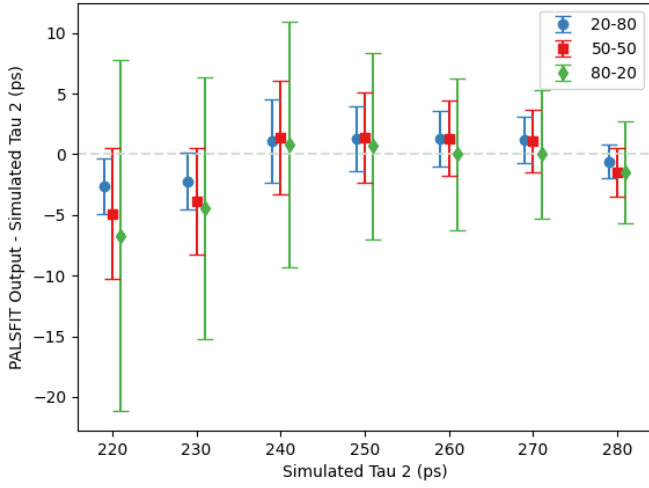
Figure 1.2



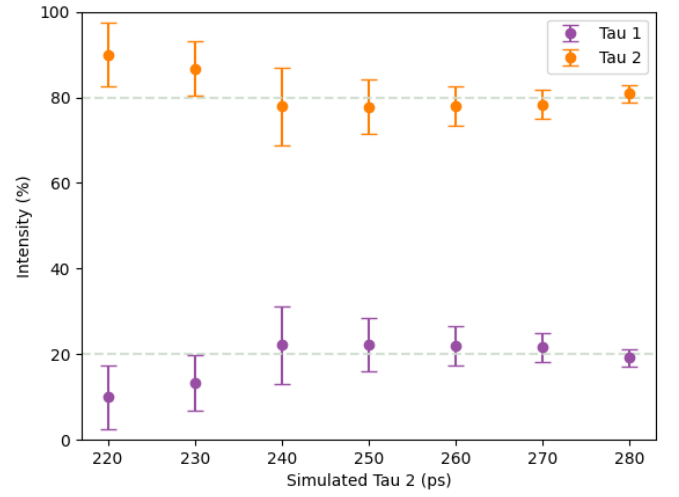
(a) fixed  $\tau_1 = 180ps$



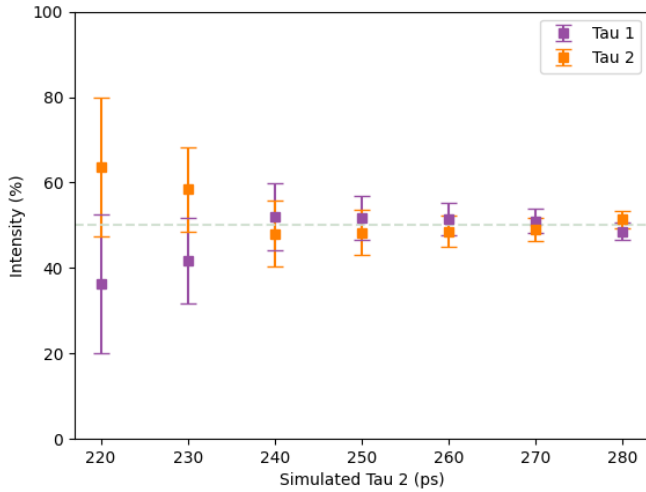
(b) close-up  $\tau_1 = 180ps$



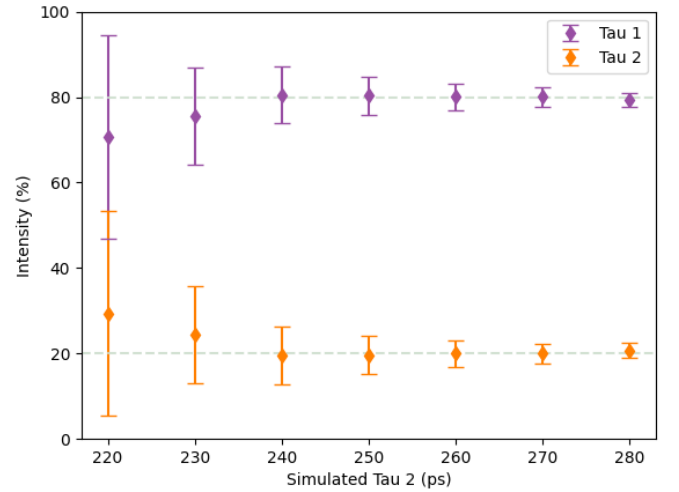
(c)  $\tau_2$  difference



(d)  $\tau_1 = 20\%, \tau_2 = 80\%$



(e)  $\tau_1 = 50\%, \tau_2 = 50\%$



(f)  $\tau_1 = 80\%, \tau_2 = 20\%$

## 1.4 Modifying $\tau_1$

A similar procedure was performed for  $\tau_1=150$  and  $\tau_1=220$ . To keep the relative time interval the same in between batches, the corresponding spacing between  $\tau_1$  and the  $\tau_2$  range was kept consistent. For  $\tau_1 = 150$ , this meant a  $\tau_2$  range of 190-250ps, and for  $\tau_1 = 220$ , this meant a corresponding  $\tau_2$  range of 260-320ps.

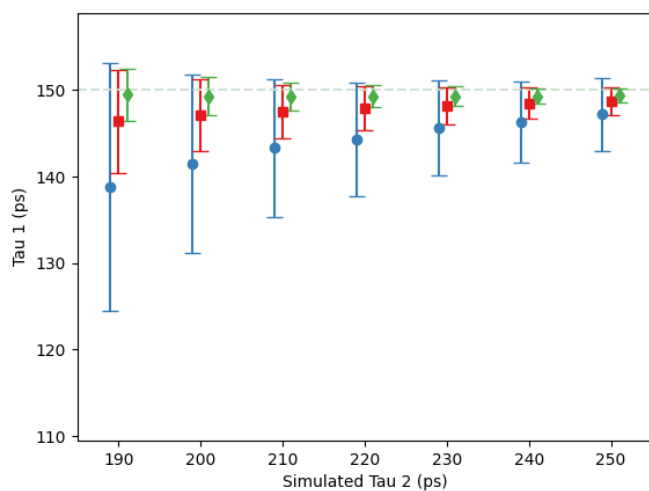
This was first done for  $\tau_1 = 150$ . As can be seen in Figure 1.3, the results for this batch are all within error, with both accuracy and precision getting better as the lifetime separation increases, in line with what would be expected.

The other batch, meanwhile, where  $\tau_1$  was set to 220ps, was not as successful. The results can be seen in Figure 1.4, but in general, PALSFIT struggled with fitting the lowest values for  $\tau_2$  and the error bars are noticeably larger. The program even gives nonsensical results when  $\tau_1 = 260$ ps and the relative  $\tau_1$ - $\tau_2$  intensity is set to 80%-20%, as can be seen in Figures 1.4a, 1.4b and 1.4e.

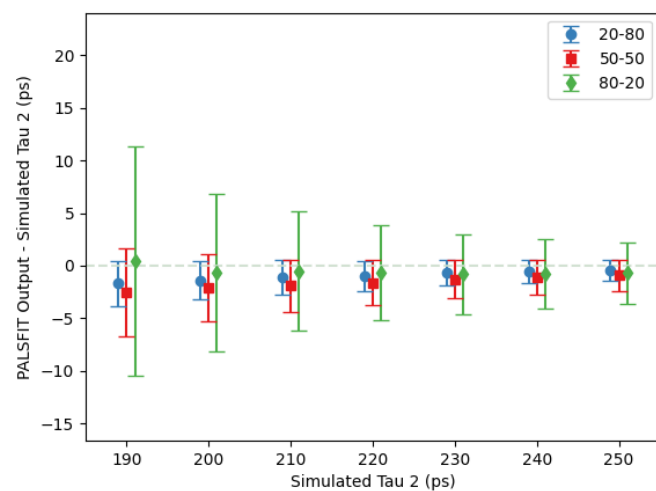
In order to compare all three datasets, the absolute difference between the fitted and simulated values of each relevant variable,  $\tau_1$ ,  $\tau_2$  and intensity was plotted. Additionally, plots for their respective standard deviations were generated. The resulting plots are represented in Figures 1.5-1.7. Two general observations are apparent from analysis of these figures:

The first is that the  $\tau_2 = 180$ ps data seems much more scattered than the other two datasets, which both follow a relatively clear increase in precision and accuracy as the lifetime separation increases. The second is that as  $\tau_1$  decreases, PALSFIT seems better able to resolve the two lifetimes, with both the difference from the true value and the size of the error bars being the smallest when  $\tau_1 = 150$ ps.

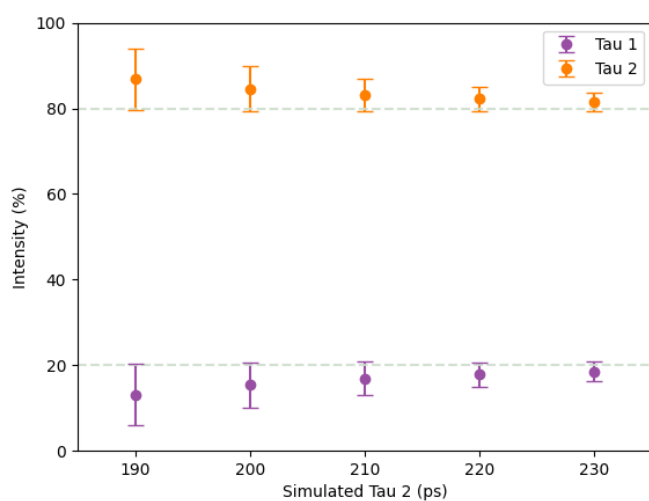
Figure 1.3



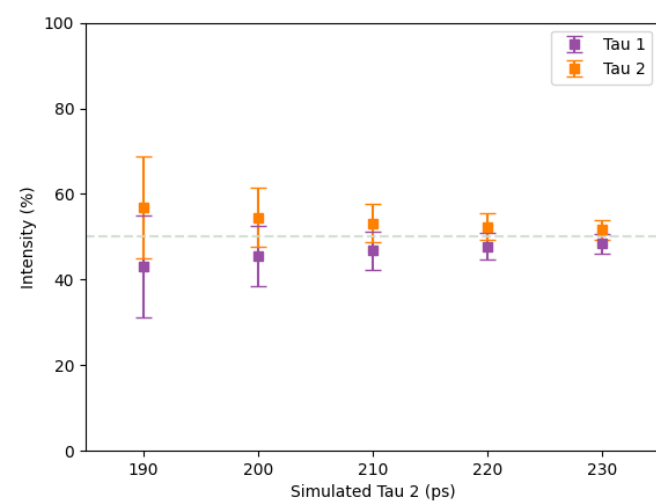
(a) fixed  $\tau_1 = 150$  ps



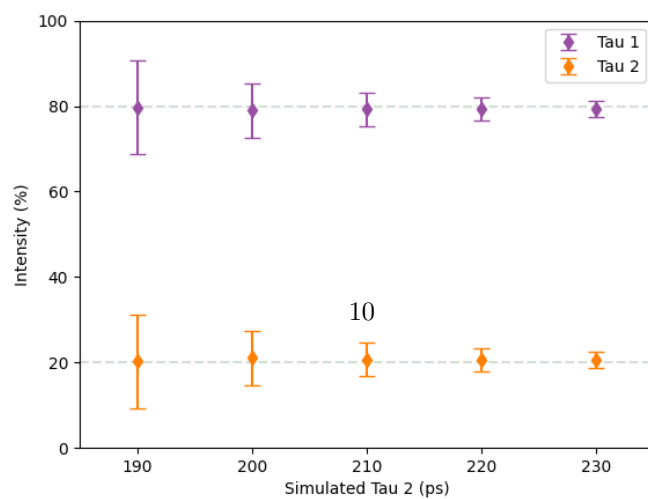
(b)  $\tau_2$  difference



(c)  $\tau_1 = 20\%$ ,  $\tau_2 = 80\%$

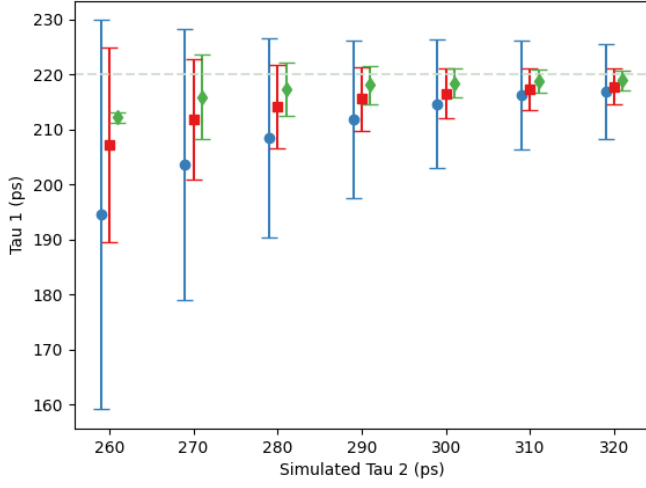


(d)  $\tau_1 = 50\%$ ,  $\tau_2 = 50\%$

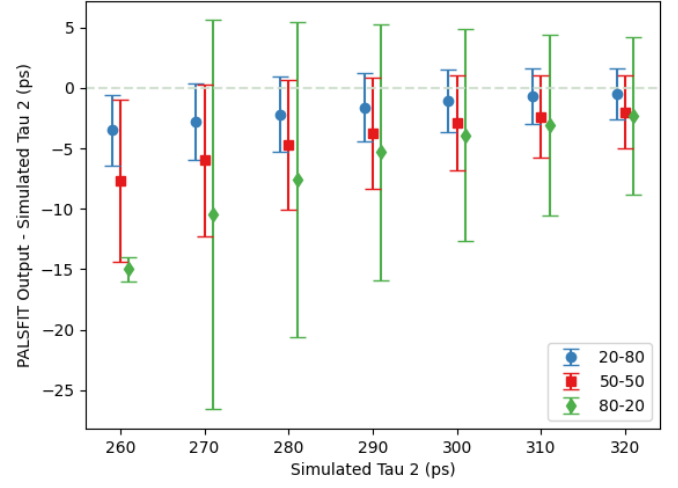


(e)  $\tau_1 = 80\%$ ,  $\tau_2 = 20\%$

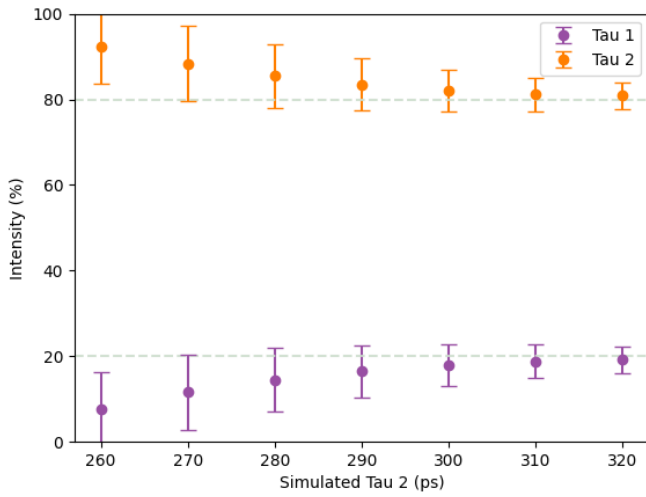
Figure 1.4



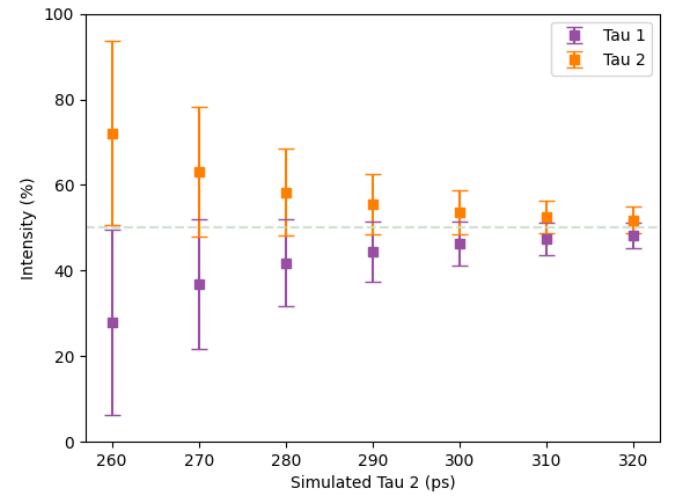
(a) fixed  $\tau_1 = 220ps$



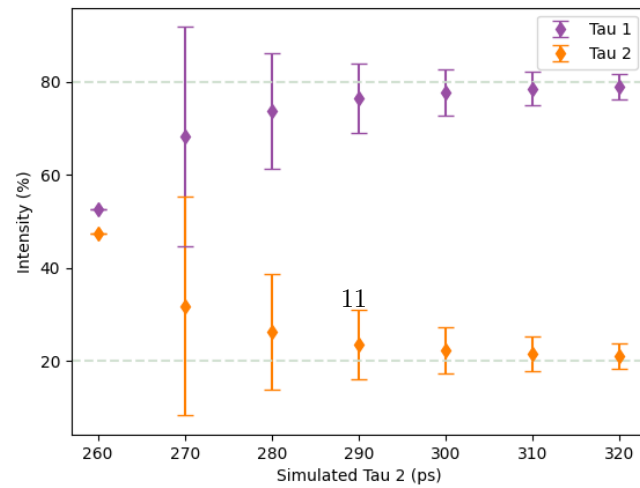
(b)  $\tau_2$  difference



(c)  $\tau_1 = 20\%, \tau_2 = 80\%$

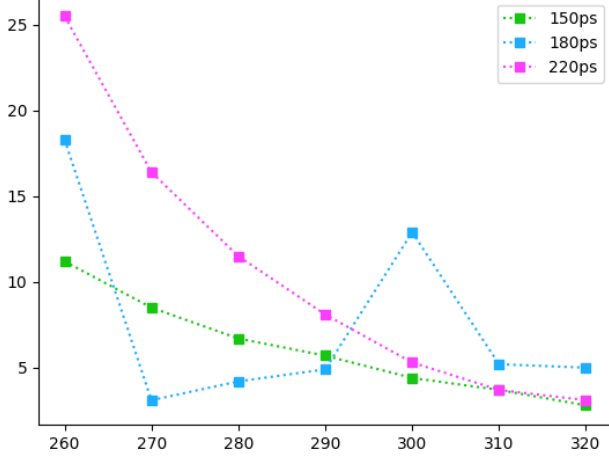


(d)  $\tau_1 = 50\%, \tau_2 = 50\%$

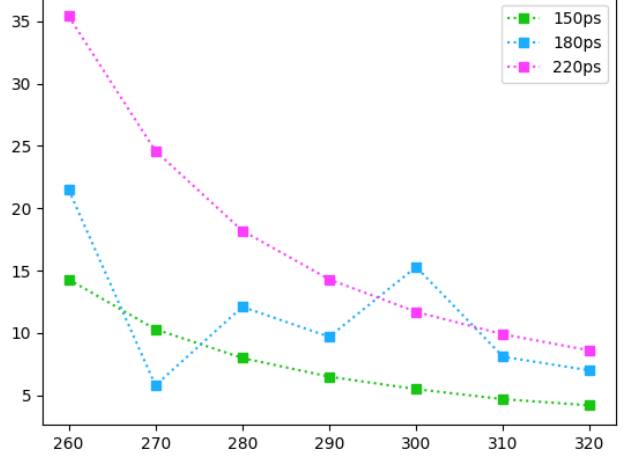


(e)  $\tau_1 = 80\%, \tau_2 = 20\%$

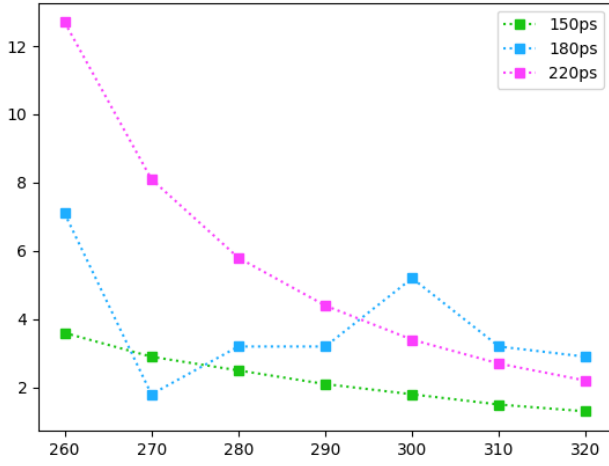
Figure 1.5:  $\tau_1$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



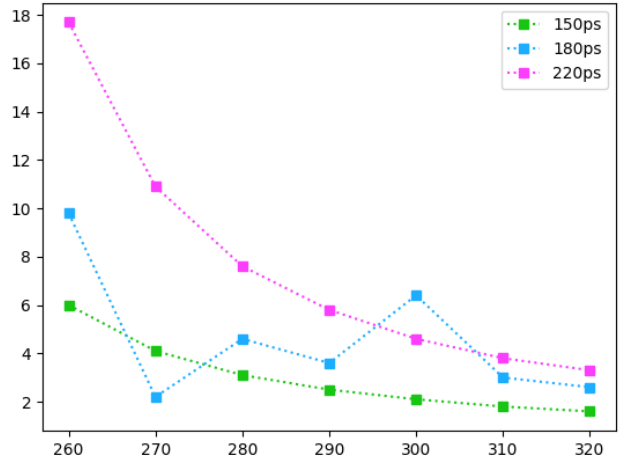
(a)



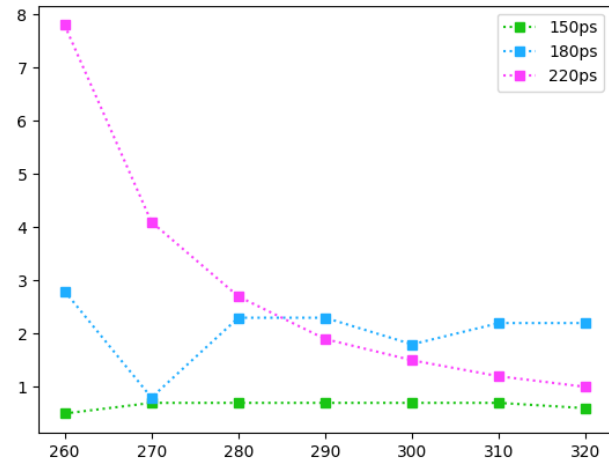
(b)



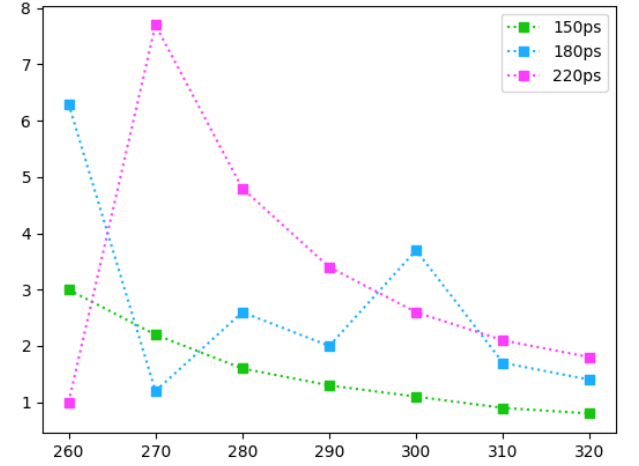
(c)



(d)

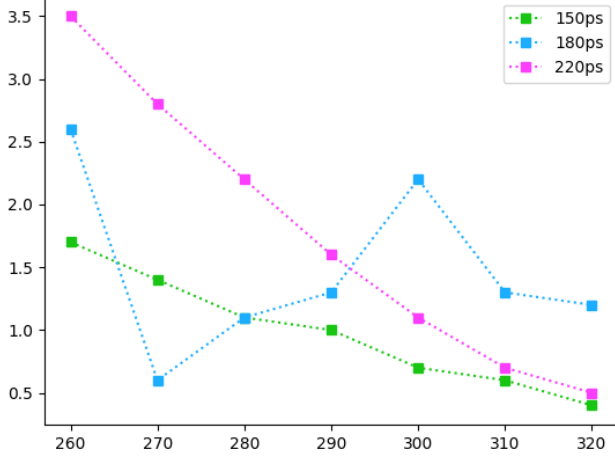


(e)

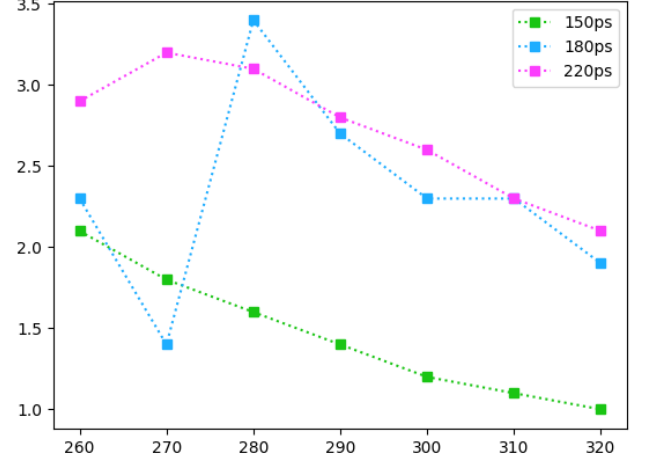


(f)

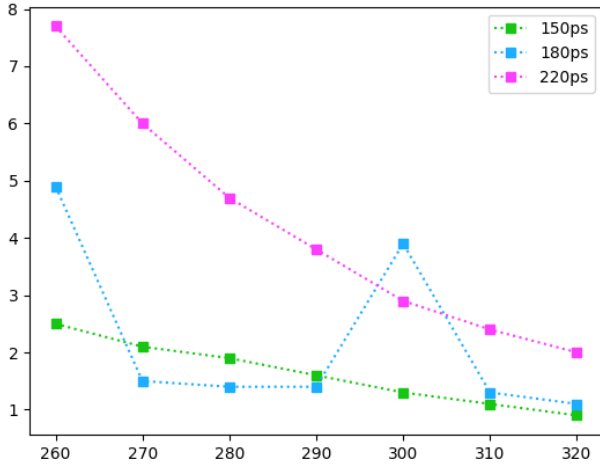
Figure 1.6:  $\tau_2$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



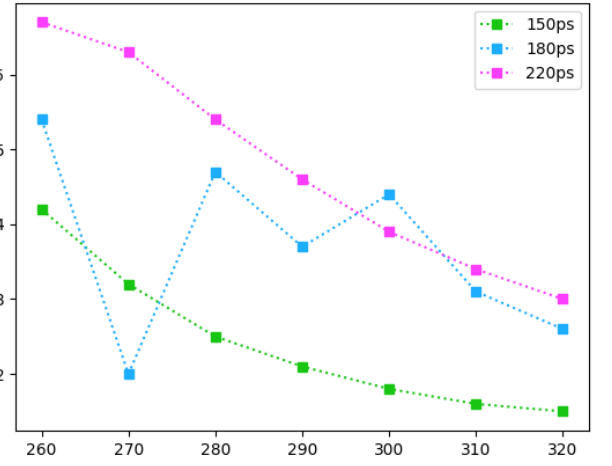
(a)



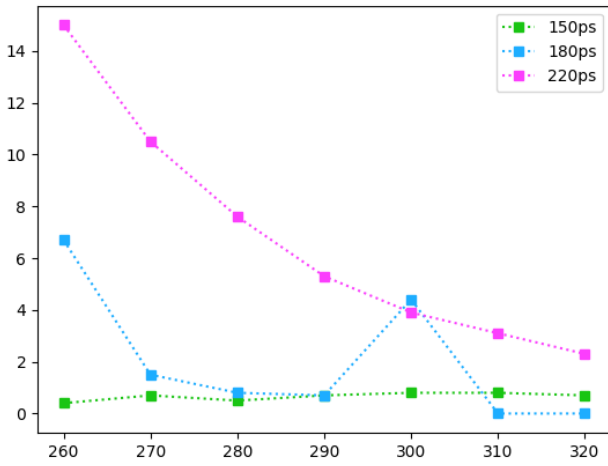
(b)



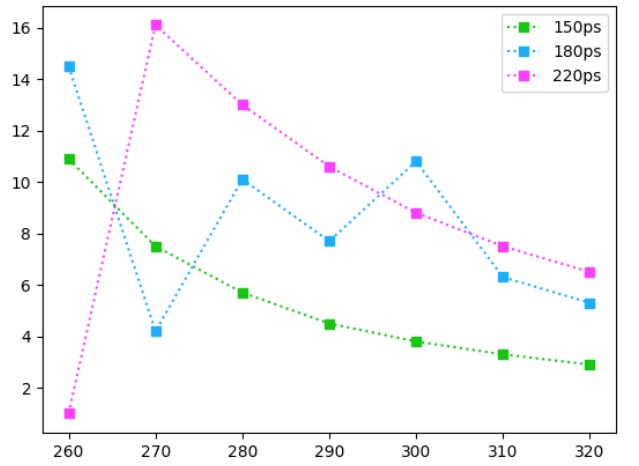
(c)



(d)

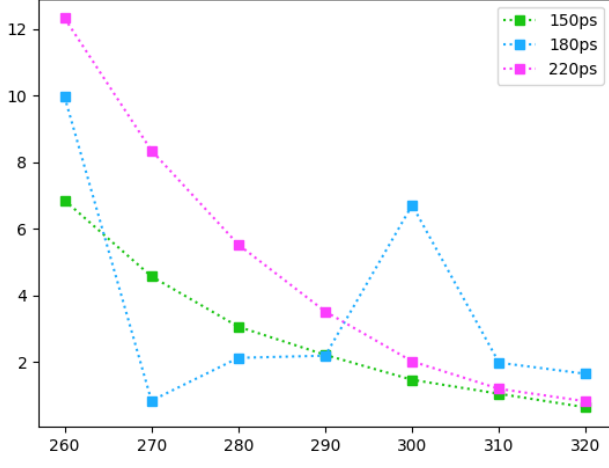


(e)

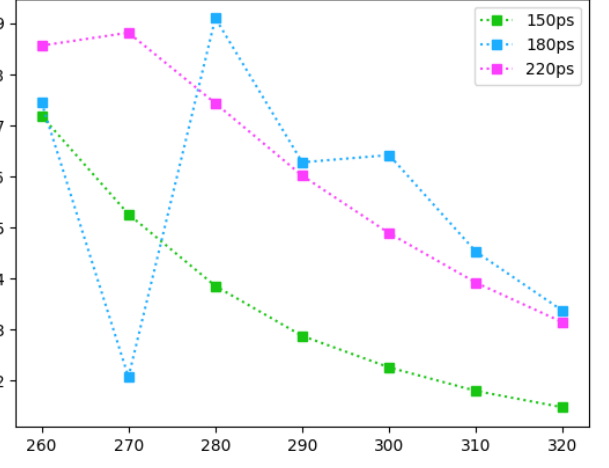


(f)

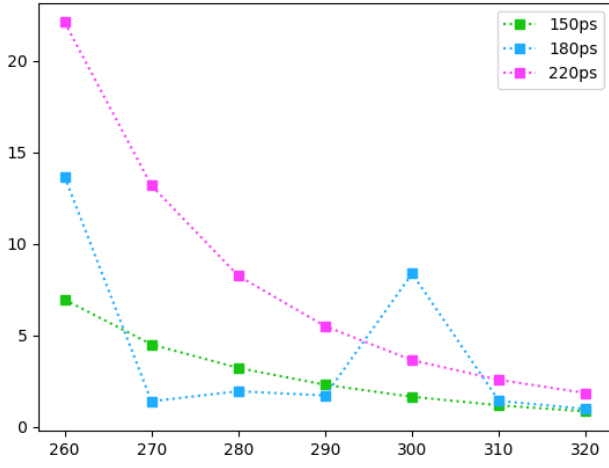
Figure 1.7: intensities, rows = 20-80, 50-50, 80-20, columns = diff, std dev



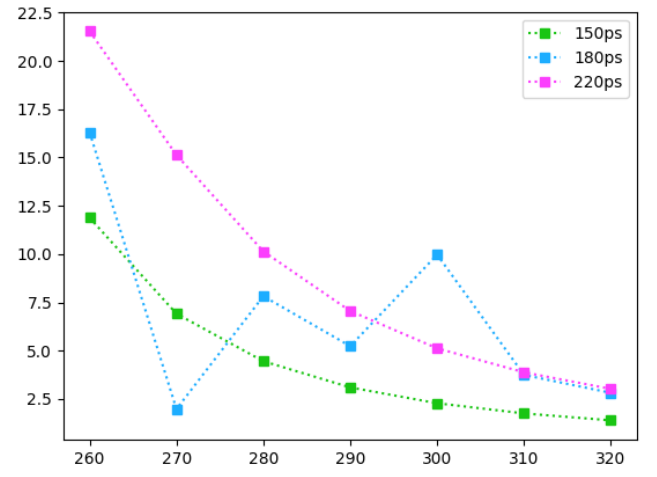
(a)



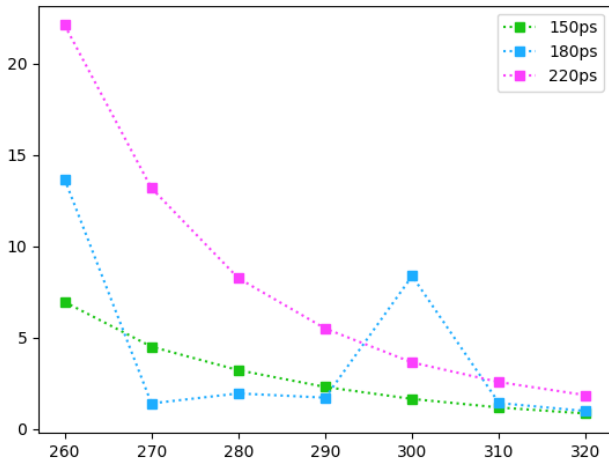
(b)



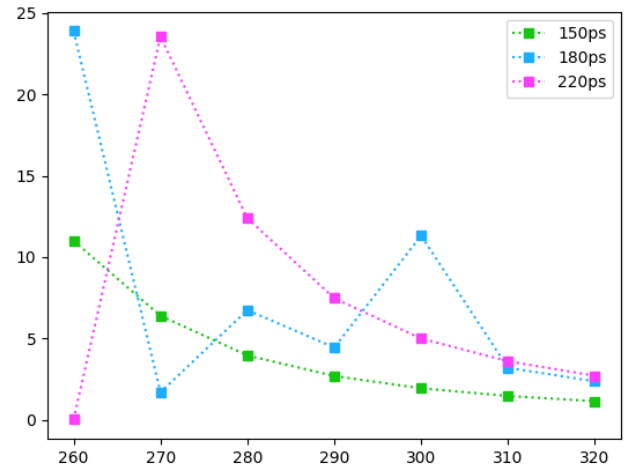
(c)



(d)



(e)



(f)

## 1.5 Single Gaussian IRF

To investigate how the instrument resolution function affects the software, the three Gaussian resolution function in use so far can be simplified to a single Gaussian (refer to ...) . As a sanity check, we can compare how the program performs with a single Gaussian resolution function versus the previous three Gaussian resolution function. Setting  $\tau_1 = 150\text{ps}$ , as this previously gave us the best results, we get the data in Figure 1.8. We can then compare this to Figure 1.3 and see that the single Gaussian resolution function produces remarkably similar results, demonstrating the single Gaussian approximation to be valid.

reference to justification, placed earlier

With that established, spectra were generated and fitted for FWHM values of 220ps, 180ps, 150ps and 100ps, with  $\tau_1$  set to 150ps and  $\tau_2$  ranging from 180-230ps. In general, the trend we expect is the same increase in precision and accuracy with increasing  $\tau_2$  that we see previously, but also an increase in both with narrower resolution functions. We would see this as multiple, separate, decreasing trendlines, ordered so that those corresponding to wider resolution functions were always higher. The results we get are shown in Figures 1.9-1.11, and we can analyse them in two parts.

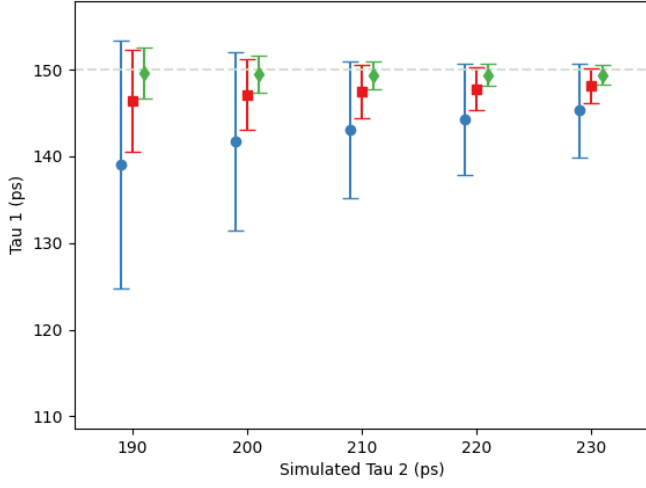
If we just look at the subfigures on the left ((a),(c) and (e) for each figure), which represent the difference between the fit and original data, we see that for the 20%-80% and 50%-50% data the data follows the expected trend. When the  $\tau_1$  intensity is set to %80, however, we notice that accuracy doesn't seem to follow the expected relationship with the width of the resolution function. In many cases we have an outright reversal of the predicted relationship and in each figure we have significant crossing between the trendlines.

Looking at the subfigures on the right, representing the standard deviation of the fitting given by PALSFIT, we again see three outliers where the predicted relationship between precision and resolution function width is reversed, in Figures 1.10b, 1.10d and 1.11b, though in the case of 1.10d, this reversed relationship quickly reverts to the expected one as  $\tau_2$  increases.

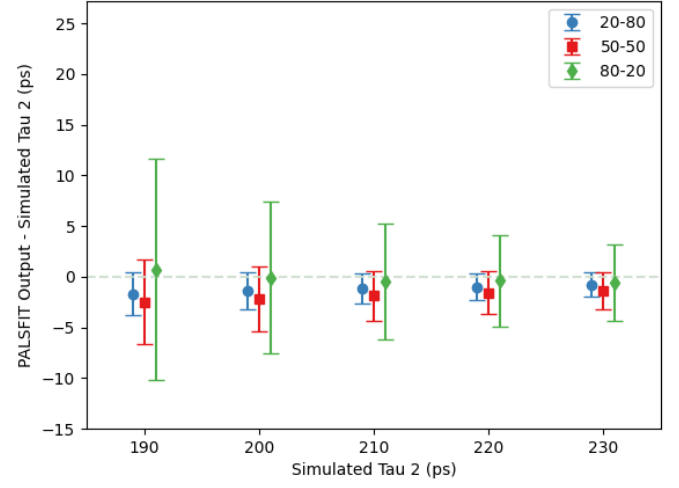
possibly expand



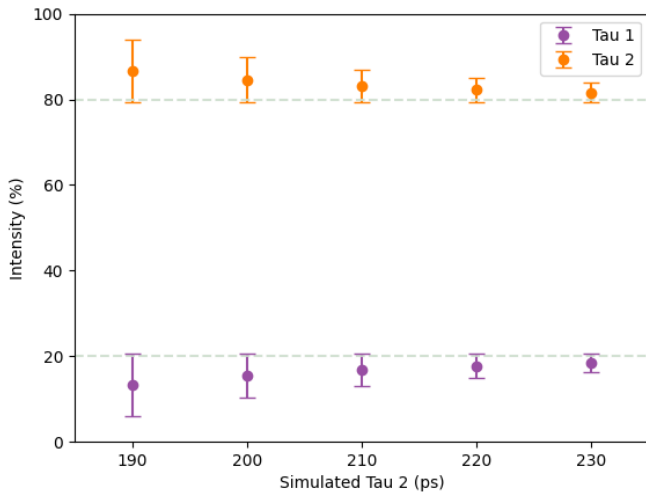
Figure 1.8



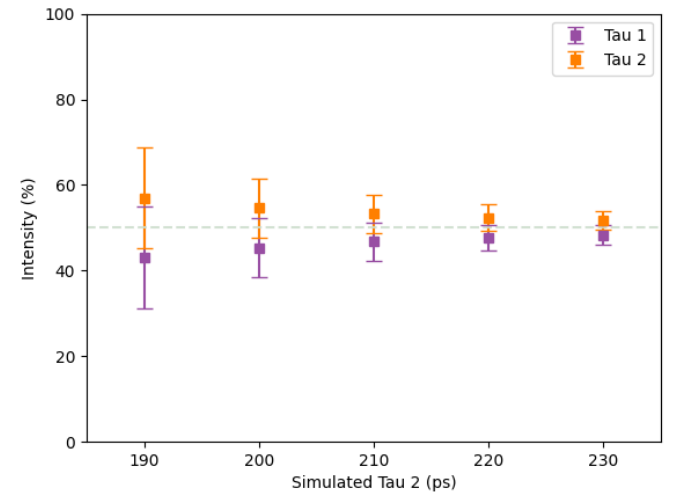
(a) fixed  $\tau_1 = 150$  ps



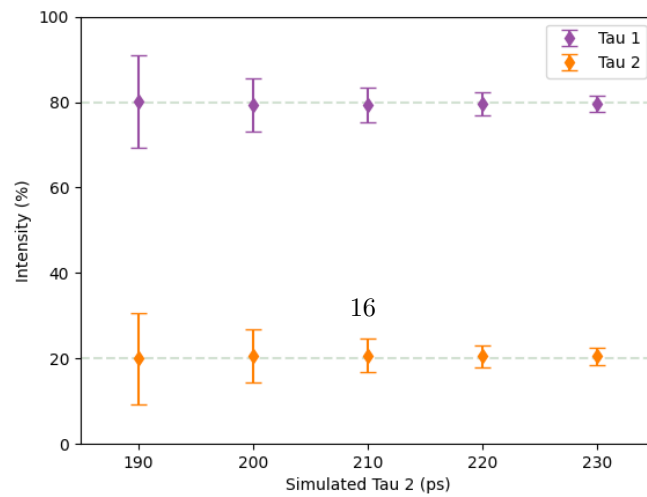
(b)  $\tau_2$  difference



(c)  $\tau_1 = 20\%$ ,  $\tau_2 = 80\%$

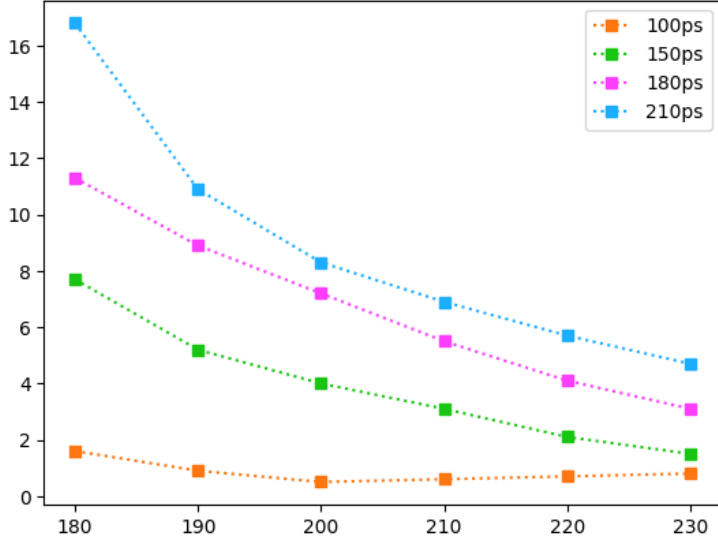


(d)  $\tau_1 = 50\%$ ,  $\tau_2 = 50\%$

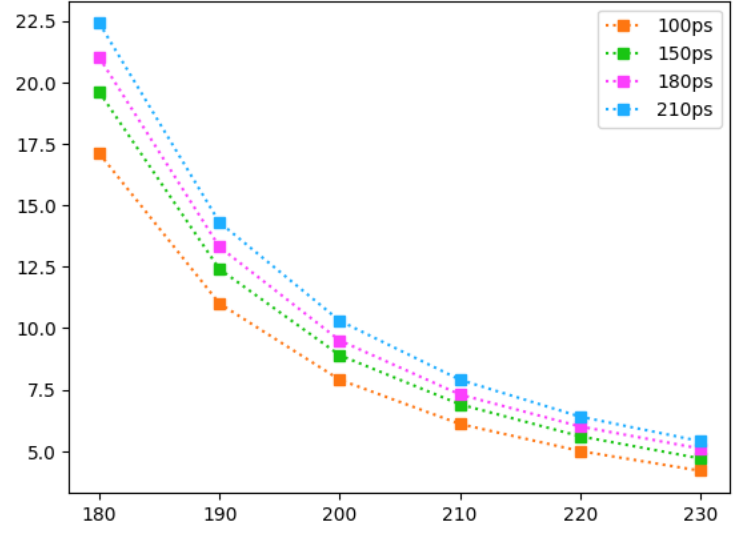


(e)  $\tau_1 = 80\%$ ,  $\tau_2 = 20\%$

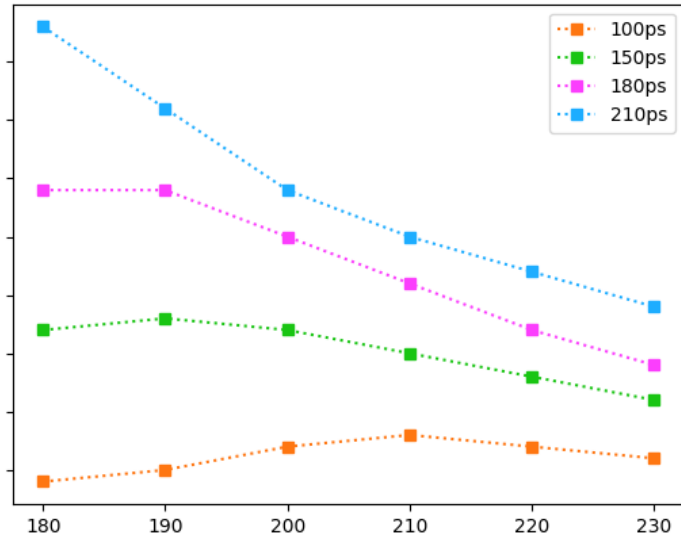
Figure 1.9:  $\tau_1$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



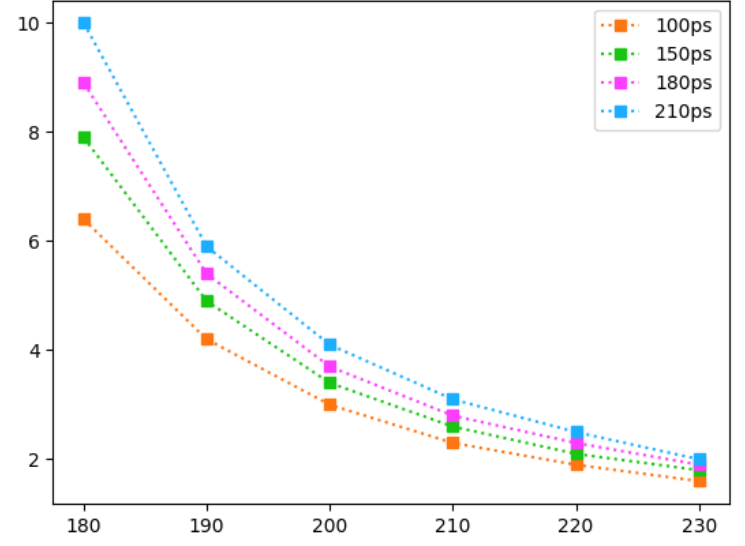
(a)



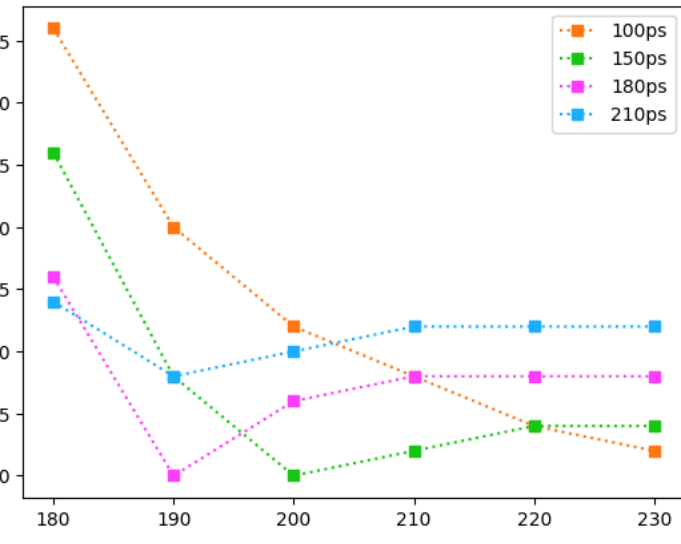
(b)



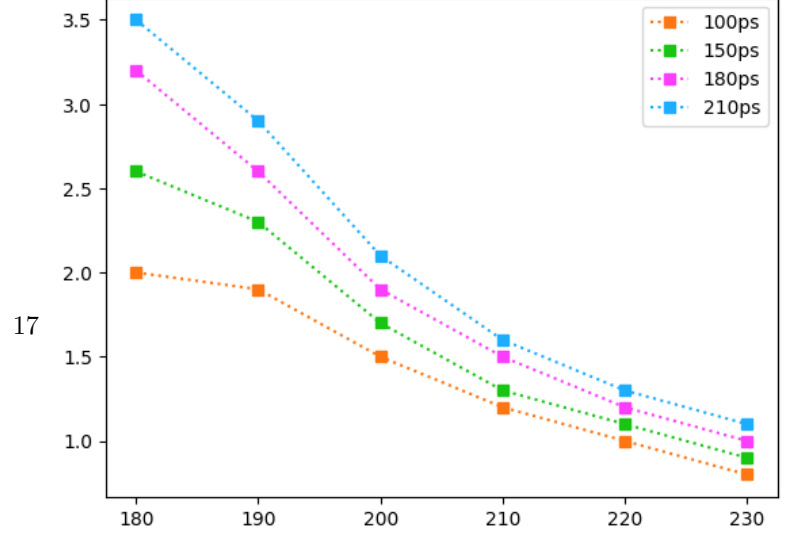
(c)



(d)

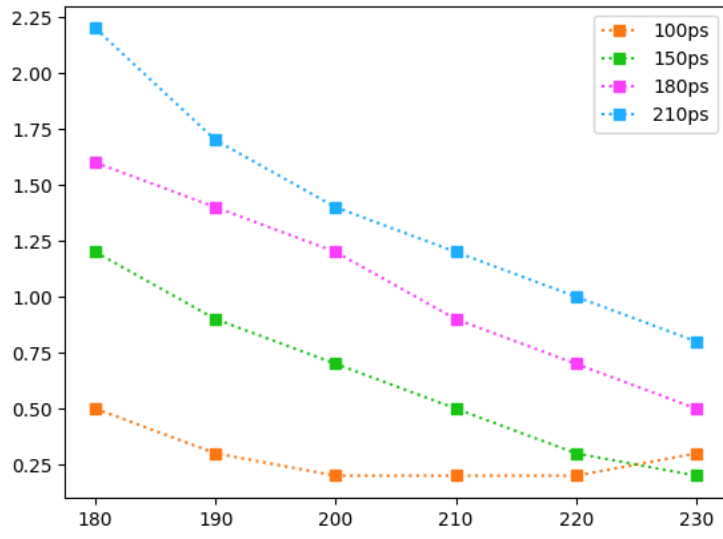


(e)

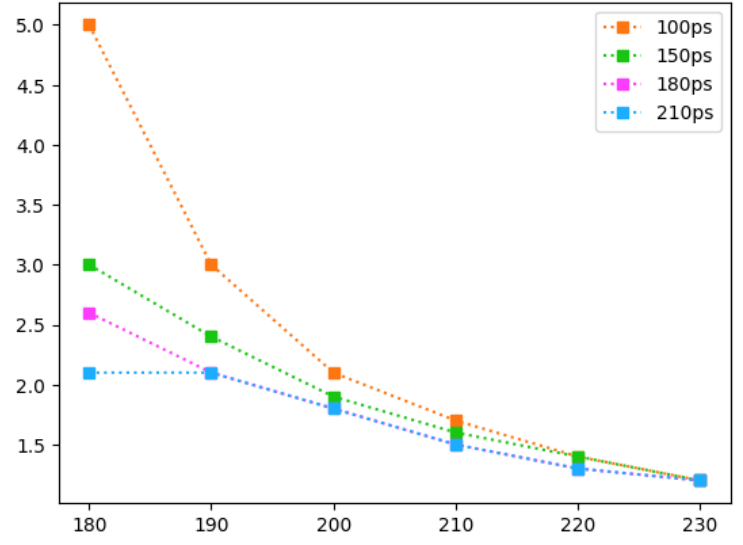


(f)

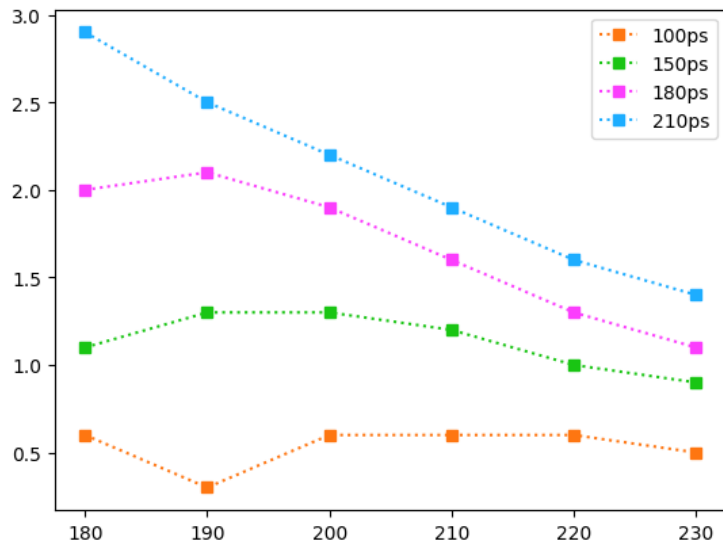
Figure 1.10:  $\tau_2$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



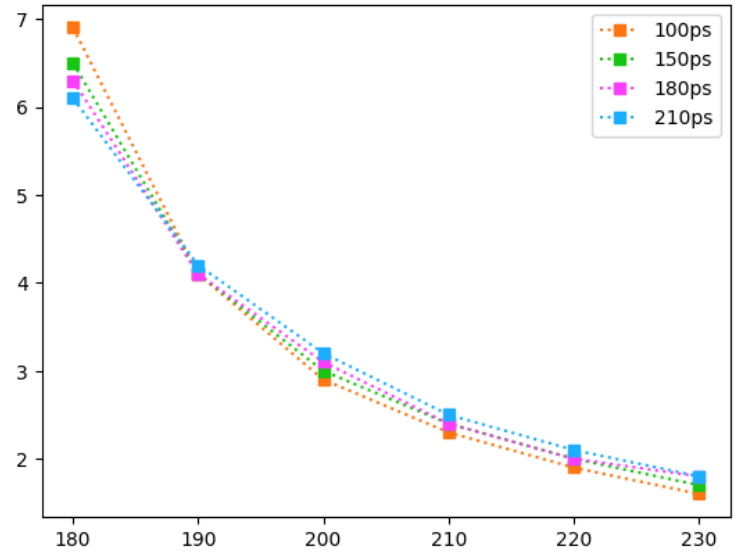
(a)



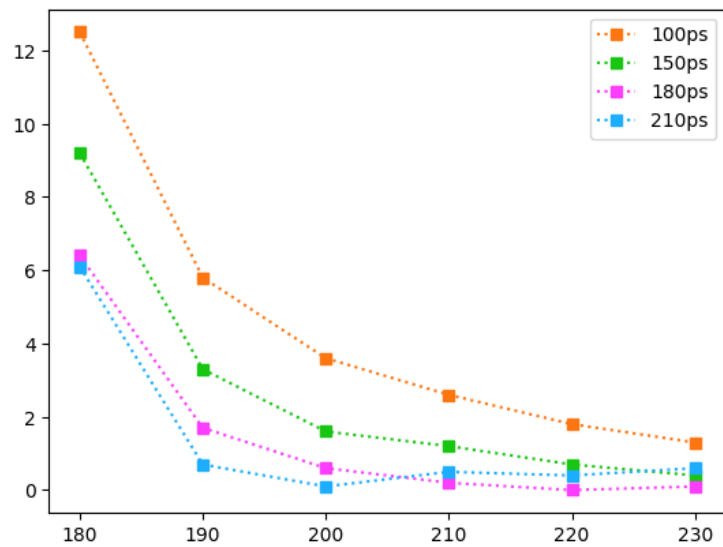
(b)



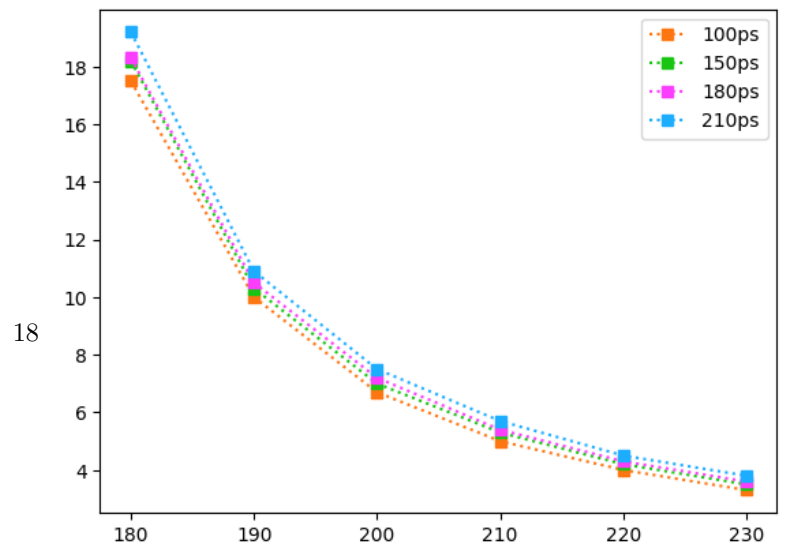
(c)



(d)

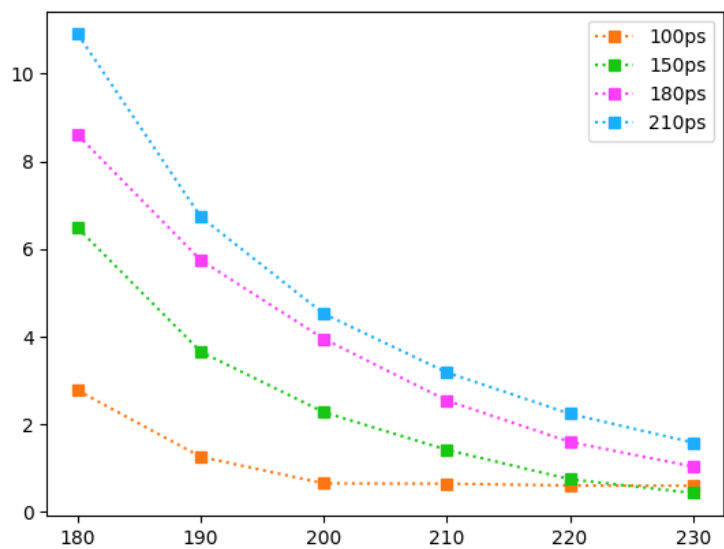


(e)

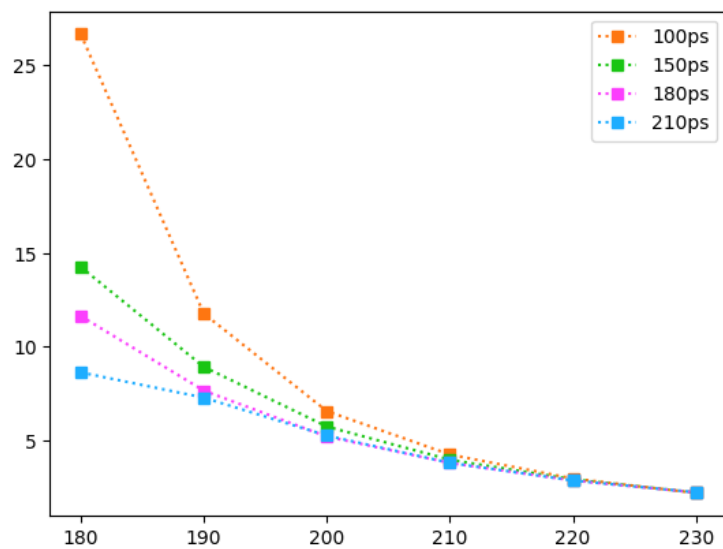


(f)

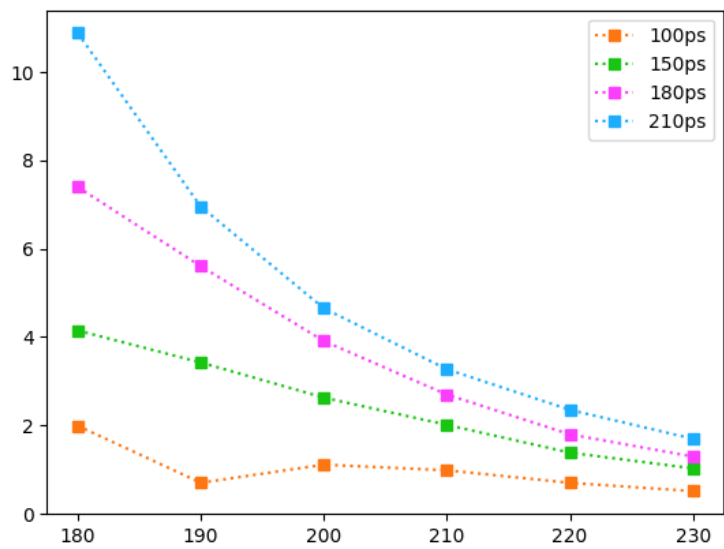
Figure 1.11: intensities, rows = 20-80, 50-50, 80-20, columns = diff, std dev



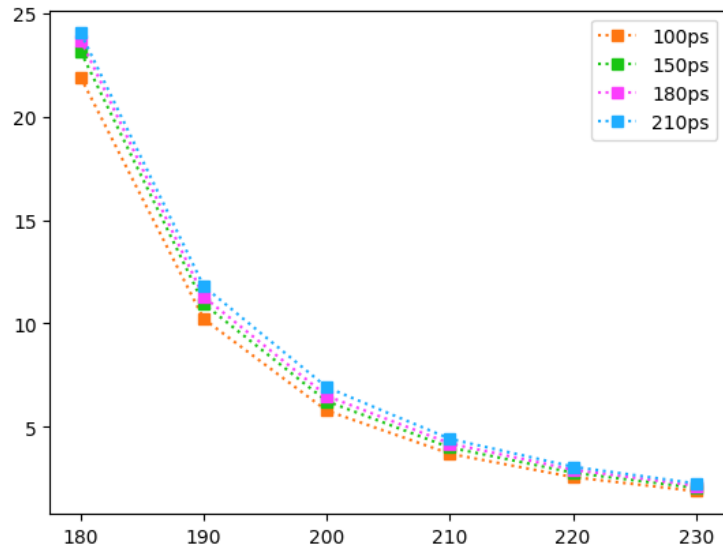
(a)



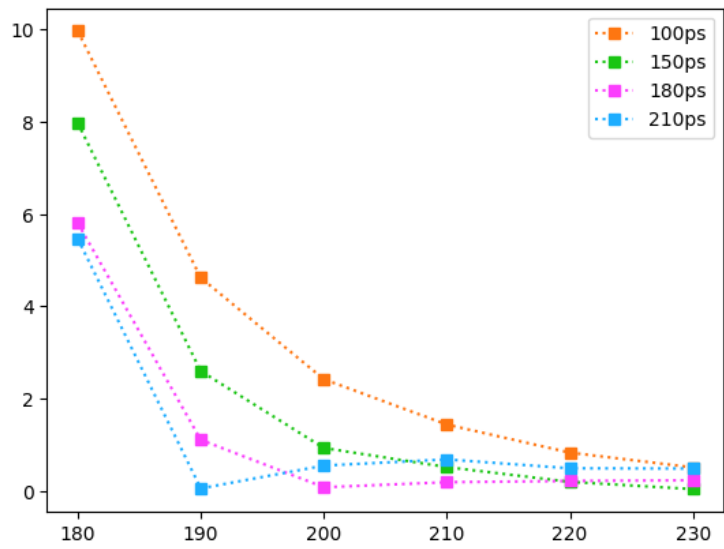
(b)



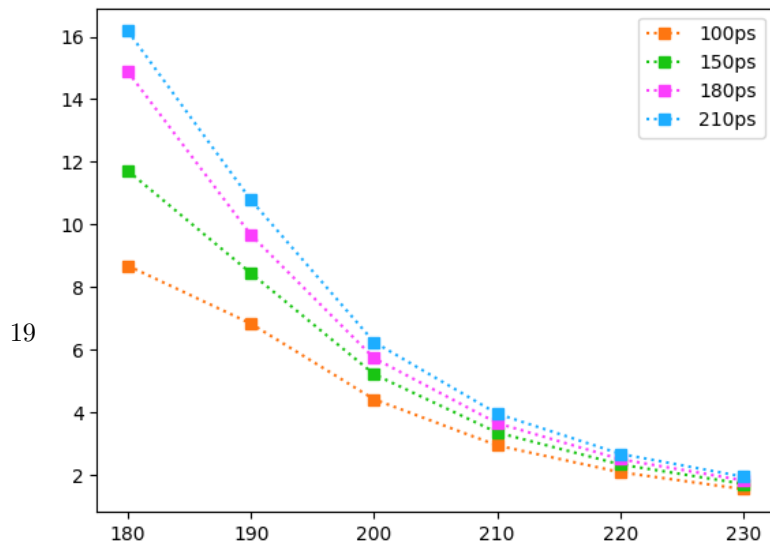
(c)



(d)



(e)



(f)

## 1.6 Number of counts

Another variable worth examining is the total number of counts in a given spectrum. When generating spectra using PALSSIM, this is given as the area of the spectrum without the background and a separate background value. Both were kept constant in all previous runs, with the area set to  $5.65 \times 10^6$  and the background value set to 8.5. Expectations are that a higher number of counts would make spectrum analysis easier. To test this hypothesis, spectra were generated for  $\tau_1 = 150\text{ps}$ ,  $\tau_2 = 180\text{-}230\text{ps}$  and a 210ps FWHM single gaussian resolution function, tripling the background area to  $1.7035 \times 10^7$ .

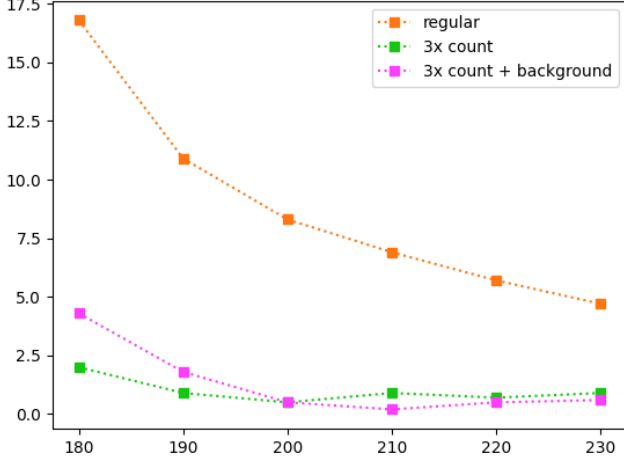
If we just increase the total area of the spectra and maintain the same background value as before, however, we inadvertently end up increasing the signal to noise ratio. While this would be desirable, in a practical setting an increase in total number of counts most likely corresponds to an increase in background noise. As such, spectra were also generated with both the tripled area and a proportional 3x increase in the background value (set to 25.5).

The generated spectra were then analyzed with PALSFIT, and the results are summarized in Figures 1.12-1.14, with a "regular" run for comparison. The regular run is simply the one represented in Figure 1.8, which used the original, unaltered area and background value.

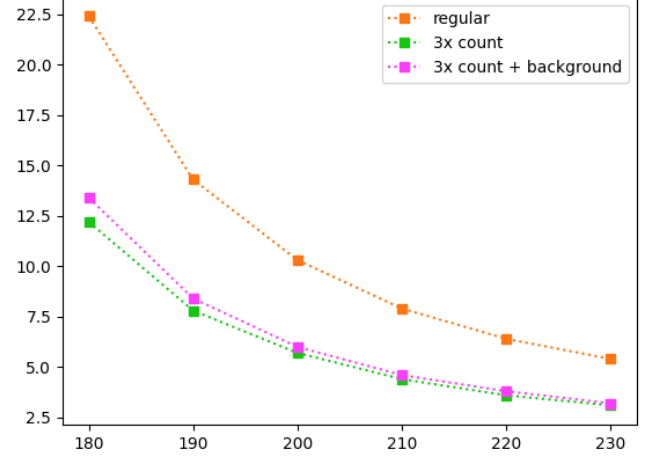
Looking first at the standard deviations (subfigures (b), (d) and (f), on the right side of the page), we see that, as expected, the data from the regular run is in most cases significantly less precise than the other two runs. Additionally, the data with the proportionally scaled background value seems to be slightly less precise than its unscaled counterpart, though in many cases just barely so, if at all. Exceptions tend to occur, however, for the shortest lifetime separation, when  $\tau_2$  is set to 180ps.

Looking at the left side of the figures ((a), (c) and (e)), we see that for the most part the first trend seem to hold true, with the regular run being significantly less accurate than the other two. The exception to this seems to be when the relative  $\tau_1$ - $\tau_2$  intensity is 80%-20%, in which we often have a reversal of expectations, at times with some significant crossover of the relevant trendlines. While, at a glance, the normal and scaled background triple area runs are still for the most part relatively close, with the proportional background looking less accurate overall, this isn't as universally true as it is when looking at the standard deviation. Specifically in Figures 1.12a, 1.13a and 1.14a, when  $\tau_1 = 20\%$  and  $\tau_2 = 80\%$ , PALSFIT seems to have performed worse with the non-scaled background, at least for significant portions of the graphs.

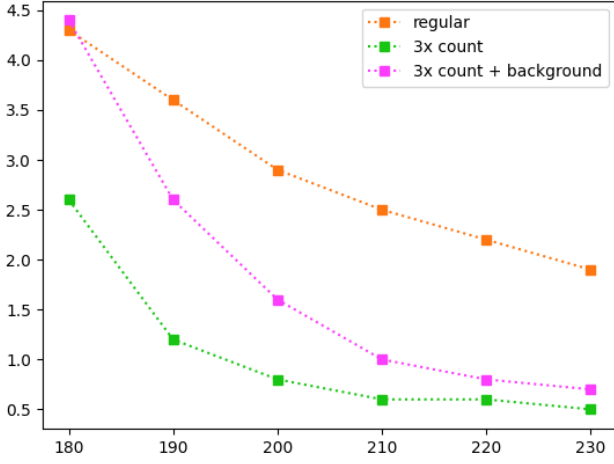
Figure 1.12:  $\tau_1$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



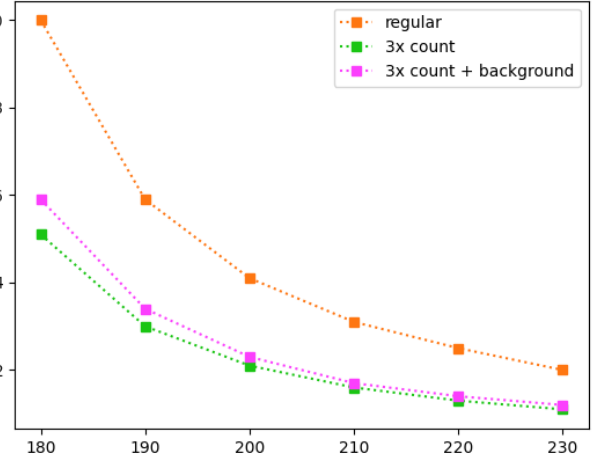
(a)



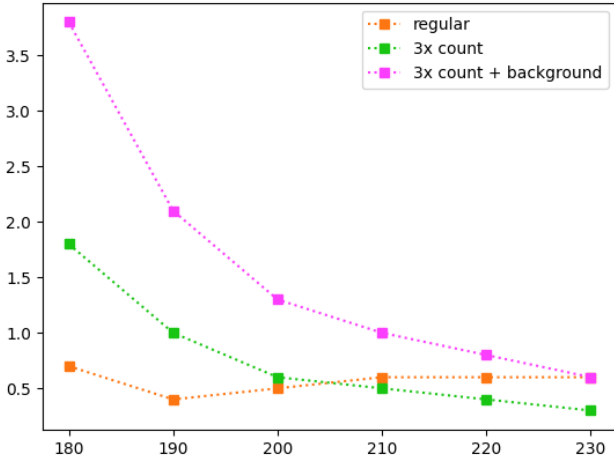
(b)



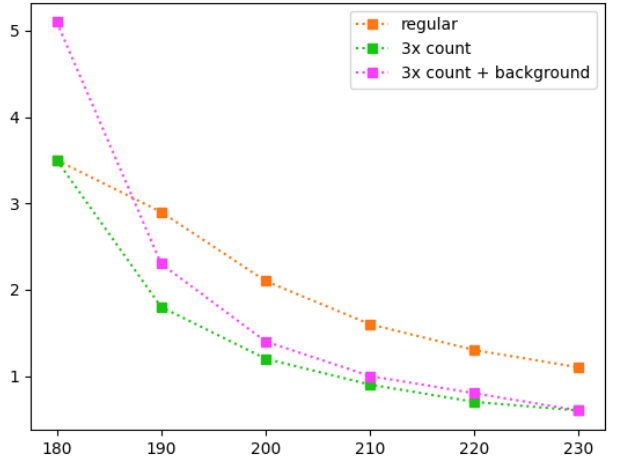
(c)



(d)

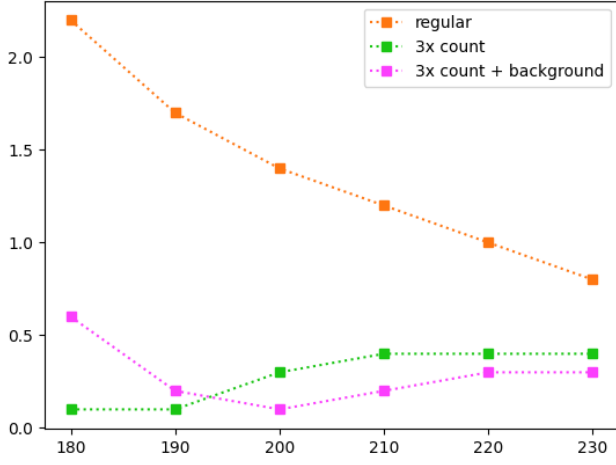


(e)

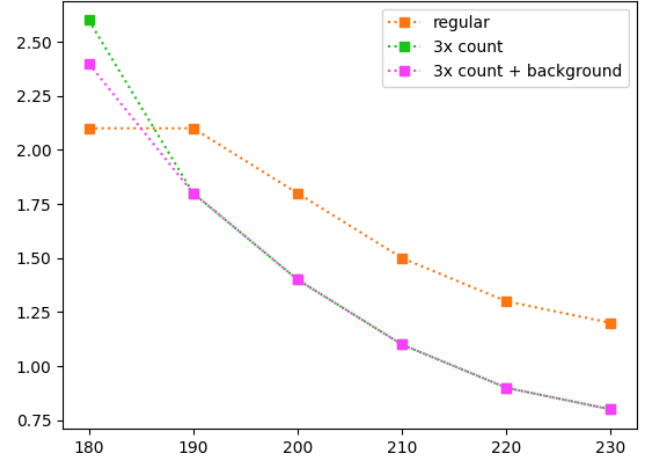


(f)

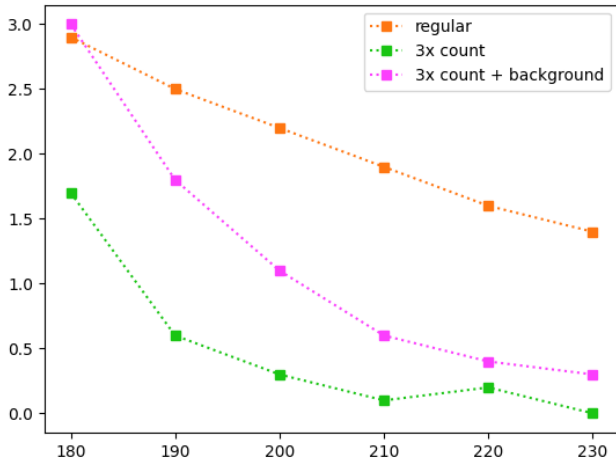
Figure 1.13:  $\tau_2$ , rows = 20-80, 50-50, 80-20, columns = diff, std dev



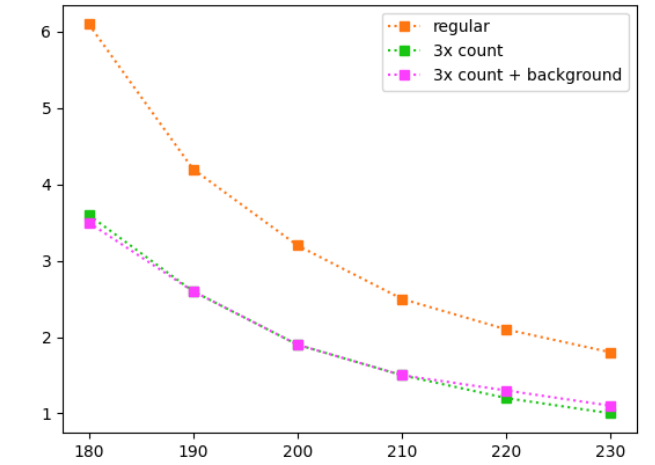
(a)



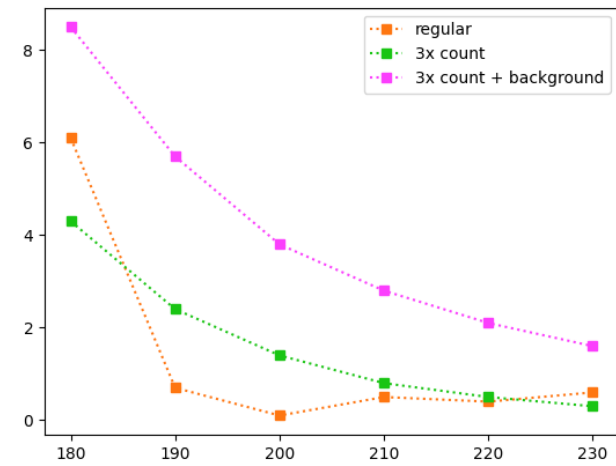
(b)



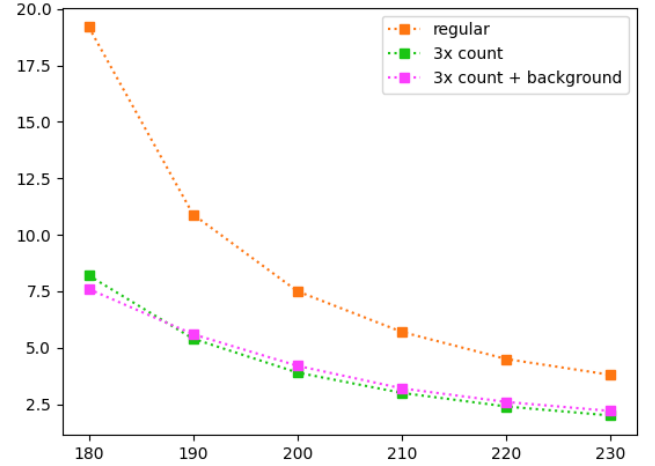
(c)



(d)

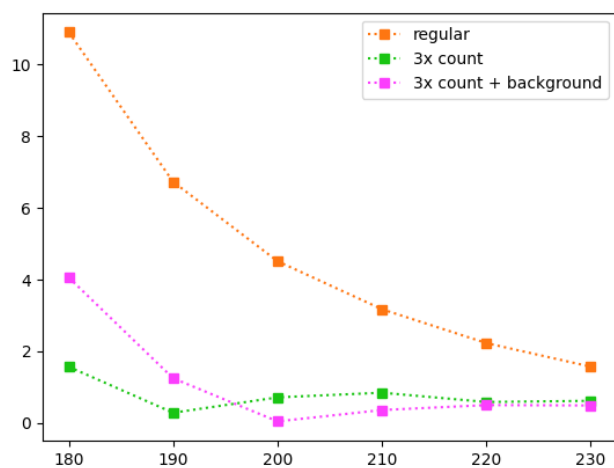


(e)

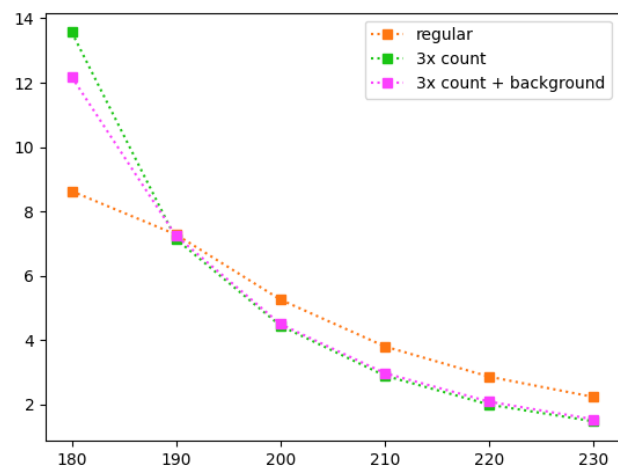


(f)

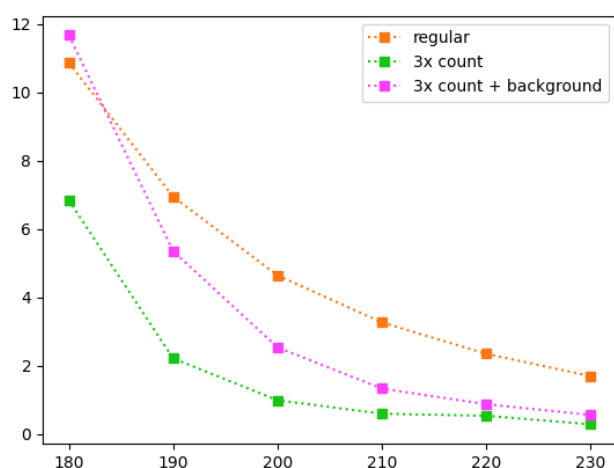
Figure 1.14: intensities, rows = 20-80, 50-50, 80-20, columns = diff, std dev



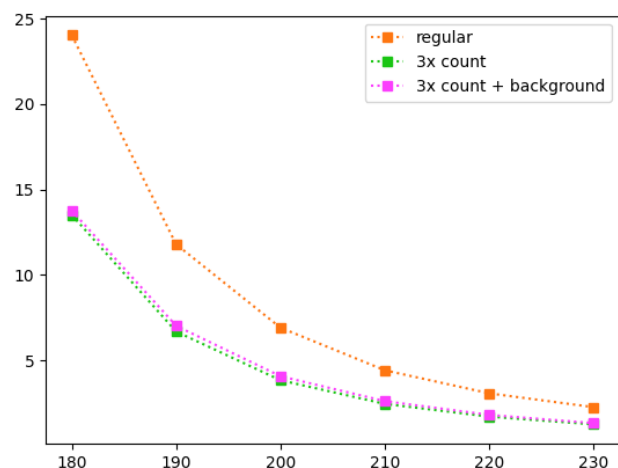
(a)



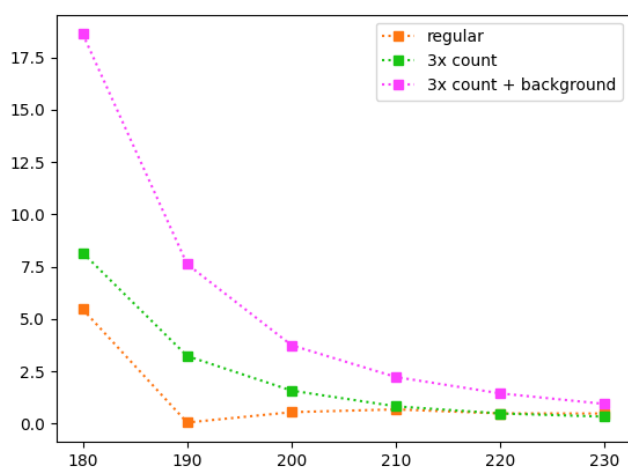
(b)



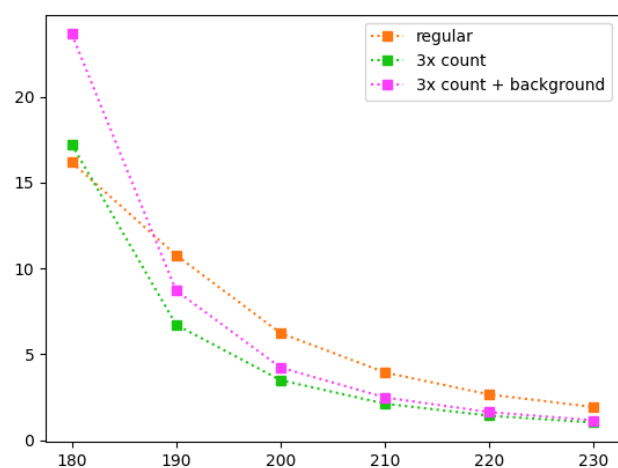
(c)



(d)



(e)



(f)



## 1.7 Realistic data

Analysis of real world data often differs from how we've conducted our analysis so far. One key factor is that usually the number of lifetimes present in a real life spectrum isn't known beforehand. As PALSFIT requires the user to input the number of fitted lifetimes, this might lead to situations in which the user predicts the wrong number of lifetimes, thus it might be useful to explore how the software behaves in such a situation.

To do so, three materials (?) were modeled, with three lifetime components. All three have two shorter components that differed between them, and a long lifetime component set to 2.6ns with a 0.15% intensity that was kept equal. The relative intensities of the two shorter lifetime components were adjusted to the following values:

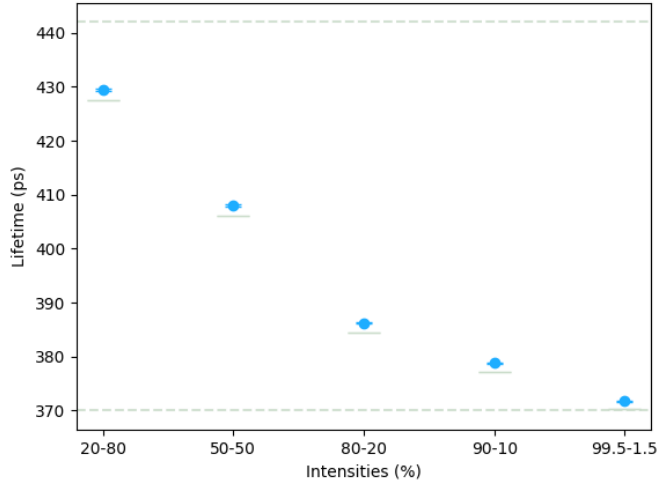
relative		absolute	
$\tau_1$	$\tau_2$	$\tau_1$	$\tau_2$
99.5%	0.5%	99.3508%*	0.4993%*
90%	10%	89.865%	9.985%
80%	20%	89.88%	19.97%
50%	50%	49.925%	49.925%
20%	80%	19.97%	9.985%

(a) intensities

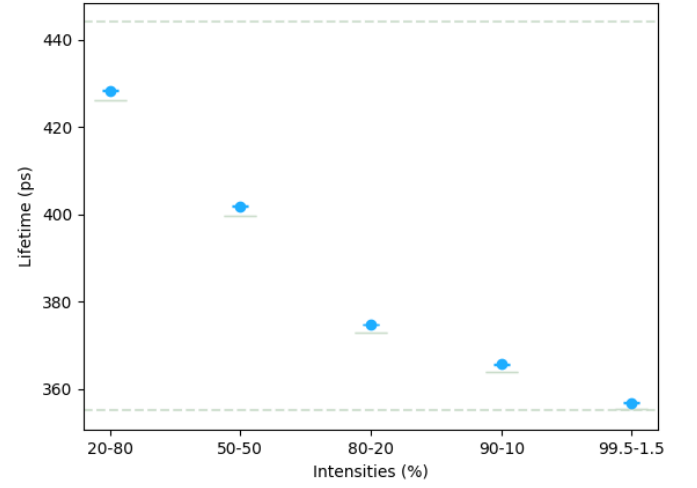
$\tau_1$ (ps)	$\tau_2$ (ps)	$\tau_3$ (ns)
370	442	2.6
355	444	2.6
348	440	2.6

(b) lifetimes

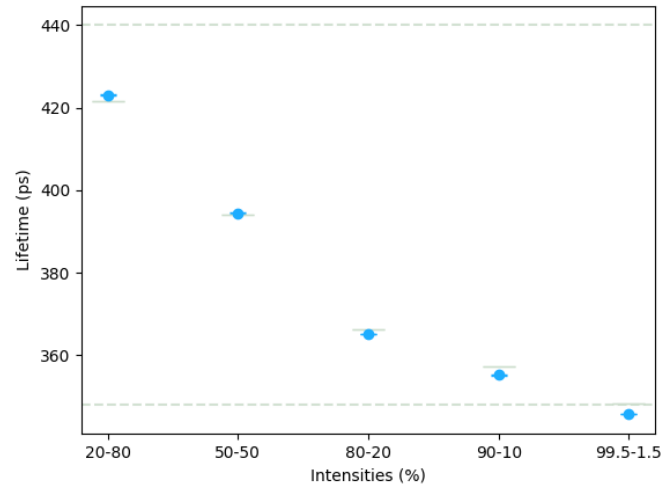
When a single lifetime is fitted, the resulting lifetime seems to tend towards a weighted average of the two shorter lifetimes, as can be seen in the following figures:



(a)  $\tau_1 = 370$ ,  $\tau_2 = 442$

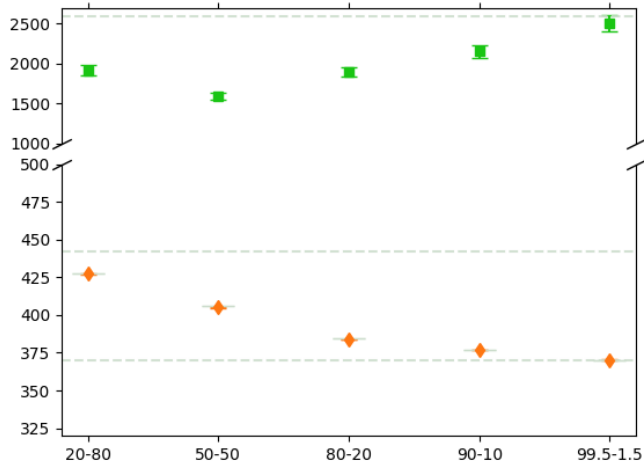


(b)  $\tau_1 = 355$ ,  $\tau_2 = 444$

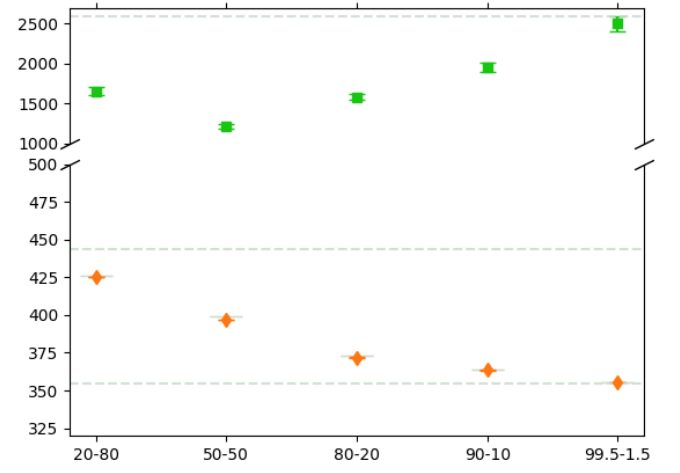


(c)  $\tau_1 = 348$ ,  $\tau_2 = 440$

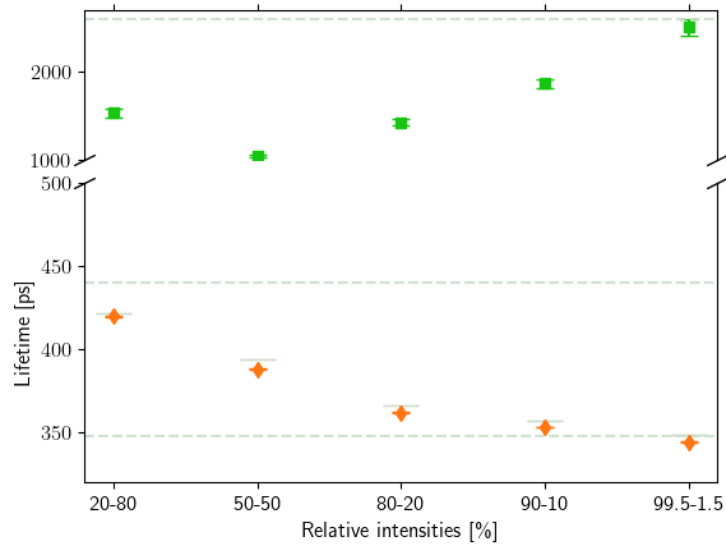
Figure 1.15: single lifetime fit



(a)  $\tau_1 = 370, \tau_2 = 442$

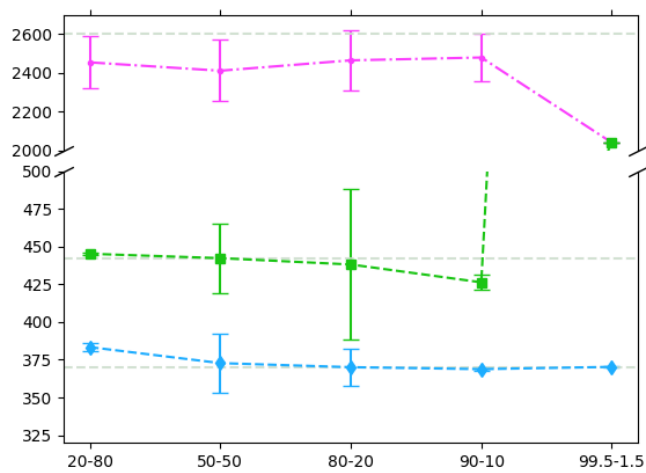


(b)  $\tau_1 = 355, \tau_2 = 444$

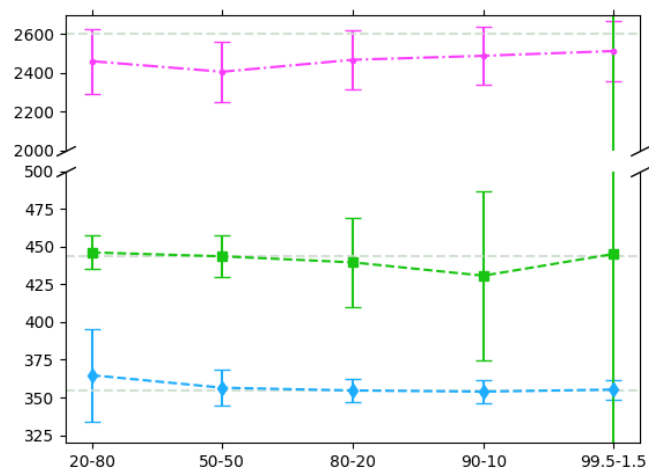


(c)  $\tau_1 = 348, \tau_2 = 440$

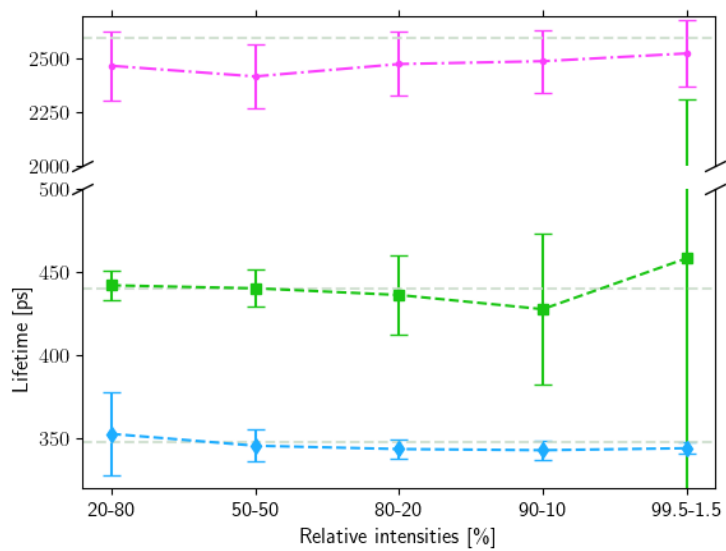
Figure 1.16: two lifetime fit



(a)  $\tau_1 = 370, \tau_2 = 442$



(b)  $\tau_1 = 355, \tau_2 = 444$



(c)  $\tau_1 = 348, \tau_2 = 440$

Figure 1.17: three lifetime fit

## 1.8 Using the STM to vary $\tau_1$

:

Using STM, vacancy concentration rate was varied for a given value of defect lifetime  $\tau_2$ . This was done for  $\tau_2 = 370,442$ . Vacancy conc. rate was varied as such:

Table 1.3: Vacancy Conc.

Conc.
1E15
2E15
4E15
6E15
8E15
1E16
2E16
4E16
6E16
8E16
1E17
2E17
4E17
6E17
8E17
1E18