Fluorescence Lifetime Fitting Module for HPD-TA

Ta-FIT

User Manual Version 8.1

This manual was produced using *ComponentOne Doc-To-Help.*™

Contents

Legal terms

Please read this license carefully before installing or using the software. By using the software, you are agreeing to be bound by the terms of this license. If you do not agree to the terms of this license, do not install or use the software and erase it from your computer.

COPYRIGHT

The TA-Fit software (herein called the Software) and all accompanying documentation (herein called the Documentation) are copyrighted by Hamamatsu Photonics Deutschland GmbH (herein abbreviated as HPD) with all rights reserved. It is illegal to copy (except as granted under section "License Agreement" below), duplicate, sell, or otherwise distribute the Software or the Documentation or any parts of them without written permission from HPD.

Any attempts to "reverse-engineer", "disassemble", "re-source" the Software or to modify the Software in any way are violations of HPD's copyright and may be legally prosecuted.

LICENSE AGREEMENT

HPD gives you the permission to copy the software for your own use as long as it is absolutely necessary for using the Software for its intended purpose. This includes your right of loading the Software into the computer's memory and making copies of the installation disks for the purpose of backup. You may also run multiple copies of the Software on several computers for the purpose of data inspection and analysis. Under no circumstances, however, are you allowed to use the Software to drive data acquisition hardware on more than one computer simultaneously at any given time.

WARRANTY

All efforts have been made to make the Software and the Documentation as accurate and free of "bugs" as possible. However, HPD makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. In no event is HPD liable for direct, indirect or consequential damages of product or property, personal damage, inconvenience, loss of time, data or profits, which may result from the use of the Software, even if advised the possibility thereof. The entire risk as to the results and performance of the Software is assumed by the purchaser.

Specifically, where life or health is involved, results obtained with the Software should be examined carefully before any conclusions are made. The Software and the Documentation are not designed and tested intending to ensure a level of reliability suitable for use in diagnosis and treatment of humans.

TRADEMARKS

Product names and company names mentioned in the Software or the Documentation are trademarks or trade names of their respective holders.

UPDATES

Update conditions may vary depending on where you purchased this product. Please contact your local Hamamatsu subsidiary or retailer to obtain information about the update conditions valid for you.

HPD is committed to a policy of customer satisfaction and product reliability. If you find any errors, "bugs" or other inconveniences in the Software or the Documentation, please inform HPD or your local retailer so that countermeasures are possible with the next update.

HAMAMATSU Photonics Deutschland GmbH

Arzbergerstr. 10, D-82211 Herrsching, Germany

Phone: +49-8152-3750, Fax: +49-8152-2658, e-mail: info@hamamatsu.de

A. Preface

Dear User,

thank you very much for purchasing TA-Fit. Great care has been invested in the development of this software as well as in the production of its documentation. We hope that this document will serve you well in becoming familiar with the software and its efficient use.

If you **upgraded** TA-Fit from a previous version, please refer to Appendix A6 for a list of changes.

Chapter B is the **Introduction**. Please be sure to read it. It gives you hints **how to use this document** and explains the **installation** of TA-Fit. The chapter also includes a "**Quick Start**" section that allows you to see your first fitting result after only a few minutes.

Chapter C (**Lifetime Fitting - The Basics**) supplies the theoretical background of fluorescence lifetime fitting. It is divided into two parts. The first part is about the "**task**", the second about the "**method**". The first explains the character of fluorescence decay data, from the easy cases to the more convoluted ones. This part is centered on the notion of the **theory function**. The second part introduces the concept of **least-square approximation** and explains that lifetime fitting is an **iterative** process. This part contains also some notes on **statistics**, which are important for a good understanding.

Chapter D (**How to Use the Fitting Windows**) is the **operating instruction**. In this chapter, every part of the user interface is explained - what it does and how to use it correctly.

Chapter E (**Judging the Quality of the Fit**) explains the definition and usage of those output data of TA-Fit which allow you to judge a given fit. Its understanding is important for obtaining reliable results.

Chapter F (**Questions & Answers**) addresses some randomly collected questions and problems which are probably typical for a new user.

Chapter G is the **Tutorial**. Working through this tutorial is probably the most efficient way of getting familiar with TA-Fit. We **highly recommend** that you spend the time for it. You do not need to operate your streak camera for doing so, since the tutorial self-contains all required data.

In the **Appendix** you will find various resources such as a description of **file formats**, a brief **bibliographical reference** and a **glossary** with the most frequently used terms.

A final note. If you encounter any serious problem with the software or if you find a mistake in this document (missing points, unclear descriptions, or downright errors), we will be grateful if you inform us. HAMAMATSU is committed to a firm policy of customer satisfaction. In order to succeed this, we are dependent on feedback from our users.

Your HAMAMATSU Team

B. Introduction

In this chapter, we describe how to get TA-Fit up and running, and what is the best and quickest way to get familiar with it.

But first let us start with an overview about the basic purpose of TA-Fit.

1. The Purpose of TA-Fit

The purpose of TA-Fit is the extraction of physically relevant parameters from fluorescence decay data taken with a HAMAMATSU streak camera system and the HPD-TA streak control software. By physically relevant parameters we mean the decay lifetimes and relative amplitudes of the decay components.

This is done by means of an iterative algorithm which tries to change the parameters of a "theory function" (= a function which describes the fluorescence intensity in a general way) so that it matches the measured data best. Finding this match is termed "fitting", and the general strategy is to minimize a global measure of the error of the fit. This global measure is the sum of the error squares, and so the method is an instance of the widely known principle of "nonlinear least mean square approximation".

In order to do this job well, TA-Fit also has to take other complications into account, such as a limited temporal resolution of the detection system or distorted or broadened excitation light pulses (the employed technique here is known as "deconvolution").

Originally, the data obtained by a streak system are two-dimensional (images), but TA-Fit will require to extract one-dimensional data (intensity profiles) before the fitting procedure can be started.

TA-Fit allows the user to change various conditions of the fitting procedure in a way which is easy to understand and easy to use. It should enable even persons who never have done fluorescence lifetime fitting before to get customized to it very quickly and to produce reliable results after only a short training period.

2. How to Use this Document

How you use this document depends on whether you are already familiar with fluorescence lifetime fitting.

If you are new to the field of lifetime fitting, we recommend reading the whole document (there are hints in front of very technical sections that may be safely skipped).

If you are familiar with lifetime fitting in combination with a streak camera but have never used TA-Fit before, we recommend to browse at least chapter D (operation instruction), as well as section F8 (on statistics) and Chapter E (on quality parameters). Then, go through the tutorial tour.

If you are very impatient read section 4 below (Quick Start) and go ahead on your own risk.

If you upgraded from a previous version of TA-Fit and want to get only an overview of the changes, please simply refer to Appendix A6.

In any case, we recommend trying "Quick Start" (see below).

If you encounter a phrase or technical term that you are not familiar with, you should consult the glossary or the main text via the index.

3. Installing the Software

Please install the HPD-TA in the usual way on your hard drive. (Refer to the HPD-TA manual.or help file) The TA-Fit program files will reside in the HPD-TA directory itself.

Please be sure that you plug the hardware lock (sometimes called a "dongle") you obtained together with TA-Fit on the parallel port of your computer. If the hardware lock is missing, you will obtain a message during startup of the HPD-TA, and TA-Fit and other functions will be disabled. Also, if you use a wrong version of the hardware lock (perhaps an old one which you had before you purchased TA-Fit) HPD-TA will start normally but TA-Fit will be disabled.

The installation is correct if you don't get a message on startup and if you can find the entry "Fitting" in the "Analysis" menu of HPD-TA.

If you want to go through the Tutorial tour (which we *highly* recommend), you also need to install it from the distribution diskettes or CD-ROM. If you have chosen "Typical" during installation, this has already been done. If not, you can repeat the installation choosing "Custom", deselecting all options except "TA-Fit Tutorial". The tutorial files will reside in a directory called "Tutorial" below your HPD-TA directory.

4. Quick start

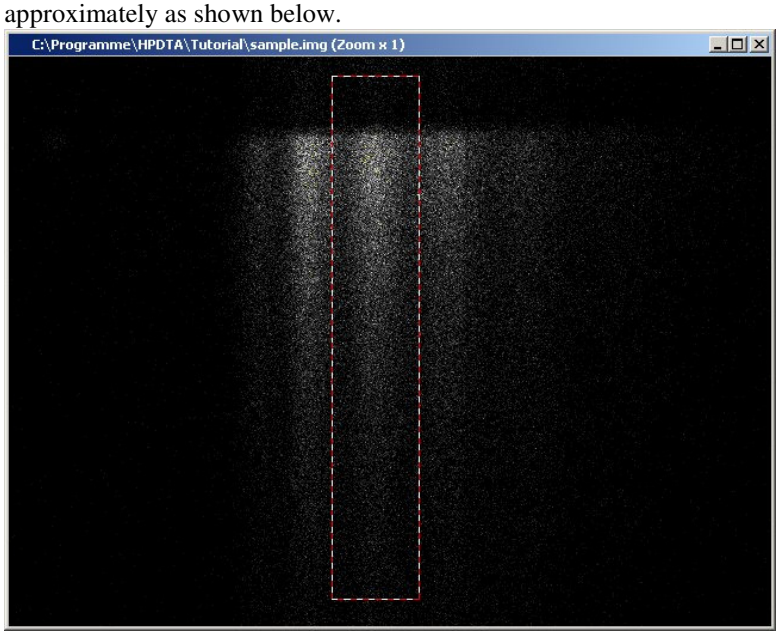
This section guides you through the basic steps of extracting data from a streak image and performing some simple mono-exponential fit on them. This is also a good test to confirm that TA-Fit is correctly installed and running well on your machine.

Please note that this Quick Start tour requires that you have installed the Tutorial files on your hard drive (refer to section 3 above).

Please also note that the tour assumes that you are already familiar with HPD-TA itself, since you will need some basic HPD-TA skills that are not detailed here (e.g. loading images and profiles, handling of ROIs, etc.). We do *not* recommend using TA-Fit before being familiar with HPD-TA.

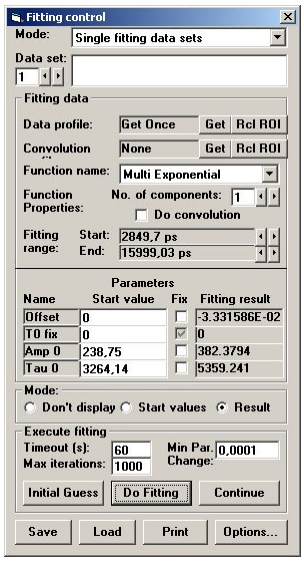
Please follow each step exactly as described below. If something fails, please first ensure that you didn't do a mistake or skipped something. If you cannot resolve the problem, please contact HAMAMATSU.

1. Start HPD-TA in the usual way.
2. Load the image named "Sample.IMG" from the Tutorial directory. (The image shows the time-resolved spectrum of anthracene, taken with a C4334 streak camera in photon-counting mode with an integration time of about half a minute.)
3. In the image display, define a region-of-interest in the usual manner at a position where we want to extract the intensity profile. Select a central region approximately as shown below.



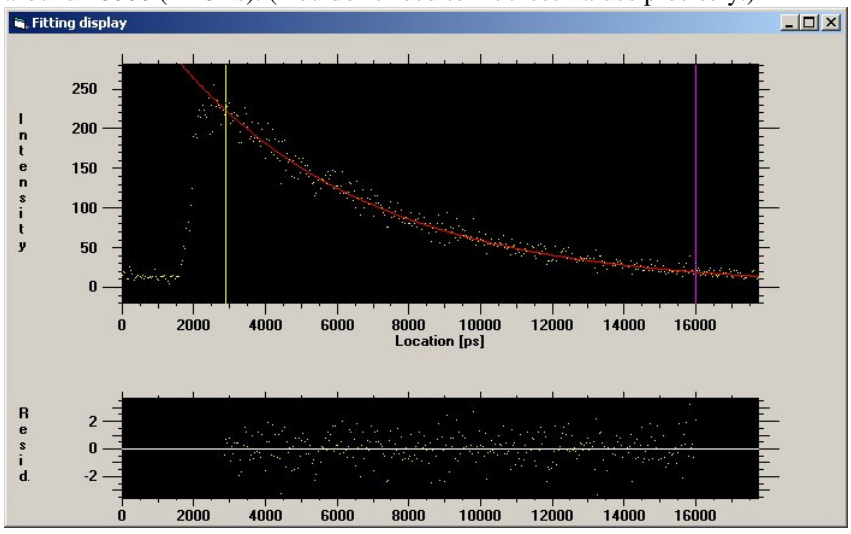
4. Invoke TA-Fit by selecting "Fitting" from the "Analysis" menu. Two windows will open, a control window and a display window. Increase the horizontal size of the display window until it looks convenient.
5. In the control window, press the "Get" button at the end of the "Data profile" line (see picture right). A small window will open. Select "Get once from image" and "vertical" and hit OK. This loads the fluorescence profile into the Fitting Display

window.



6. In the control window, after "No. of components", hit the right arrow once, changing the number to 1. This selects mono-exponential fitting.

7. In the Fitting Display window, click with the mouse inside the black area near the left border. You will see a cursor (vertical yellow line). It indicates the start of the fitting range. Drag it to a position of around 4000 (= 4 ns). Do the same near the right border and drag the violet cursor (indicating the end of the fitting range) to around 16000 (= 16 ns). (You don't need to hit these values precisely.)



8. Press the "Initial Guess" button (at the lower left corner of the control window).

9. Press the "Do Fitting" button. The fitting engine starts running and should terminate within a few seconds.

10. You can now read the approximate value of the fluorescence lifetime in the right column behind "Tau 0" (see picture). It should be around 5.3 ns. The fitting curve can be seen in red color in the display window.

11.  Terminate the fitting session by pushing the Close button in the upper right corner

C. Lifetime Fitting - The Basics

This chapter describes the "*what*" of the fitting job (extracting the relevant information from the measurement data) as well as the basic "*how*" to achieve that goal (the principle of the fitting algorithm).

If you are already familiar with lifetime fitting, you may skip this chapter.

1. The Task: Extracting Lifetimes from Streak Data

This section describes the basic task of the fitting job and the various complications that need to be considered. These are the presence of noise, multiple components, limited resolution and distortion of the excitation and detection system, and others.

a. Exponential Decay

In its most simple (unrealistically simplified, we should say) form, the fluorescence decay curve is a simple mono-exponential curve, without any noise.

This is not a textbook about physics or chemistry, but let us briefly consider the origin of the exponential decay law. As a most simple case, suppose we have a physical system with two energy levels (E_1, E_2). Suppose further that the system is originally in its ground state (E_1) and is "pumped" at time t_0 to the state E_2 by an (idealized) infinitely short light pulse (a "delta" pulse). This means, at time t_0 the system will undergo an instantaneous transition from E_1 to E_2. After it is in state E_2, it will have a constant probability per time unit to spontaneously decay back to the ground state. Thus, this decay will happen with a *constant rate* which we call the *transition rate r*. Let's assume that the total number of elements (e.g. atoms, molecules, and so on) in the energy state E_2 is N, then the constant rate law says that at each unit time interval dt a *constant fraction* of N will return to the ground state. If we denote that fraction by dN, the law simply states that (under the consideration that the decay *de*-creases the number of available elements and under the consideration that N=0 for $t<t_0$):

$$dN = \begin{cases} -r \cdot N \cdot dt & ; t > t_0 \\ 0 & ; t \le t_0 \end{cases} \qquad \text{(C 1)}$$

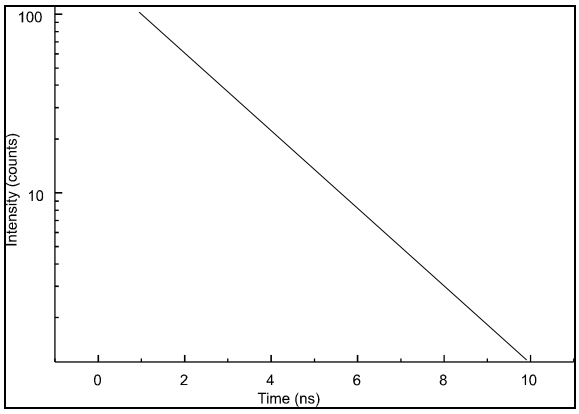
Integration yields:

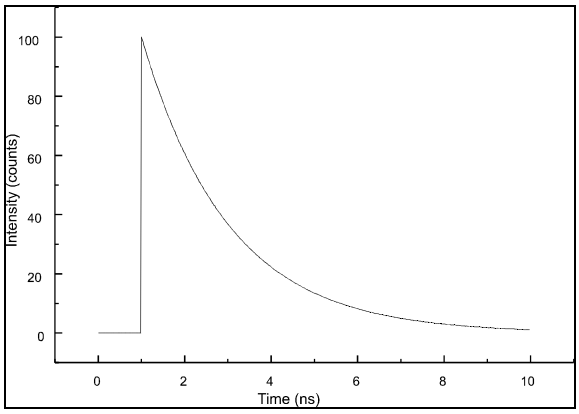
$$N(t) = \begin{cases} N_0 \cdot \exp(-rt) & ; t > t_0 \\ 0 & ; t \le t_0 \end{cases} \qquad \text{(C 2)}$$

This is the well-known *exponential decay law*. Since each decay is associated with the emission of a single fluorescence photon, the fluorescence light intensity is equally given by:

$$I(t) = \begin{cases} a \cdot \exp(-t / \tau) & ; t > t_0 \\ 0 & ; t \le t_0 \end{cases} \qquad \text{(C 3)}$$

Here, we have introduced the *fluorescence lifetime* τ ("tau") which is defined as the inverse of the decay constant ($\tau = 1/r$).





The graph of eq. (C3) is shown below (with a linear scale on the left and a logarithmic one on the right).

An analysis (determination of τ) in this oversimplified case could be simply performed by merely reading the slope of the curve in the logarithmic plot.

b. Noise

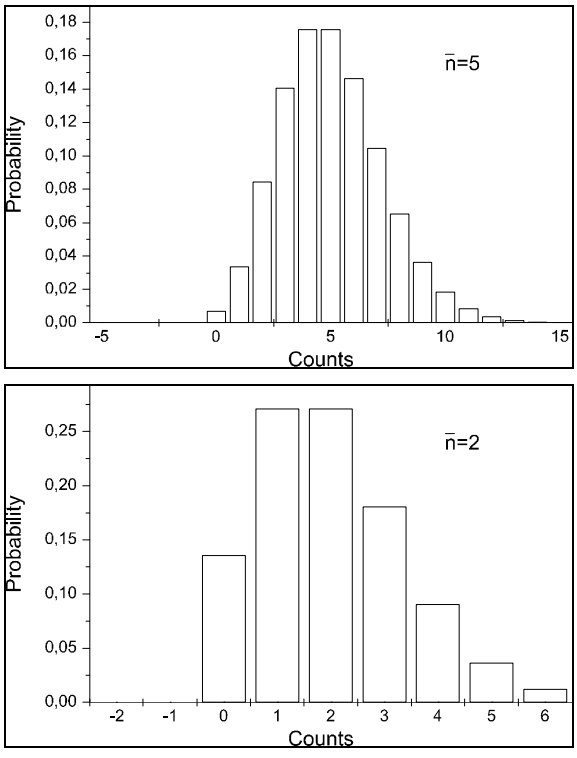
The above case was unnaturally oversimplified mainly because we did not consider any noise. In this section, we will consider the *photon shot noise* (and ignore any other potential noise sources). The photon shot noise is simply the effect of the limited number of photons contributing to each data point in the decay curve, and of the intrinsic statistical variation of this photon count number. In the ideal case, it is generally assumed that the photon statistics is Poissonian. (For theoretical accounts why this assumption is justified, refer to textbooks.)

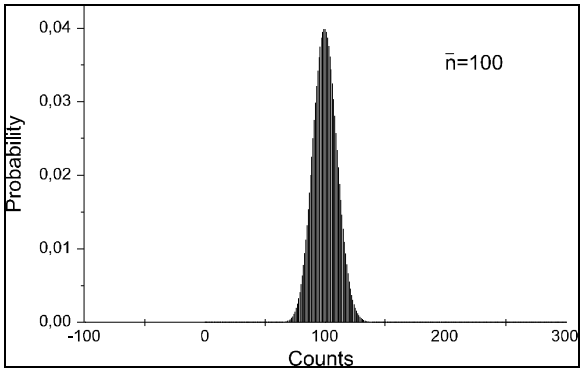
The *Poissonian statistics* is given by

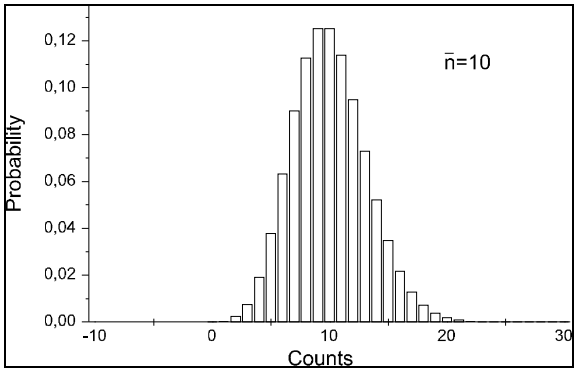
$$P(n) = \frac{\overline{n}^n}{n!} e^{-\overline{n}} \quad \text{(C 4)}$$

where P(n) is the probability that a certain photon number n will be found in a channel which has a mean photon number $\overline{n}$.

The four plots below show the statistics for several different numbers of $\overline{n}$. As we can see, as $\overline{n}$ gets larger, the average deviations of n from $\overline{n}$ (which are the origin of the shot noise) increase, but the *ratio* of the average deviations (= the "noise") to the mean value $\overline{n}$ (= the "signal") *de*creases.





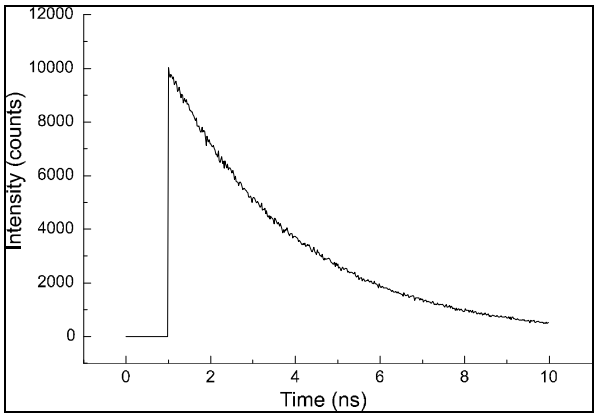


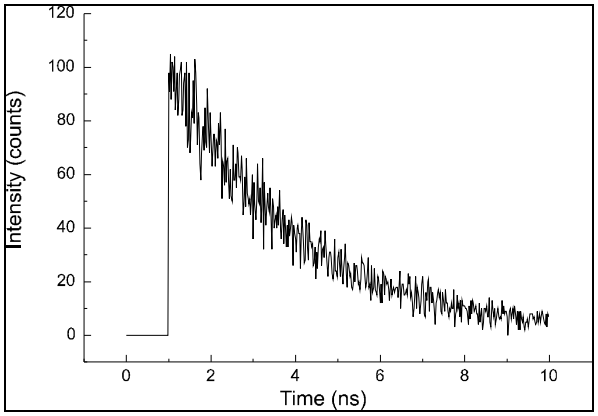
Actually, it can be shown that in case of Poissonian statistics the mean square deviation (= variance) is equal to the mean $\overline{n}$, so if we define the noise strength as the "root mean square" (= standard deviation), we obtain:

$$Noise = \sqrt{\overline{n}} \qquad \text{(C 5)}$$

$$S/N\ ratio = \frac{\overline{n}}{\sqrt{\overline{n}}} = \sqrt{\overline{n}} \qquad \text{(C 6)}$$

and





The two graphs below show the same signal as on the previous page, but this time realistic noise has been added. The graph on the left shows the case for a signal of 100 peak counts, while the graph on the right shows the case of 10.000 peak counts with a 10-fold improvement in signal-to-noise ratio.

If we want to determine the fluorescence lifetime from a curve like the above, obviously simply "measuring the slope from the graph" is not a useful strategy anymore. What we have to do is data *fitting* now. The aim of the fitting procedure will be the determination of the parameters τ and a in such a way that the measured data are resembled best. While it is intuitively clear what a good fit should look like, we will still need a quantitative measure what "best" shall mean and how the quality of a fit can be determined. This issue will be covered in section C2a later.

Note that the fit process must determine two numbers here, the lifetime τ and the amplitude a, although the researcher will typically be interested only in the lifetime itself. (See also next section.)

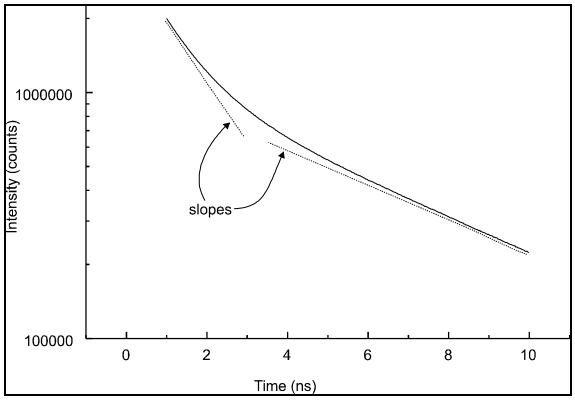
c. Multiple Components

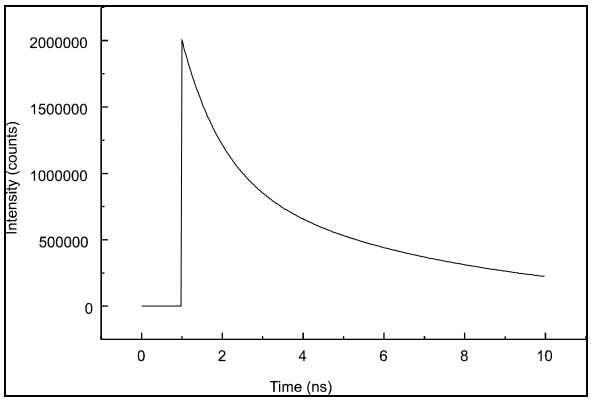
Up to now, we only considered one possible transition in a simple two-level system. In practice, however, there are frequently additional transitions involved. These may be additional levels and transition processes in the sample under study itself, or radiation generated in other materials such as the solvent, impurities, and so on. In the first case, the determination of the involved lifetimes may be of generic interest to the researcher, while in the others cases the extra signal is considered as artifact which shall be finally ignored in the interpretation of the sample's data. It is important to understand, however, that even in the latter case the existence of the additional decay components must be taken into account during the fitting process if the sample's lifetime shall be obtained correctly.

Let us consider the case with two components (both will be simultaneously excited at time t_0, of course):

$$I(t) = a_1 \cdot \exp(-t / \tau_1) + a_2 \cdot \exp(-t / \tau_2) \qquad \text{(C 7)}$$

A sample graph may look like this (linear and log scales):





As you will notice, in this example the two contributions from the two lifetime components are visible by bare eye as two different tangential slopes in the log plot. This will not be so easy when there are more components, when there is higher noise, and so on.

The generalized version of eq. (C7) is given by:

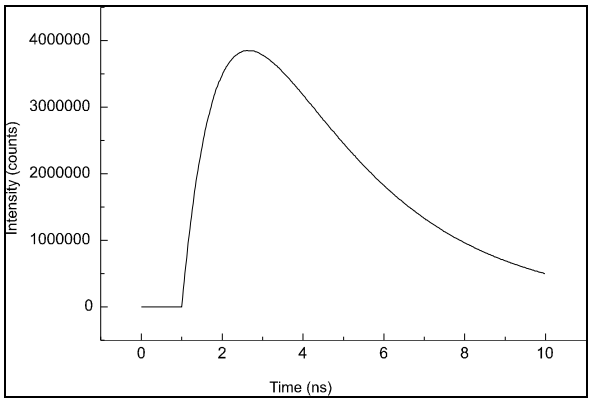
$$I(t) = F(t - t_0) := \begin{cases} \sum_{i=1}^{K} a_i \cdot \exp[-(t - t_0) / \tau_i] & ; t > t_0 \\ 0 & ; t \le t_0 \end{cases} \quad \text{(C 8)}$$

Notice that here we have defined the generalized decay function F which will help us to keep the following formulas shorter. Since it is this function which serves the fitting engine as the theoretical model of the physical process, this function is called the *theory function* of the fit. The parameters a_i are called the *amplitudes* of the components.

Theoretically, the number of components (K) could be unlimited, but practically cases where K is larger than 3 or 4 will be very hard or impossible to treat well.

So, the extended task is to find not only one lifetime τ, but to find K lifetimes τ_1, τ_2, ...,τ_K. In addition, note the following: given a fixed set of those lifetimes τ_i, there are still many (actually infinitely many) ways how these can be combined in terms of their *relative* amplitudes. Accordingly, the fitting routine will have to determine a set of 2K numbers, namely τ_1, τ_2, ..., τ_K, a_1, a_2, ..., a_K.

Recall that also in the mono-exponential case (section b above), we had to determine the amplitude a, but this was just a necessity to compute the fitting curve and we were not particularly interested in that amplitude value itself. The reason is that there is no physical relevant information in the absolute value of these amplitudes (they scale with arbitrary experimental conditions such as the amount of light, the length of the measurement period, and so on). But their *ratio does* contain valuable information since it reflects the amount of matter in the different energy levels, the relative yield of their excitation, and so on.



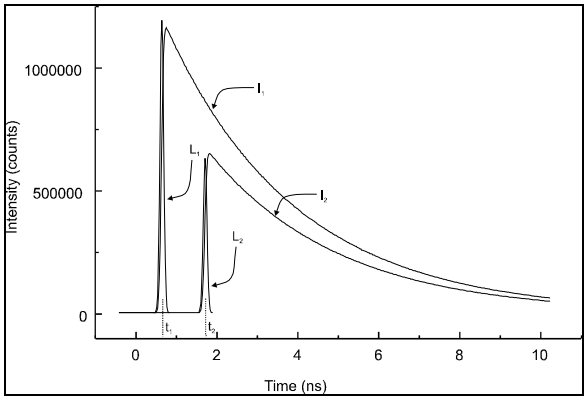
Finally, we need to mention that fluorescence amplitudes may also be *negative*. Look at the graph below. It shows a decay profile with two components, a 3 ns component with positive and a 1 ns component with negative amplitude. The latter is causing the rising part at the leading part of the curve.

Eq. (C8) still describes this kind of curve well (since nothing prohibits us to insert negative amplitudes there), but our derivation of the formula (in section a) is not capable to explain their origin. The reason is that in our simplified derivation we only considered two-level systems. However, if there is a third level decaying with a lifetime t_1 into the second level that in turn decays to the ground level with a lifetime t_2, then the fluorescence seen from the second level would show the lifetime t_1 as a negative component, because the second level is "pumped" from the third level with that rate. (So, we could understand the behavior as caused by a pumping pulse which is not delta-shaped as we assumed up to now but exponential-shaped. This way of looking at it will smoothly lead us to the next section.)

Note, however, that the *sum* of all amplitudes must always be positive; otherwise negative intensities around t_0 would result, which would not correspond to physical reality. Also note that negative lifetimes can never occur, of course.

d. Convolution with IRF

Now we will consider another important deviation of real data from our idealized considerations so far. Up to now, we assumed that the exciting light source has an infinitely short pulse width, i.e. we could treat it as an ideal delta pulse. This was a condition to obtain the clean, simple exponential decay law in eq. (C8). In practice, however, the laser pulse width is always finite and will have a certain shape. How does this influence the shape of the decay curve?



In order to understand this, let us first assume another case. Consider that we do not excite with a single delta pulse, but with a light source which emits a short *train* of "near-delta" pulses. Each of these pulses will do some excitation of the sample (a case with two pulses is sketched on the left).

We (in a mathematically informal way) symbolize the pumping contribution of each pulse by $L_j(t_j)x\Delta t_j$ (the Δt_j considered to be "very small") and obtain the contribution of each pulse to the fluorescence signal as:

$$I_j(t) = L_j \cdot F(t - t_j) \cdot \Delta t_j \qquad \text{(C 9)}$$

(remember the definition of F in eq. (C8)).

Obviously, in order to obtain the total decay curve we need to sum over these contributions:

$$I(t) = \sum_j L_j(t_j) \cdot F(t - t_j) \cdot \Delta t_j \quad \text{(C 10)}$$

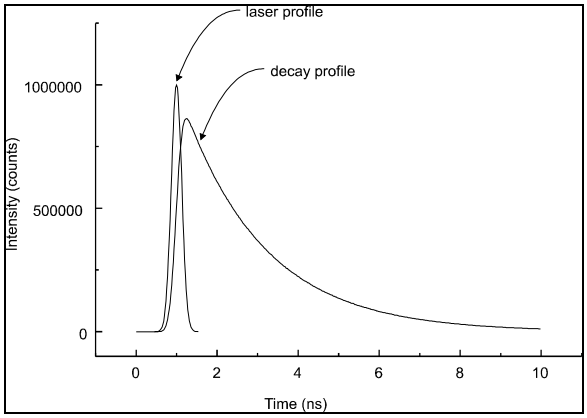
Now we imagine the number of sub-pulses go infinite while their width Δt_j approaching zero. We assume that the shape of the envelope of all these pulses is given by the (now continuous) function L(t). Accordingly, the sum in eq. (C10) will be transformed into an integral of the form

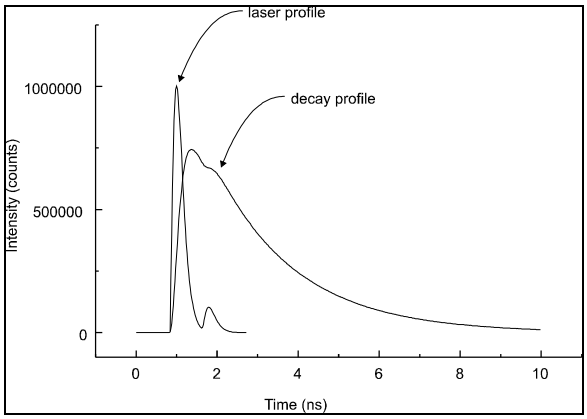
$$I(t) = \int_{-\infty}^{t} L(t') \cdot F(t - t') \cdot dt' \quad \text{(C 11)}$$

Sometimes, this integral relation is further abbreviated by a more compact formalism as:

$$I(t) = L(t) \otimes F(t) \qquad \text{(C 12)}$$

This is called the *convolution integral*. An example is shown in the left graph below. It shows the same decay as in paragraph c, but this time excited by a laser pulse of finite width (which is about 30% of the lifetime constant in this example). Notice that the rising edge and the peak portion of the fluorescence curve are somewhat flattened. An even more striking example is shown in the graph on the right, where the laser pulse is not only asymmetric but also has a strong afterpulse, causing a severe distortion of the decay curve.





So, the additional task of the fitting process will be: Given the measured intensity curve I(t) and given the curve of the laser pulse L(t) , re-calculate the sample's decay curve F(t) (which will be subject to fitting then).

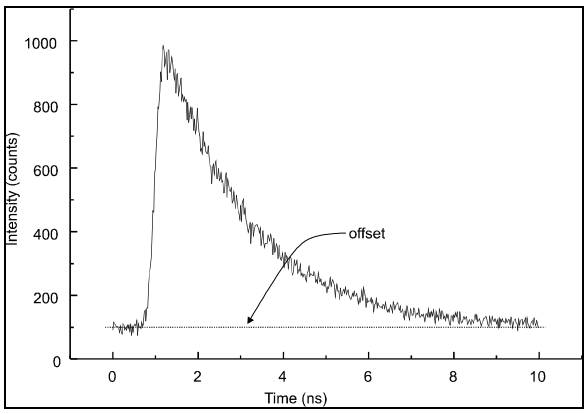
This process of inversion of eq. (C11) or (C12) is called *de-convolution*, i.e. reversing the effect of the convolution.

Up to now, we only considered the effect of the laser's pulse shape. In practice, however, there will be also other sources of convolution. Consider for example that the temporal resolution of the detection system is limited, too. (The temporal resolution of the streak camera system is high but not infinite, the emission may be temporally broadened by time dispersion in a spectrograph, and so on.) This kind of signal broadening will lead to additional convolution.

So, strictly speaking, we need to treat various contributions of convolution, what we could symbolize as $I(t) = L_1(t) \otimes L_2(t) \otimes L_3(t) \otimes \cdots \otimes F(t)$. However, let us consider from where we will derive these functions L_i in practice: they will be *measured* (under experimental conditions which should resemble those during the sample measurement as close as possible). But they are usually not measured individually, but only altogether. This means, what we experimentally determine is the "total convolution" $L_1(t) \otimes L_2(t) \otimes L_3(t) \otimes \cdots \otimes L_m(t)$. Hence we just can identify this total convolution with our L(t) and continue to use eq. (C11) and (C12) without modification.

This total convolution term is the "*instrument response function*" (IRF) of the total measuring system (including the excitation and detection part). By using the IRF in this manner, we don't need to care about what part of the measuring system is responsible for what feature of the response behavior.

e. Offset



Under some experimental conditions, it may happen that there is a *constant intensity offset* on the data. Such offset signal may originate from static stray light (room light leakage, for example), dark current of the detector, or other sources.

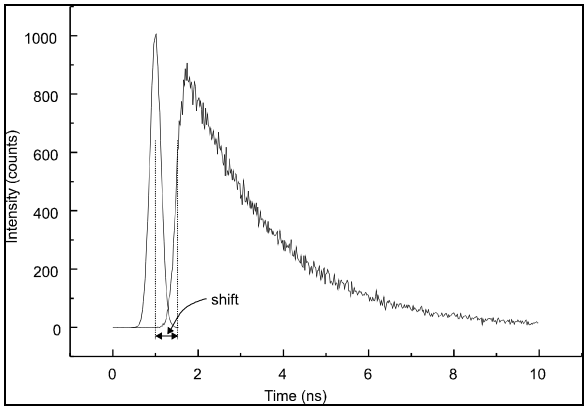
In order to cope with this analytically, we need to introduce a constant amplitude into our description. Accordingly the theory function F (see eq. (C8)) now becomes

$$F(t-t_0)=\begin{cases} a_0+\sum_{i=1}^{K} a_i\cdot\exp[-(t-t_0)/\tau_i] & ;t>t_0 \\ 0 & ;t\le t_0 \end{cases} \qquad \text{(C 13)}$$

If deconvolution is needed, this new theory function must be used as F(t-t') in eq. (C11).

The determination of a_0 will be an additional task of the fitting procedure.

f. Shift



As outlined in paragraph d, for obtaining a full set of data for the de-convoluting analysis, we need to record two curves, the decay curve and the laser curve. This yields the possibility that there may be a *shift on the time axis* between the two curves. Obviously, such a shift will be an artifact and not a property of the sample. Nevertheless, it is frequently observed. A typical cause for such a shift is the dispersion in a spectrograph or an optical fiber, causing the laser pulse to arrive on the photocathode at a different time than the fluorescence signal (at other wavelengths). Another reason - if the laser and the decay had not been recorded simultaneously - may be a true timing shift between the two measurements, which may be caused for example by slow drifts in involved electronic components.

Even though it is an artifact and we may not be interested in the shift value itself, a correct determination of the shift turns out to be essential to obtain a good fit.

In the deconvolution equation (C12), we can easily account for the shift by adding the parameter t_{shift}:

$$I(t) = L(t - t_{shift}) \otimes F(t) \qquad \text{(C 14)}$$

h. Summary

Eq.(C14) is our final generalized theory function. In full, expanded form it looks like:

$$I(t) = \int_{-\infty}^{t} L(t'-t_{shift}) \cdot F(t-t') \cdot dt' \qquad \text{(C 15)}$$

, where $$F(t-t') = \begin{cases} a_0 + \sum_{i=1}^{K} a_i \cdot \exp[-(t-t')/\tau_i] & ; t > t' \\ 0 & ; t \le t' \end{cases} \qquad \text{(C 16)}$$

The above includes the convolution with the IRF. The simplified case without convolution can be re-derived by assuming that the IRF is an ideal "delta" function:

$$L(t') = \delta(t'-t_0) \quad \Rightarrow \quad I(t) = \int_{-\infty}^{t} L(t') \cdot F(t-t') \cdot dt' = F(t-t_0) \qquad \text{(C 17)}$$

(This is the same as eq. (C8)).

Notice that we ended up with quite a lot of parameters the fitting process needs to extract from the data. Given that we want to do a fit with K fluorescent components, there will be 2K+2 free parameters (τ_1,τ_2, ..., τ_K, a_1,a_2,...,a_K,a_0,t_{shift}).

Now, let's turn to section 2 and see *how* this will be achieved.

2. The Method: The Fitting Algorithm

After we defined in section 1 *what* we want to do, this section describes the basic method *how* this is achieved. Section a will explain why "least-square fitting" is a useful approach and will also give some important remarks about statistics. Section b gives an impressionist view of the fitting method, while section c cites some real mathematics.

a. Least-square data fitting

The job of the fitting engine is: find the parameters of a pre-defined theory function so that the resultant function gives a "best match" for the measured data (which are noisy). The standard approach to such kind of a problem is: *least-square approximation*. The subject of this section is: what does this mean? What is a "best match" and why does the least-square method approximate it?

Let us consider the *deviations* of the measured data from our arbitrary - perhaps preliminary - theory function. For each channel on the time axis, we define this deviation simply as the difference

$$R_i = M_i - C_i \qquad \text{(C18)}$$

where M_i are the measured and C_i the calculated data points. These differences are called *residuals*.

Since the statistical significance of the channels with larger count numbers is higher due to their lower relative noise, the residuals should be *weighted by their standard deviation*. If we denote the standard deviation of each measuring point by σ_i, we obtain:

$$R_{wi} = \frac{M_i - C_i}{\sigma_i} \qquad \text{(C19)}$$

These numbers are called the *weighted residuals*.

Least-square approximation means that the *sum of the squares of these weighted residuals shall be minimized*. It is obvious that if this sum were zero the fit would be perfect since in this case all differences must be zero. Actually, the sum is always greater than zero, since there will always be noise on the data.

So far, so clear. But, why do we use the "sum of squares" and not the "sum of absolute differences" or any other measure? Why does this "sum of squares" method give the "best match" in a useful sense? The next half page or so gives a rough answer.

The crucial step is that we *assume* that our measured data M_i vary randomly around the "real" (theoretical) value C_i *in a Gaussian manner*. (If you wonder why this assumption is justified, refer to the end of this section.) A *Gaussian* (or "*normal*") *probability distribution* is given by:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left[-\frac{(x-\bar{x})^2}{2\sigma^2} \right] \qquad \text{(C20)}$$

This is a *probability density*, defined for a continuously variable x. The *probability* P that the value x will be found between some values a and b is given by:

$$P(a < x < b) = \int_a^b p(x')dx' \qquad \text{(C21)}$$

If we have discreet data (as it is the case when counting photons) eq. (C21) will still be valid assuming that the data are sufficiently dense. In this case, the integral will be replaced by a sum. If we are interested in the probability that x is equal to one specific (discreet) value a, the integral will range from a to a+da, while we set da equal to 1. So, we simply obtain $P(x = a) = p(a)$.

Note that the Gauss distribution is peaked around the mean value $\bar{x}$, and its standard variation $\sqrt{mean((x-\bar{x})^2)}$ is equal to σ .

So, if the data are Gaussian, then the probability of obtaining just the set M_i as a result of a measurement (given that the "true" value is correctly described by the chosen theory function, i.e. the set C_i) is the product of the probabilities of all the data points, i.e.:

$$P \propto \prod_{i=n_1}^{n_2} \left\{ \exp\left[-\frac{1}{2}\left(\frac{M_i - C_i}{\sigma_i} \right)^2 \right] \right\} \qquad \text{(C22)}$$

(n_1 and n_2 denote the fitting range.)

What we want to do is maximizing this probability P. If we take the logarithm of eq. (C21), we obtain

$$\ln P = const. - \frac{1}{2} \cdot \sum_{i=1}^{N} \frac{\left(M_i - C_i \right)^2}{\sigma_i^2} \qquad \text{(C23)}$$

This means that maximizing P is equivalent to minimizing the sum in the above formula. This sum is exactly the sum of the squares of the weighted residuals, mentioned above.

We now give it a name. The above-mentioned sum of squares is conventionally called the χ^2 ("*chi-square*") of the approximation. Since we assume that the standard deviation is equal to the square root of the count number (see eq. (C5)), we obtain:

$$\chi^2 = \sum_{i=n1}^{n2} \frac{(M_i - C_i)^2}{M_i} \qquad \text{(C24)}$$

Note that the values C_i are the values of our chosen theory function which depends on a set of M free parameters (by "free" we mean those parameters which are allowed to change during the fit, i.e. which have not been "fixed" by the

user). Here, we denote these parameters as $p_1,...,p_M$. (If you think of our actual theory function given in eq. (C15/16), obviously the p_i refer to a subset of the parameters a_i, τ_i, t_{shift}, a_0, namely those that are not fixed.)

So, the task of the fitting procedure is to find those parameters p_i for which the above value χ^2 is minimized. The next two sections (C2b and c) will tell in a bit more detail *how* the fitting engine does this. Before doing so, however, we need to mention a few other important points.

First, we need to give one further remark and definition. As a detailed statistical analysis (which we don't report here) shows, the probability distribution of different values of χ^2 behaves like a so-called "chi-square distribution" (a term with a precise definition in statistics science) with N-M degrees of freedom, where N is the number of data points ($N=n_2-n_1+1$) and M is the number of free parameters. As an abbreviation we define ν=N-M. It can be shown that (for sufficiently large ν) the distribution of χ^2 becomes normally (Gaussian) distributed with a mean value of ν.

Due to the above reason, we define the "*reduced chi-square*" as

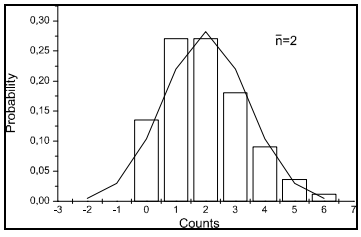
$$\chi_\nu^2 = \frac{1}{\nu} \cdot \chi^2 \qquad \text{(C25)}$$

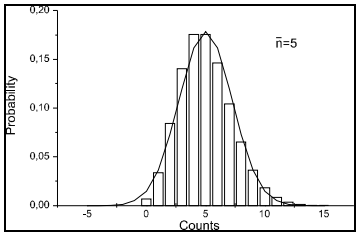
and *expect it to converge against 1* in case of a good fit and a sufficiently large ν. In most cases in this document, when we say "chi-square" we actually mean the reduced chi-square since it is exactly this number which TA-Fit outputs to the user.

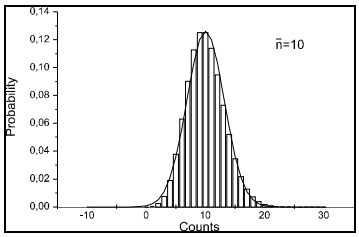
Finally, we need to return to an important consideration: *why* was it justified to assume that the statistics of the measuring data is represented by the Gaussian distribution? Actually, the Gaussian distribution is assumed in a huge class of physical phenomena otherwise completely unrelated. In many cases, the Gaussian is simply used "because it works", an *assumption* which could be empirically tested (but often is not). In some cases, there may be theoretical reasons from which it can be *deduced* that the statistics should be Gaussian (e.g. in idealized cases in thermodynamics). In other cases, the causal processes leading to the statistics are not sufficiently known in detail. The interesting feature of the Gaussian distribution is that it approximates the reality surprisingly well even in a great variety of those latter cases.

How sure, then, can we be in our specific case (fluorescence data taken with a streak camera) that the assumption of a Gaussian distribution is a reasonable one? For an answer on this question, please refer to Chapter F8.

The Gaussian statistics (which the fitting engine is using) and the Poissonian statistics (which is the supposed ideal behavior of counted photons) tend to converge against each other in case of large count numbers. This fact can be proven mathematically. We will not reproduce such formal proof here (refer to math textbooks), but we just illustrate the subject by showing some graphs.







These graphs are the same as those in chapter C1b, showing the Poissonian distributions (the histogram graph) for some different mean values (2, 5, and 10 counts). In addition, also the Gaussian curve is plotted now, adjusted for the same mean value $\bar{n}$ and with a standard deviation set to $\sqrt{\bar{n}}$ (according to Poisson statistics). As we can see, the match is quite good, even for these low count numbers. (We don't show the case with $\bar{n}$ =100, since the deviation between the two graphs would be invisible there.)

Even though the match between these two statistics is very fine in most cases, we should be aware that *some systematic errors* must be expected in case of extremely low count numbers.

Summary

We can say, the fitting engine works with three sources of input:

the measured data (decay data, possibly also laser data)

the theory function with its parameters

the characteristics of the statistics of the data

The first source is always freely specified by the user. Concerning the second source, the form of the theory function is fixed (given in eq. (C15/16)), but it is up to the user to specify the number of parameters and which are free or fixed. The third source, however, is always fixed. TA-Fit always assumes a Gaussian statistics, and it also assumes that the weighting factors are equal to the inverse of the square root of the intensity (a criterion derived from Poissonian counting statistics).

b. Finding the way in a multi-dimensional landscape

We do not talk about details of the fitting algorithm here. Instead, we just want to characterize the general approach, namely *iterative optimization in a multi-dimensional space*.

In the previous section, we said that the task of TA-Fit will be to "calculate" the set of parameters of the theory function in such a way that the global error (which we identified as chi-square) will be minimized. In section C1d, we also mentioned that TA-Fit will need to undo the convolution, i.e. to perform a "de-convolution" in some way.

We now have to correct these statements slightly. It turns out (a rough mathematical explanation is given in the next section) that it is *not* possible to "calculate" the parameters in one step. We need to apply some *iterative algorithm*, which starts with a certain parameter set, inserts it into the theory function, calculates the chi-square, and tries to improve the parameters, converging against the optimal set. So, although we sometimes use the word "de-convolution" in this document, "iterative re-convolution" would be a better description of what is actually done.

Of course, there must be a criterion about when to stop the iteration (for example, that the relative variation of the parameters should fall below a predefined threshold), but that's the easy part. The real difficult part is to find out in each cycle *which* parameter to change, into *what* direction, and by *how much*.

As a help for our imagination, let us consider an analogy in two dimensions. Imagine a two-dimensional plane spawned by two parameters, x and y. Above this plane, there is the (three-dimensional) "chi-square landscape", with hills, valleys, and so forth. The size of the landscape is infinite. We start at some arbitrary point in this landscape (defined by our start values for the parameters) and try to move around in such a way that we find the deepest spot (i.e. the coordinates of x and y where chi-square is minimum).

The difficulty is that all information available is "local". That means that at our current position we can only measure the actual height, its slope, its curvature, and so on, but we do not have a priori knowledge of the landscape on a greater scale.

In addition, our strategy must be predetermined and cannot be adjusted on the fly. This means, we are using an "algorithm which cannot learn". You could visualize the scenario as that of a small robot moving around on a real landscape, seeking for the deepest spot.

Obviously, programming such a robot is not an easy task. The general danger is to get trapped in some small valley or hole that is not the deepest, i.e. in a local minimum. Even with a sophisticated algorithm, it cannot be completely excluded that the robot might be trapped if the landscape is very rough and irregular.

So, we can summarize our problem as follows: an unknown (perhaps weird) landscape (with a shape which depends not only on the data but also on the fitting range), an arbitrary starting position, a robotic search engine which can follow only strict rules.

This sounds already difficult enough, but the actual task of TA-Fit is even worse. Our imaginary robot was in a lucky position, since it had to cope with only two dimensions. The landscape of fluorescence fitting is much richer, maybe four-dimensional, six-dimensional, or even worse. So many directions to go!

An additional requirement is that the search shall be fast.

The above is our excuse that we did not create "the perfect" fitting engine. Actually, we are sure that no such thing can exist. All one can do is: design the general search method, put some heuristics into it, test it, refine it, test it again, and so on, and optimize the code so that it behaves well in most typical situations while calculation times stay reasonable.

c. Rough sketch of the fitting mathematics

In this section, we will give an overview how the minimization procedure is actually performed mathematically. Users who are not interested in these technical details can safely skip this section. On the other hand, readers who are interested in full details should be aware that this document is not a textbook on analysis or numerical mathematics; for more detailed accounts please refer to the literature. (By the way, please don't ask us for source code; we don't publish it.)

The formal problem of fitting is to minimize chi-square which is given by eq.(C24). Recall that the M_i are the measured data while the C_i are the values of the theory function, which depends on a set of parameters p_i (i=1,..,M). These parameters shall be adjusted so that χ^2 is minimized. For brevity, let us denote the theory function simply by f, so we have:

$$\chi^2 = \sum_{i=1}^{N} \frac{1}{M_i} (M_i - f(t_i; p_1, \ldots, p_M))^2 \qquad \text{(C26)}$$

The condition for a minimum of χ^2 is:

$$\frac{\partial \chi^2}{\partial p_k} = 0 \qquad \text{for all k = 1,...,M} \qquad \text{(C27)}$$

By substituting (C26) into (C27), one can obtain after a few calculations:

$$\sum_{i=1}^{N} \frac{(M_i - f(t_i))}{M_i} \cdot \frac{\partial f}{\partial p_k} = 0 \qquad \text{for all k = 1,...,M (C28)}$$

Note that (C28) is a set of M equations for the M parameters p_k.

How can the equation system (C28) be solved? This would be easy if the theory function f would depend on the parameters p_k *linearly*. If f were given by $f(t) = \sum_{k=1}^{M} p_k \cdot g_k(t)$, then (C28) would become $\sum_{i=1}^{N} \frac{1}{M_i} \cdot \left[M_i - \sum_{j=1}^{M} p_j \cdot g_j(t_i) \right] \cdot g_k(t_i) = 0$. The latter is a linear system of equations for the parameters p_k, so that the usual numerical methods for linear systems (e.g. Gauss-Jordan elimination etc.) could be applied.

However, in our case the theory function's dependence on the parameters p_k is far from linear (consult eq. (C15/16)). The solution to this problem is the *Taylor series expansion* of the theory function. The price that must be paid is that the solution cannot be calculated by one step, but must be *iterated*.

An analogy in the one-dimensional case (where f depends on only one parameter), is the well-known Newton's method for finding the root of a function. If the function were linear (i.e. a straight line), its crossing with the zero axis could simply be calculated by solving the equation (similar as in the above linear case, where we would solve a set of linear equations). In more complicated cases of nonlinear functions which - in the general case - cannot be

resolved for their free parameter, Newton's method offers a solution at least in the sense that it gives an *algorithm* for obtaining ever better parameter values in an iterative way.

Recall that in the one-dimensional case (where f depends on only one parameter), the Taylor series is given by $f(p+\delta p) = \sum_{k=0}^{\infty} \frac{f^{(k)}(p)}{k!} \cdot (\delta p)^k$, where $f^{(k)}$ denotes the k-th derivative of f with respect to p. If we omit all terms of higher order than linear, we obtain $f(p+\delta p) = f(p) + \delta p \cdot f'(p)$. This yields Newton's method directly, if we set the equation to zero and interpret dp as the increment from one iteration to the next (i.e. $p_{next} = p_{current} + dp$).

In the multi-dimensional case, we must use a vector formalism where $\vec{p}$ denotes the vector $(p_1,...,p_M)$ and so forth. If we limit the expansion up to the second-order term, the Taylor series yields:

$$f(\vec{p}+\delta\vec{p}) = f(\vec{p}) + \vec{\nabla} f(\vec{p}) \bullet \delta\vec{p} + \delta\vec{p} \bullet H(f)(\vec{p}) \bullet \delta\vec{p} + o(|\delta p|^2) \qquad \text{(C29)}$$

where $\vec{\nabla} f = (\frac{\partial f}{\partial p_1},...,\frac{\partial f}{\partial p_M})$ is the gradient vector,

and $H_{i,j}(f) = \frac{\partial^2 f}{\partial p_i \partial p_j}; i,j = 1,...,M$ is the Hessian matrix.

(The last term in (C29) symbolizes that the rest converges to zero faster that the square of δp.)

Similar as in case of Newton's method, eq.(C29) can be used to obtain ever better values for $\vec{p}$ by calculating the "correcting terms" $\delta\vec{p}$ and iterating the procedure with $\vec{p}_{next} = \vec{p}_{current} + \delta\vec{p}$. Note that each step involves some matrix inversions and the like in order to obtain $\delta\vec{p}$.

A difficulty with the method is that although the quadratic term delivers precise values when $\vec{p}$ is already close to the optimum (where f can be well approximated by quadratic functions), it may deliver quite erratic values if $\vec{p}$ is still far away from the optimum. So, in practice a method is employed which is using only the linear term (which is a descent in the steepest direction) while the iteration is still far away from the minimum, and which will later switch to using the Hessian term also. Actually, this switch is not applied suddenly but via a smooth transition using a regulating parameter λ that is dynamically adapted. (It is decreased or increased, depending on whether χ^2 had been improved or not since the previous iteration; a failure of improvement may indicate an overshooting of the descent over the minimum valley). This adaptive method is described in the literature as the *Levenberg-Marquardt algorithm*. Advantages of the algorithm are that it yields good estimates for the initial step size and that this initial step size may be relatively large, yielding a good speed of convergence

The fitting engine of TA-Fit is using a form of Levenberg-Marquardt algorithm (with some modifications and heuristics added).

We will not dwell upon more details here. For those interested further, we recommend the literature.

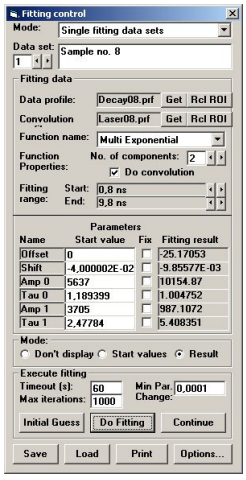
D. How to Use the Fitting Windows

This chapter describes each window of the TA-Fit module, its controls, and how to use them. Most of the stuff here should be quite obvious, and perhaps you might refer to this chapter only for reference if something seems unclear.

When TA-Fit is activated via the HPD-TA menu Analysis/Fitting two windows open: the Fitting Control window and the Fitting Display window. These are the main windows of TA-Fit and will be described first.

1. The Fitting Control Window

The Fitting Control window is the window in which the user defines all the parameters relevant for the fitting job. This window must always be kept open during the fitting session. Closing the window terminates the session.



a. Where the data comes from

You use the small "Get" buttons to define from where TA-Fit will get its input data, i.e. the decay data and - if required - the laser data.

This opens a small window with three choices:

The first option ("Get from file") lets you load a previously saved HPD-TA profile (.PRF) file from disk. You simply choose the file from a dialog window. You don't need to specify the "direction" in this case.

The other two options are for extracting new profiles from a current image. Here, you need to define an region-of-interest (ROI) in the streak image in the usual way. (If you are not familiar with this basic technique, please refer to the HPD-TA manual or help file). You also need to specify the direction in which to extract the data. Since we are exclusively dealing with lifetime fitting here, this direction should always be the time axis (normally vertical). If you choose "Auto update from image", the data in TA-Fit will be automatically updated after a new image has been acquired or loaded from disk, sampled always at the same ROI positions.

If you want to re-activate a previously used ROI associated with either the decay or the excitation profile, you can use the "Recall ROI" buttons. They work in the same way as the corresponding button in HPD-TA's Profile Control window.

Please note that the definition of the decay data source is mandatory, while the source in the "Convolution profile" edit box is optional and needed only if the fitting shall be done with Convolution (see below).

If Convolution is off, the source in the Convolution profile box might be displayed in the Fitting Display window, but it will not be used in the fitting calculations.

b. Type of theory function

With this combo box you select the theory function to be used by the fitting engine. You will find several choices here, including "Multi Exponential", "Gaussian", "Lorentzian", and so on. *Throughout this document, we are concerned with fluorescence lifetime fitting only, so we will ignore those other choices. So please always select "Multi Exponential" here.*

Note: the other choices are there mainly because it was easy to implement them. The fitting engine is designed in such a way that its core is independent of the theory function. This will allow HAMAMATSU to develop extensions of TA-Fit in future versions. In the current version of TA-Fit, however, those other theory functions do not serve any specific purpose. Of course, you are free to experiment with them if you like. (Perhaps the "Gaussian" may be of some limited value for you if you want to measure the pulse width of your laser, for example.) We believe these

additional functions work correctly, but we did not test them extensively. So, please consider them as a bonus function of the TA-Fit and use them at your own risk. (For the definitions of these theory functions, please refer to the Apendix.)

c. Number of fluorescent components

Please use the spin buttons (the two arrows) to specify the number of components (in chapter C the number was termed K) of the fitting function. In most cases, this number will be set to 1, 2, or 3. The maximum number that can be specified here is 8 but this will surely never be reached in practice!

Note that the central part of the Fitting Control window will be dynamically changed according to the number of components you specify. For each new component, a new Tau/Amp pair will be created.

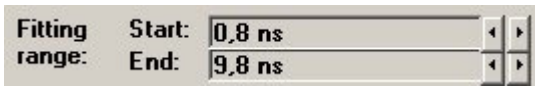
d. De-convolution

This check box specifies whether the fitter will use de-convolution or not.

If checked, the fitter will require specifying a convolution profile (see section a above).

Note: if you check this box, you may get an error message from the fitting engine, depending on your other settings (especially your start values). This is normal, and you can safely ignore the message for the moment by just hitting the OK button or pressing the Enter key.

e. Defining the fitting range

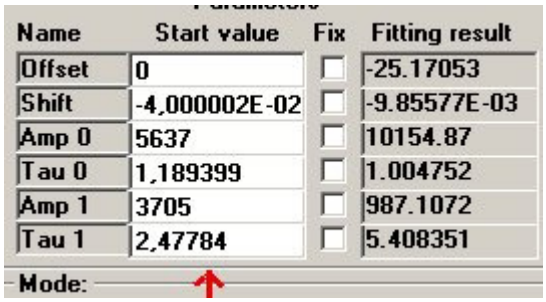


It is possible to specify the range of the decay curve that shall be fitted. In general, you won't want to fit the whole curve. The currently selected range is always displayed in the Start and End edit boxes. This range can be changed in two ways: either by dragging the cursors in the Fitting Display window (refer to section 3b below), or by using the spin buttons at the right end of the two edit boxes.

Normally, you will use the first method since it is easier and faster, but the spin buttons can be used to fine-tune the range what might be difficult with the dragging method. Note that each single mouse click on the spin button will move the cursor by exactly one channel (the time interval that corresponds to one channel will depend on the current time axis scaling, of course)

(For advice how to select a good fitting range, please refer to chapter F1.)

f. Defining the start values



In the central part of the Fitting Control window there is the "Parameters" section. This is organized as a kind of table:

The *rows* consist of the parameters of the chosen theory function, with the names "Offset", "Shift" (or "T0 fix"), "Amp i" and "Tau i". The correspondence of these terms with those in chapter C (e.g., eq.(C15/16) should be obvious.

And there are three *columns*: "Start value", "Fix", and "Fitting result". Only the contents of the first two columns can be changed by the user. The third column cannot be directly edited and is a pure output field.

Here we will describe the first column; the other two will be subject to the next two sections

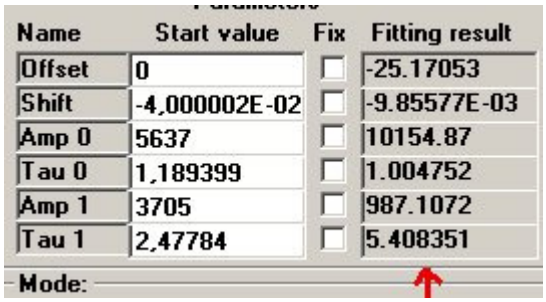
In the Start value edit boxes, you can directly type in the start value of each fitting parameter. It is important for the fitting process that these start values are chosen reasonably. (This point is detailed in chapter F2).

You can type in arbitrary floating-point values in these boxes. (It doesn't matter whether you use the dot or the comma as the decimal point. Also, you can use the scientific notation with an exponent such as "1.5E4" for example.)

Note that the name of the second-top parameter will be either "T0 fix" or "Shift", depending on whether Deconvolution is switched on or off. This corresponds to the treatment in section C1g. Note that "T0 fix" is *always* fixed. This point is further detailed in chapter F4.

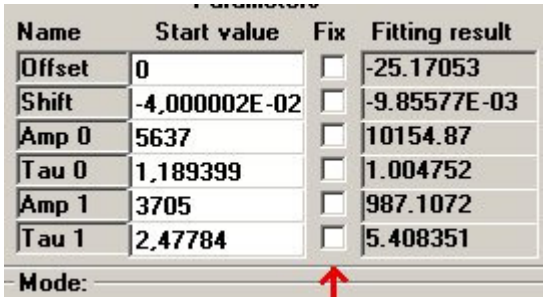
The display of the theory curve in the Fitting Display window will be temporarily disabled while you edit values here. If you want to see the theory function, you must enable it again (see paragraph i below).

g. Obtaining the results



In this column the fitter will display its results after the fitting procedure has terminated successfully. Note that if the procedure terminated prematurely, these values will remain unchanged.

h. Fix the parameters or let them float



This column of check boxes allows you to define each single parameter as either "fixed" or "free". If it is fixed, the fitter will leave the parameter alone and will always use your start value in its calculations. (Please refer to the Tutorial on examples how to use this feature).

i. What is displayed in the Fitting Display window

If "Start values" is selected, the theory curve resembling your start values is drawn. If "Result" is selected, the fitting result curve is drawn. And if "Don't display" is selected, no such curve is drawn at all.

Note that you can change the setting by yourself, but sometimes the system will change it automatically. For example, when you are editing the start values, it will automatically switch to "Don't display". (You need to switch to "Start values" again after you finished editing in order to see and judge the effect of your changed start values.) Also, after the fitting procedure has terminated successfully, the system will automatically switch to "Result".

j. Defining the stop conditions

This field allows you to change the values the fitting engine is using for the determination when the iteration shall halt. There are three conditions:

Minimum parameter change: This is the *relative* change of a fitting parameter of the current iteration cycle compared to the previous cycle. Hence it is a measure of the convergence achieved. The condition is true *only* when *all* parameters converged within the specified stability limit (logical AND).
The default is 10E-4. This is usually a good value, and you will probably not get much different results even if you specify a smaller number (such as 10E-6 or 10E-8) here. Be careful not to use too small numbers, or the fitter may not be able to reach the limits due to numeric precision limitations.

Timeout: Here you can specify the maximum amount of time the fitter will spend in the iteration.
The default is 60 seconds, which is rather long. You can safely input a smaller number here (like 10 sec), since - if the iteration should not terminate within the period - you can easily continue it (see next paragraph).

Max iterations: Here you can specify the maximum number of iterations the fitter will spend.
The default value of 1000 is very large and is probably never reached. There is little reason to change this value, but the choice is yours. The combination of a value of 1 with the Continue button (see next section) allows single-stepping.

These three conditions are combined by logical OR. It means, the fitter will terminate its iteration if *at least one* of the conditions becomes true.

k. Starting, interrupting and continuing the fitting process

Hitting the Start button starts the fitting procedure. You will see the small Fitting Progress window while it is running (see below). The fitting job can terminate due to the following three reasons:

One of the stop conditions became true (see previous section).

You pressed the Stop button in the Fitting Progress window (see below).

An error occurred (see section F3).

In the first two cases, the result will be treated as *valid*, the "Display theory" radio button will switch to "Result" and the fitting curve will be drawn in the Fitting Display window. In the last case, however, the fitting result will be treated as *undefined*: a new fitting curve will not be drawn, and also the "Fitting result" values in the "Parameters" section of the Control window will remain unchanged.

While the fitter is running, the small Fitting Process window stays open; it can't be disabled. This window shows the current iteration number, elapsed time, and the currently reached chi-square. However, the main purpose of this window is the "Stop Fitting" button that terminates the fitting procedure unconditionally. Although none of the stop conditions will have been reached in this case, the fitting result will be treated as *valid* (see above). This means, it is up to the user's responsibility to manually terminate the fitting process and interpret the obtained result correctly.

One reason why the user may wish to stop the fitter manually is "looping", a condition which is explained in more detail in section F9.

When the fitting procedure has been halted (either manually or automatically) without an error and you think the result could be improved by running the fitter a bit longer (e.g., by giving it a smaller value for the "Min. Parameter Change" condition), you can simply hit the Continue button. In this case, the fitter will internally use the already reached result values as the new start values (this is not reflected in the Start value column, i.e. your start values remain unchanged) and then continue the iteration. The timeout counter restarts from zero.

l. Getting help for setting the start values

Frequently you will find it inconvenient to figure out reasonable start values by trial-and-error and type them into the start values column. In such a case, you can let the fitter do the job for you. Simply hit the "Initial Guess" button and it will insert values in the Start value fields.

Please note that there may be cases where these start values are not useful (refer to section F2). Nevertheless, also in these cases the initial guess function may be convenient for finding approximate values. In any case, you should visually inspect the start values in the Fitting Display window (set the radio button to "Start values" before you start the fitting procedure.)

***Caution**:* the "Initial Guess" function does not respect the "Fix" flags. So, be careful to reset start values to your desired values which may have been changed by the "Initial Guess" function (especially the offset value!).

m. Configuring the Fitting Display window

Hitting the Options button opens the Options window that is explained in section 2 below.

n. Handling Fitting Sets

TA-Fit allows you define more than one set of fitting conditions at the same time, and switch between them. (By fitting conditions we mean the set of *all* entries which can be specified in the Fitting Control window.) This is very convenient for example if you want to check the effect of various parameters on the same data curve (for example, one set may try a 1-component fit while the other is setup for a 2-components fit). Another use would be the fitting of different curves (perhaps derived from the same streak image at different wavelengths), one in each set.

You switch between the sets by using the "Data set" spin buttons on top of the Control window. You can also enter a user-definable text string associated with each set. (If you don't type in your own text there, the program may insert a default text.)

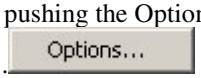
Fitting sets can be saved and loaded to/from the hard disk using the Save and Load buttons. Simply specify the filename and the directory in the usual manner. The filename extension defaults to ".SET", and it is probably a good idea if you don't change that

In addition to the .SET file, six other files are written to disk, containing the data. For details, please refer to Appendix A3.

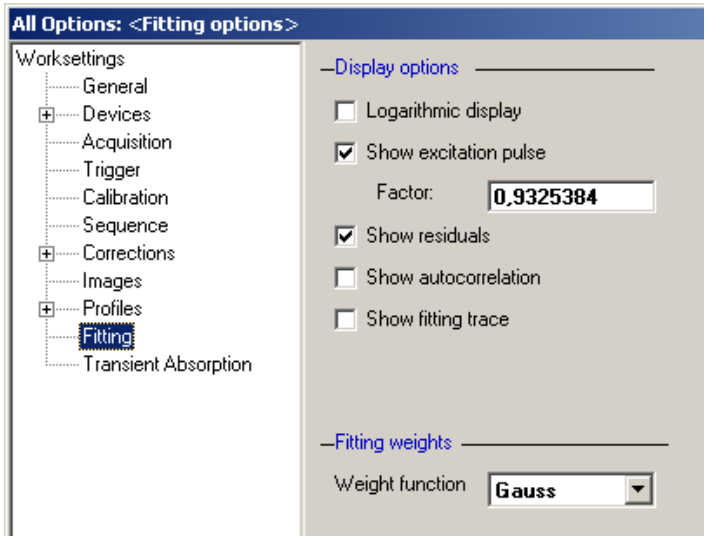
o. Ending the fitting session

Hitting the Close button in the upper right corner terminates the fitting session and closes all associated windows.

2. The Options Window

The Options window is opened by pushing the Options button in the Fitting Control window or by right click to the fitting main window and selecting . 

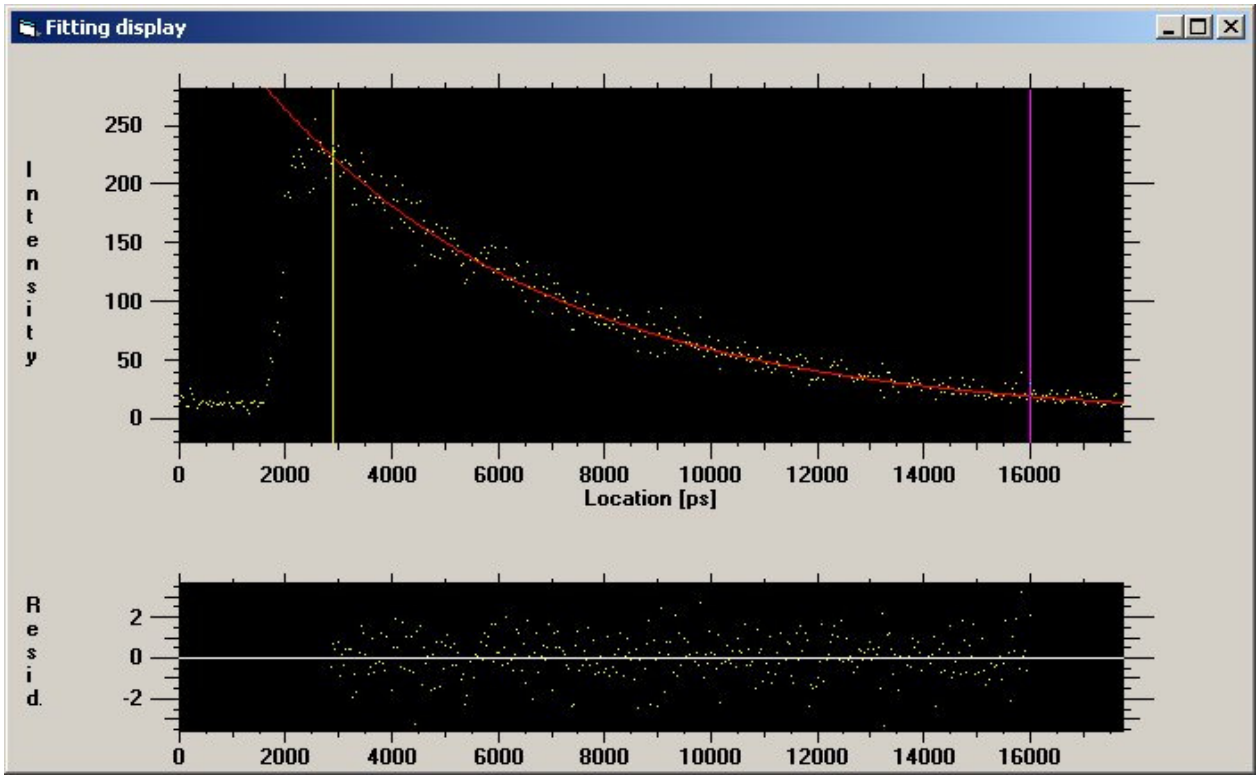
It lets you choose among several options related to the Display window.



You can choose the following options:

	Logarithmic display	The "Log display" checkbox selects a logarithmic intensity plot in the Display window, otherwise the plot is linear.
	Show Excitation pulse	Check the "Excitation pulse" checkbox if you want the laser profile be displayed in the Display window. We recommend enabling this feature. Since the intensity levels of the laser profile and the decay profile may be different, it is frequently necessary to re-scale the laser profile so that both profiles can be displayed in the Display window simultaneously. This re-scaling is done by applying a multiplication factor to the laser profile. Normally, TA-Fit automatically determines a useful re-scaling factor. If you want to change that factor you can manually edit the "Factor" box
	Show residuals	The checkbox "Residuals" control whether the residuals display are enabled in the Display window.
	Show autocorrelation	The checkbox "Autocorrelation" control whether the Autocorrelation display are enabled in the Display window. Note that autocorrelation can only be enabled if residuals are enabled.
	Show fitting trace	The checkbox Show fitting trace activates the "Fitting Trace" window (see section 4 below).
	Select the fitting weight function	This combo box selects between Gaussian and constant weight function.

3. The Fitting Display Window



This window is the place where you visually inspect the data curves, the fitting curve, the quality of the fit, and so on. It is also used to specify the fitting range.

a. What is displayed in this window

The Display window can show up to three displays simultaneously: the *data display* (showing the decay curve, the laser curve, and the fitting curve), the *residuals display* (showing the residuals and the fit quality parameters), and the autocorrelation display (showing the autocorrelation trace). The latter two can be selectively enabled or disabled using the checkboxes in the Options window (see section 2 above).

• *Data display*: Here the decay curve is shown (if one has been selected in the "Data profile" box already). Also, if the laser data have been specified in the "Convolution profile" box and the "Excitation pulse" checkbox in the Options window is checked, the laser profile is shown, too. Depending on the setting of the "Display theory" radio button, the theory function may also be displayed.

Note that the vertical scale of the display is always auto-zoomed to show the decay data in a best fashion. The scale on the left is valid (and always the same) for the data and the theory curves, but the laser curve may have been re-scaled (see section 2b above).

In addition, you may also see two vertical cursors in this display (see section b below).

• *Residuals display*: If a theory function is shown in the data display, then also its residuals with the decay data are shown in the residuals display (if that display is enabled). (For a definition of the residuals and their use refer to section E2.)

Note that the residuals are always auto-zoomed. Note also that the residuals display is aligned with the data display, so that the corresponding time channels are shown at the same horizontal positions.

Above the graphics display there is one text line where those numerical values are displayed which are measures for the quality of the given fit. These are:

"Chi^2_v": reduced chi-square

"Z": standard normal variate

"DW": Durbin-Watson parameter

"r": mean of residuals

"Sr": standard deviation of residuals

(For a definition and explanation of these parameters, please refer to chapter E.)

• *Autocorrelation display*: If residuals are defined, then their autocorrelation will be shown in the Autocorrelation display (if that display is enabled). (For a definition of the autocorrelation and its use refer to section E4.)

The autocorrelation trace is always auto-zoomed, but note that the first channel is excluded. (The first channel is always 1 by definition.) Also note that the Autocorrelation display is the only display in the Display window which has a time axis with a meaning different from the other displays. The number of channels in the Autocorrelation display is limited to a value of about 75% of the number of channels of the decay curve. Refer to section E4 for further explanation of these subjects.

Color codes:

yellow dots: decay data profile, residuals, autocorrelation

white dots: laser profile

red curve: theory function

white line: zero line

b. Moving the cursors

There are two vertical lines in the Data display: a yellow and a violet one. These are the "Start" and the "End" cursor respectively. They indicate the currently chosen *fitting range*. You can simply move around these cursors if you click with the left mouse button near one of them and then drag the mouse while keeping the button pressed. You don't need to point very precisely to the cursor since the cursor closest to the mouse cursor will automatically snap to it. (You know this method from the ordinary profile window in the HPD-TA already).

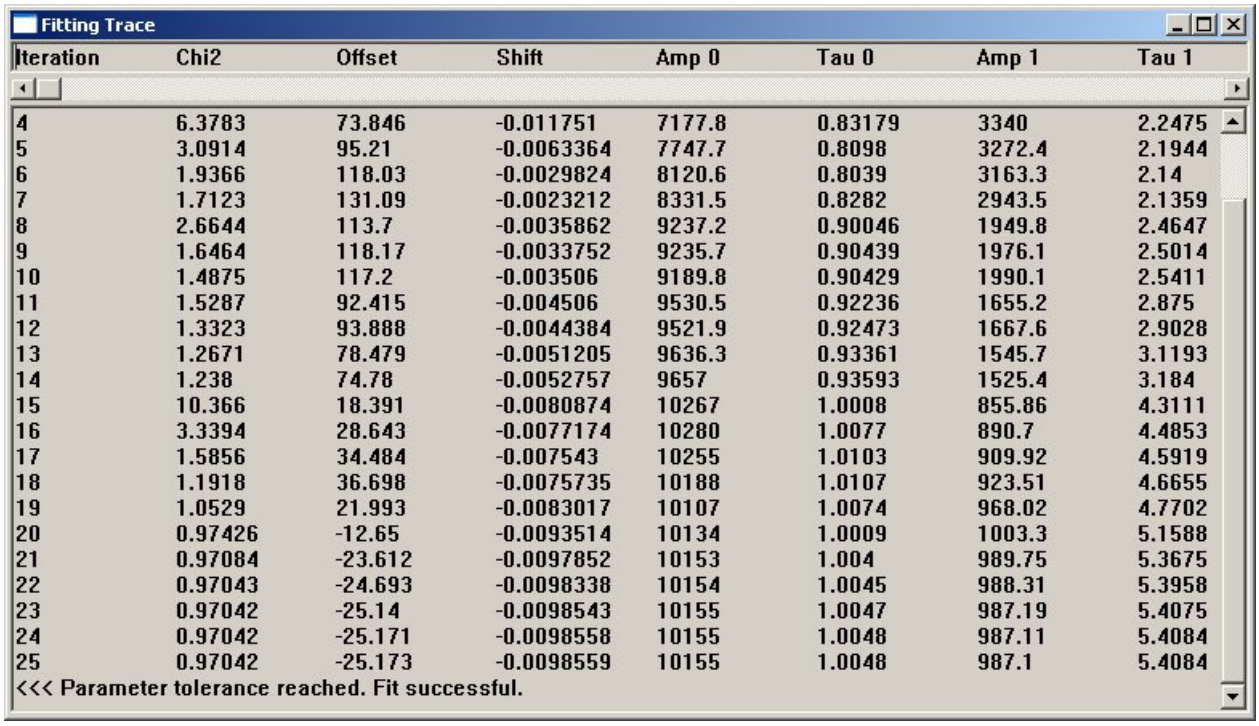
Hint: if you cannot see the cursors, they may be at the very left and right edge of the display. Simply clicking with the mouse in the display will make them appear.

You can read the numerical values of the cursor positions in the Control window, where you also can fine-tune them if desired (see section 1e).

4. The Fitting Trace Window

This is an optional window that is displayed only if enabled in the Options window.

a. What is displayed in this window



Iteration	Chi2	Offset	Shift	Amp 0	Tau 0	Amp 1	Tau 1
4	6.3783	73.846	-0.011751	7177.8	0.83179	3340	2.2475
5	3.0914	95.21	-0.0063364	7747.7	0.8098	3272.4	2.1944
6	1.9366	118.03	-0.0029824	8120.6	0.8039	3163.3	2.14
7	1.7123	131.09	-0.0023212	8331.5	0.8282	2943.5	2.1359
8	2.6644	113.7	-0.0035862	9237.2	0.90046	1949.8	2.4647
9	1.6464	118.17	-0.0033752	9235.7	0.90439	1976.1	2.5014
10	1.4875	117.2	-0.003506	9189.8	0.90429	1990.1	2.5411
11	1.5287	92.415	-0.004506	9530.5	0.92236	1655.2	2.875
12	1.3323	93.888	-0.0044384	9521.9	0.92473	1667.6	2.9028
13	1.2671	78.479	-0.0051205	9636.3	0.93361	1545.7	3.1193
14	1.238	74.78	-0.0052757	9657	0.93593	1525.4	3.184
15	10.366	18.391	-0.0080874	10267	1.0008	855.86	4.3111
16	3.3394	28.643	-0.0077174	10280	1.0077	890.7	4.4853
17	1.5856	34.484	-0.007543	10255	1.0103	909.92	4.5919
18	1.1918	36.698	-0.0075735	10188	1.0107	923.51	4.6655
19	1.0529	21.993	-0.0083017	10107	1.0074	968.02	4.7702
20	0.97426	-12.65	-0.0093514	10134	1.0009	1003.3	5.1588
21	0.97084	-23.612	-0.0097852	10153	1.004	989.75	5.3675
22	0.97043	-24.693	-0.0098338	10154	1.0045	988.31	5.3958
23	0.97042	-25.14	-0.0098543	10155	1.0047	987.19	5.4075
24	0.97042	-25.171	-0.0098558	10155	1.0048	987.11	5.4084
25	0.97042	-25.173	-0.0098559	10155	1.0048	987.1	5.4084
<<< Parameter tolerance reached. Fit successful.							

This window allows you to observe the progress of the fitting procedure. It displays the intermediary values of all parameters, one line per each iteration cycle. Only the parameters subject to fitting are shown (i.e., only those parameters which have not been "fixed" by the user). The order in which they are listed from left to right resembles the order in which they are listed in the Control window from top to bottom, with the exception that chi-square is always shown as the left-most value.

The window can be resized in the usual way, which may be necessary if you have a lot of parameters. When the list reaches the bottom, it will start to scroll.

Although it is not strictly required, we recommend enabling this window. Besides the fact that it is fun to observe the fitter doing its job, the window may also give you useful hints if a fit happens to fail.

b. Messages that might appear

After the fitter stopped running, there may be an informative message at the bottom of the list telling the reason of the stop. Here you can see whether the fitter stopped because one of the three stop conditions was met (and you can also see *which* of these conditions), or whether there was another reason such as that the fitting procedure did not converge normally.

5. Global fit

Global fits can now be performed by using several fitting sets together.

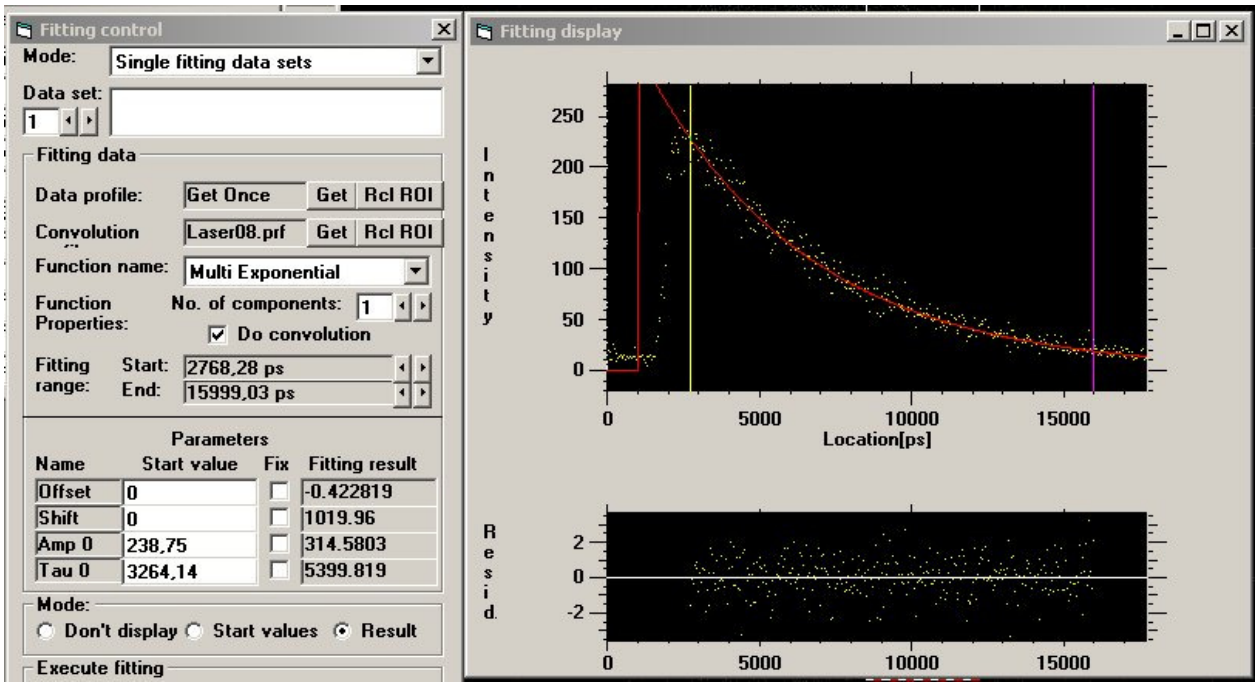
General introduction

Proceed as follows:

- Use one data set to fit a part of your data. Find the correct parameter (fit function and number of component) and reasonable start values for them.
- Use a second data set to it another part of your data. Use the same parameter to fit the second data set.
- Switch to global fitting. The set you were currently working on define which parameters can be shared with the other data set and provide the start values. Decide which parameters you want to share.
- Then execute Do Fitting. The fit engine fits both data keeping the value for the shared parameters the same.
- The result of the shared parameters can be visible immediately in the global fitting display.
- The parameters of the individual data sets which are not shared can be viewed when switching back to “Single fitting data sets”.
- Use as many data set as you wish to correlate in the same way

Use the first data set

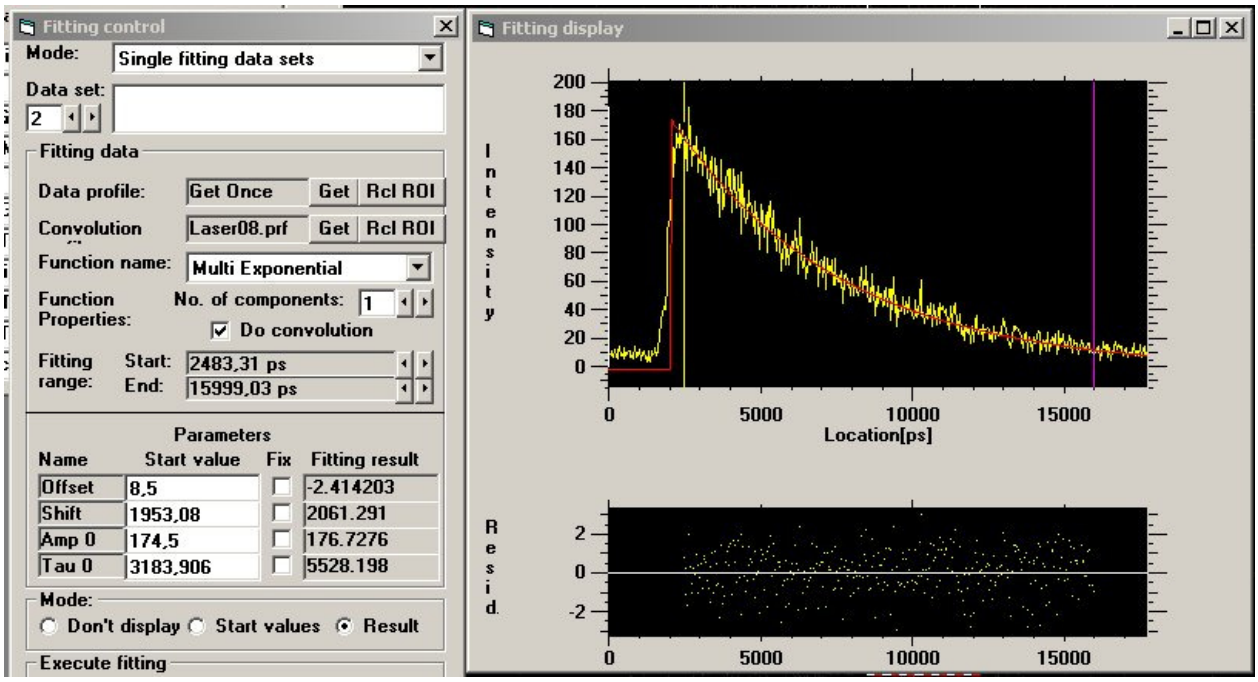
Use one data set to fit a part of your data. Find the correct parameter (fit function and number of component) and reasonable start values for them. Execute Do Fitting.



Please note that the fit engine finds the value 5399.819 for Tau0

Use the second data set

Use a second data set and load to it another part of your data. Use the same parameter to fit the second data set.

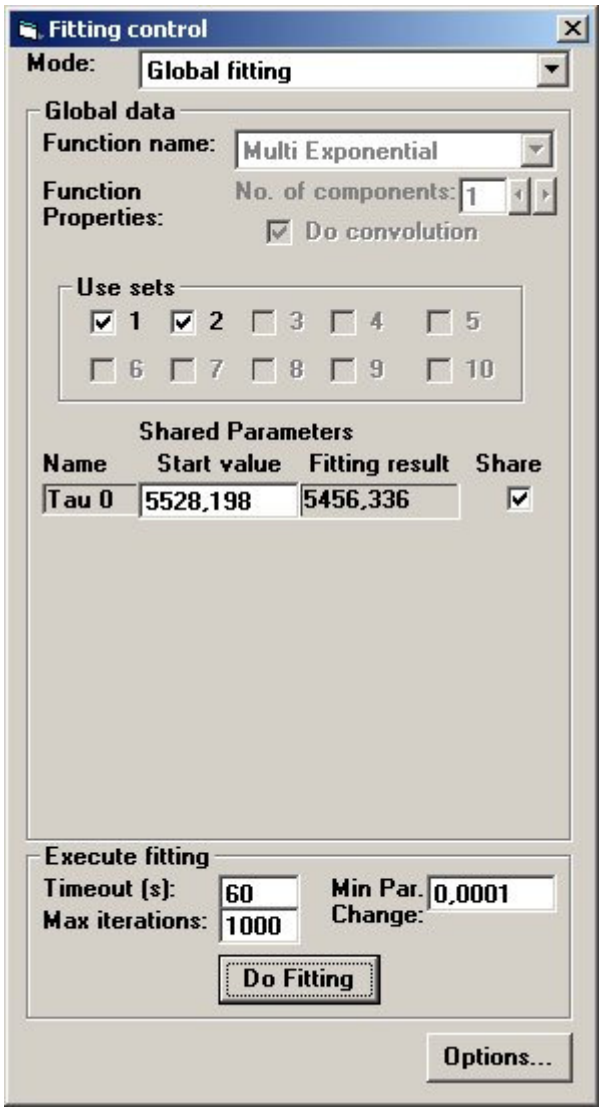


This data is derived from another part of our sample image and has much more noise. Note that the fitting yields 5528.198 for Tau0

Use global fitting

Now let's switch to global fitting. This is done with the topmost control in the main fitting window:

we get the global fitting window.



The set you were currently working on define which parameters can be shared with the other data set and provide the start values. Decide which parameters you want to share. In our case we will share tau0.

After executing Do Fitting we get the result of 5456,336 for Tau0.

The result of the shared parameters can be visible immediately in the global fitting display.

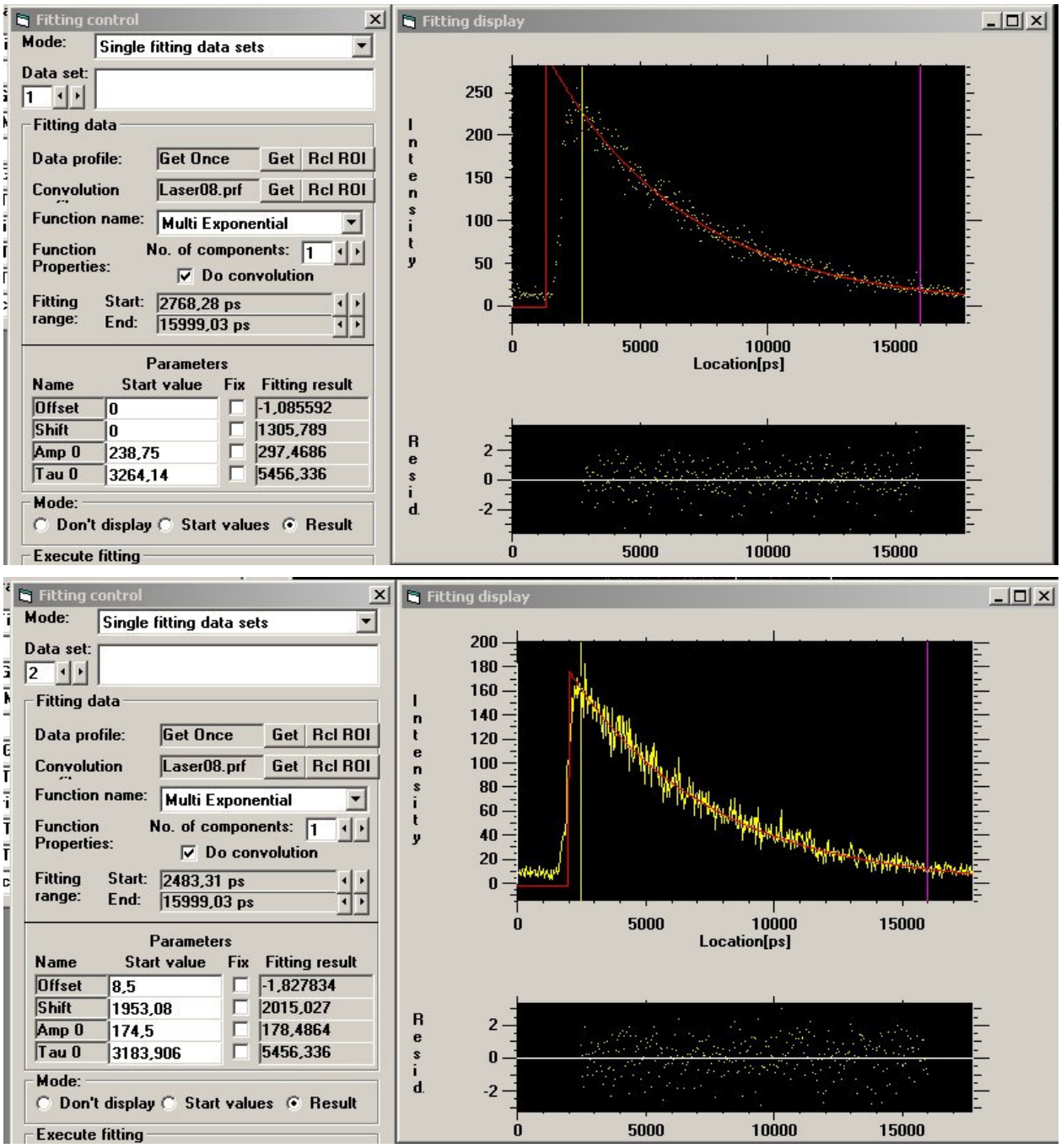
Viewing the complete parameter results for all data sets

The parameters of the individual data sets which are not shared can be viewed when switching back to “Single fitting data sets”.

Switch back to “Single fitting data sets”

This is done with the topmost control in the main fitting window:

You will get the following results for the two data sets:



Note that the result for Tau0 is the same for both data sets, whereas other parameters differ.

Use as many data sets as you wish to correlate in the same way.

E. Judging the Quality of the Fit

This chapter explains the various methods of judging the quality of a given fit result.

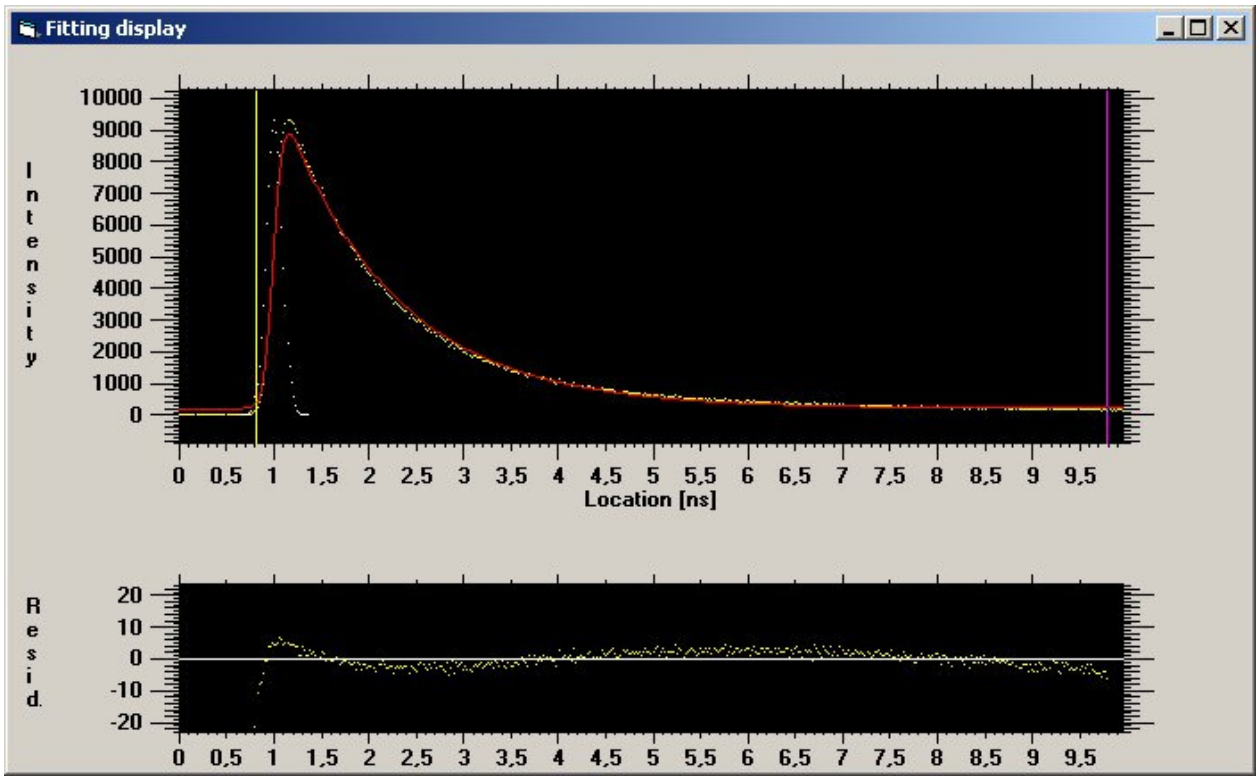
The reader should be aware there is no single definition telling what we shall understand by "quality". Here, we use the term in an intuitive manner. The methods described below are different ways of giving "quality" a concrete meaning.

1. Visual Inspection of Data and Fit Curves

The first - and most obvious - method in judging how well the result curve resembles the measured data is simply the visual comparison of the two corresponding curves. This can be done in the data display of the Fitting Display window.

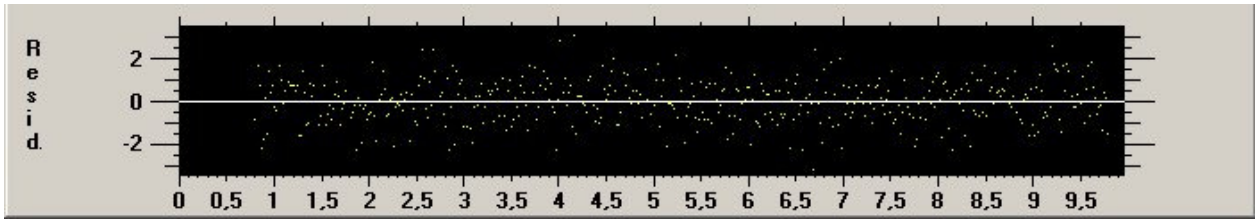
2. Residuals Curve

The trouble with the above direct method is that the deviations are frequently too small to be distinguished well by bare eyes. Since the deviations are typically small compared to the absolute values, they are hard to recognize - one would like to "zoom in" on the differences. This is exactly what the plot of the residuals does. More precisely, the plot shows the "weighted residuals" (for the definition, refer to section C2a, e.g. eq. (C19)).



The below example is a fit with too few components. By method of direct visual comparison of the data and fitting curve, the problem is hard to recognize. The residuals curve, however, clearly makes it visible.

In contrast, the residuals of a good fit are shown below (same data as above).



How should the residuals look if the fit is "good"? A large noise does not by itself mean that the fit is bad, since large deviations are simply the result of low count numbers. However, there should be no *systematic* distortion on the residuals curve. This means mainly: (a) the residuals curve must not be skewed (indicating that something went badly wrong with the fit, yielding too long or too short lifetimes); (b) it must not contain a "low-frequency ringing" (indicating that the theory function contained too few free parameters, see the example above); (c) the overall "noise level" must not vary grossly (if this were the case, it would indicate abnormal noise behavior of the data; remember that we are talking about *weighted* residuals here).

In summary, we can say that a "good" residuals curve should have a random, high-frequency variation around zero, with no obvious distortions.

Residuals curves are frequently shown when reporting or publishing fitting results.

3. Chi-square

When we watch out for a single numerical value indicating the quality of a fit, the chi-square value is the most obvious, since it is exactly this value which is minimized by the fitting engine. (Actually, the fitting engine looks *only* at this value.)

For the definition of the "reduced chi-square" refer to section C2a, eq. (C25).

As explained in Chapter C2a, the optimal value of chi-square would be 1.0, but real values differ from that optimum. So, the natural question is: how much may chi-square differ from 1.0 if I want to be sure that my fit is still "good"?

It is important to understand that there is *no* clear, unique answer to this question. In practice, values smaller than 1.2 are frequently treated as acceptable. However, this acceptance range can not be more than just a very rough rule of thumb. First, the acceptable number will depend on the degree of freedom (ν) of the fit (refer to Chapter C2a). Second, it may be known that the statistics of the data is not really Gaussian, so some deviation *has* to be expected (see also Chapter F8). Third, obviously, what counts as "acceptable" is a matter of taste anyway. To illustrate the latter point, consider that systematic measuring errors can never be completely eliminated (although they "should"), so even for large count numbers some deviation of chi-square from 1.0 has to be expected. A researcher will consider these systematic errors when judging his chi-square values. So, in some situations, a chi-square of a given fit much better than 1.2 may not be acceptable, since it is known that it "usually can do better", while in other situations a value even larger than 1.2 may be accepted, since it is "good enough under the given circumstances".

Also, one should be aware that in practice there are quite frequently cases where a relatively small chi-square is achieved, but even a brief glance to the residuals (or autocorrelation) curve reveals gross and systematic errors. (Such cases can be found in the Tutorial.) So, it is strongly recommended to always consult those curves, too. A fit with a relatively poor chi-square and a reasonable residuals curve may still be considered acceptable, while a fit with a "good" chi-square and an obviously twisted residuals curve will never!

Despite of the above, in the literature one can find some trials to obtain a more objective meaning of "fit quality" in terms of chi-square; for a brief mention, please refer to section 8 below.

Chi-square values are usually shown when reporting or publishing fitting results.

4. Autocorrelation Curve

The definition of the autocorrelation curve is given below:

$$C_j = \frac{\frac{1}{N-j}\sum_{i=n_1}^{n_2-j} R_i \cdot R_{i+j}}{\frac{1}{N}\sum_{i=n_1}^{n_2} R_i^2} \qquad \text{(E1)}$$

, where N is the number of fitted channels ($N = n_2-n_1+1$), i.e. the width of the fitting range.

The data points C_j of this curve are a measure of how correlated are those residual values with a distance of j channels between each other. These correlations are averaged over a wide range of channels. Note that the index j ranges from 0 to approx. 75% of N. (It does not range to N, otherwise there would be not enough channels for the averaging, rendering the C_j too noisy and therefore useless.)

If there were no correlations at all between the residuals, then in the limit of infinite N all C_j would tend against zero (because the random variations would cancel out). (With the exception of C_0, which is always exactly 1.) Actually, N is smaller than infinite, so the C_j will show random variation.

Similar as in case of the residuals themselves, the autocorrelation curve is "good" if its noise looks random and does not show systematic tendencies. Due to its nature, the autocorrelation curve is especially sensitive to *regular* oscillations in the residuals.

Such oscillations can sometimes be caused by certain problems in the experimental conditions, e.g. electromagnetic RF noise induced in the detector. Such interference can be a standard annoyance in case of experimental setups using photomultipliers as detectors and HV-driven flashlamps as light sources. In case of streak detectors and lasers, the significance of such types of distortions has been reduced.

In many cases, the autocorrelation curve allows to spot problems more clearly than the residuals curve. (The Tutorial contains examples.)

Autocorrelation curves are frequently shown when reporting or publishing fitting results.

5. Standard Normal Variate

As mentioned under section 3 above, a weakness of the chi-square is that its "acceptable" value range depends on the degree of freedom (ν). One approach to circumvent this is the definition of the so-called standard normal variate Z, which is supposed to be independent of ν. Its definition is

$$Z = \frac{\nu \cdot (\chi_\nu^2 - 1)}{\sqrt{2\nu}} \qquad \text{(E2)}$$

The standard recommendation is that Z should be between -3 and +3. This is approximately equivalent to a chi-square of 0.81...1.19 in case of 500 channels, or 0.87...1.13 in case of 1000 channels.

Although Z has the advantage of being independent of ν, it can still not be regarded as "the objective" measure for the quality of the fit, since other qualifications similar as in case of chi-square still hold (see section 3 above).

In practice and in the literature, Z appears to be rather rarely used.

6. Durbin-Watson Parameter

The Durbin-Watson parameter is also a kind of correlation value. Its definition is:

$$DW = \frac{\sum_{i=n1+1}^{n2} (R_i - R_{i-1})^2}{\sum_{i=n1}^{n2} R_i^2} \qquad \text{(E3)}$$

A comparison to eq. (E1) shows that DW is very similar to the second channel of the autocorrelation profile (C_1).

This parameter is said to be "good" if it lies above a certain lower limit (typically in the order of 1.8). The exact values of these limits, however, depend in a non-trivial way on the number of fitted channels and on the theory function. One can find tables of DW limit values in the literature (see Appendix).

Obviously, the autocorrelation curve gives a wealth of more information than such a single DW number. So, we believe the inspection of the autocorrelation curve is a much more useful way of judging a fit.

DW values are sometimes shown when reporting or publishing fitting results, but increasingly rarely.

7. Mean and Standard Deviation

The other two values displayed in the Fitting Display window are simply the mean and the standard variation of the residuals. They are defined as:

$$\overline{R} = \frac{1}{N}\sum_{i=n1}^{n2} R_i \qquad \text{mean:} \quad \text{(E4)}$$

$$\sigma = \frac{1}{\sqrt{N-1}}\sum_{i=n1}^{n2} (R_i - \overline{R})^2 \qquad \text{standard deviation:} \quad \text{(E5)}$$

Obviously, for a good fit these values must be close to 0 and 1, respectively, if the noise is Gaussian as assumed. However, these two values are not very sensitive parameters for judging the quality of a fit. Many bad fits will not be revealed by them. On the other hand, if there were big deviations from 0 and 1, it would indicate a distorted statistics or a bad fit with certainty. So, these parameters may play a role in the diagnostics of gross problems.

Due to their insensitivity to the more subtle points of fit quality, these values are almost never mentioned in publications.

8. Other Criteria

In our above discussion of chi-square (section 3) we said that a good chi-square will be near 1.0, but we have been vague about what can be said about "limits" of acceptance.

It should be noted that in the literature one can find many approaches to define "confidence limits" (or "confidence planes") as objective criteria for the quality of a fit. The aim is to calculate a probability that a given calculated parameter is correct or, respectively, an error range for that parameter. In most cases, such approaches are centered on the so-called "correlation matrix" which is measure of the stability of the parameters.

Actually, we decided *not* to output such confidence limits. (The reason is not that we don't want to calculate them. Actually, the correlation matrix is *always* calculated during the fitting process.) The reason is that these numbers tend to be *much too good*. They can be highly misleading since they suggest a precision of the fit that simply does not exist. (This is not a fault of TA-Fit but an experience common to all researchers in the fitting business.)

In our opinion, the only *reliable* way to estimate the confidence one should have in given fit (apart from a good deal of experience), is to *repeat the measurement* and study the variation of results.

F. Questions & Answers

1. How do I Correctly Choose the Fitting Range?

The answer depends on whether you are using deconvolution or not.

If deconvolution is ON, you should place the start of the fitting range near the position where the fluorescence curve *starts to rise*. If deconvolution is OFF, you should place the start of the fitting range *after the peak* of the fluorescence curve. (How *far* you put it after the peak depends on the width of the laser pulse; if it is broader you should put it farther away.)

The end of the fitting range should be placed near the end of the data.

The fitting range should not include large areas where the data are zero or very close to zero, since this may affect the fitting procedure negatively.

The above is just the standard rule. In practice, the details may change from case to case. In critical cases where the fitter is unstable (due to too noisy or bad data, especially when doing multi-component fits with deconvolution), it may be necessary to play with different fitting ranges. The Tutorial gives a good introduction in such practical questions.

2. How do I Choose Good Start Values?

In general, we recommend using the "Initial Guess" button to generate the start values. In some critical cases, however, the button will return values that are not optimal and may not allow the fitter to run well. In those cases, some of the start values will need manual tuning.

This tuning will usually be necessary only for the lifetime values, and you don't need to change the amplitudes. (This means, even in those cases, the "Initial Guess" function is the best way to find the right order of magnitude for the amplitudes.)

In case of fits with more than two components, we recommend to "spread" the lifetime start values maximally, in order to avoid "melting together" during the fitting process. You shouldn't use too small (say, smaller than 1% of the time window) start values, however, otherwise the component may tend to get eliminated during the fit.

Special care is required when fitting data with rising components, i.e. with negative amplitudes. In those cases, the fitter may not find negative amplitudes if you start with positive start values. So, if the "Initial Guess" did not deliver a negative amplitude, you need to enter it by yourself.

We recommend referring to the Tutorial for some practical examples.

3. Why do I Get These Error Messages?

We assume you are referring to the small error message boxes saying, "The fitting engine cannot continue due to mathematical divergences". The message indicates that the fitting engine encountered a situation such as division by zero, overflow, etc., which may be caused by a variety of reasons (infinite derivatives, singular matrices, etc.).

You should not be afraid if you see this message; it doesn't indicate a program hang-up or such. Just quit the message box by hitting the OK button or the Enter key and go ahead.

The fitting engine is tuned for speed. Due to technical reasons it is computationally cheaper to trigger an exception when an error occurs instead of checking for numerical conditions before each calculation.

Typical reasons for triggering the message include: you forgot to define a non-zero start value for some parameter; you started fitting with an unsuitable fitting range or start values; the data are difficult to fit due to their noise or too many free parameters. In the latter case, if you can't avoid the message even if you try to tune the start values and the fitting range carefully, it may simply indicate that the fit is too critical. In such a case, fitting results must be interpreted with great care (since their values may depend on the choice of those arbitrary conditions).

You should not get other sorts of error messages except the above. If you happened to get another one (perhaps from the operating system) it would indicate a problem. In such a case, please try to reproduce the error condition and contact HAMAMATSU.

4. I Can't Fit T0. What is Wrong?

Nothing is wrong. T0 *cannot* be fitted. This is why the parameter is called "T0 fix" in the Control window and the Fix checkbox is always checked. The reason is a pure technical one. The fitting process involves frequent calculations of partial derivatives (ref. to chapter C2c). In case of fitting without deconvolution, the theory function has a true discontinuity at T0 (see eq. (C8) or (C13)). This means, the derivatives get infinite here, and as a consequence T0 cannot be fitted.

Isn't this a problem? No. The lifetimes resultant from the fit are independent of T0.

Don't the amplitudes depend on T0? Yes, they do. The reason for this can be readily seen by referring to the formula of the theory function (e.g. eq. (C13)): a different T0 cannot be distinguished from a different amplitude. (Since T0 is constant, it can be pulled from the exponent of the exp-function to a position in front of the exp-function, becoming a constant factor.) So, if you want to obtain amplitudes which correspond well to the peak intensity of your fluorescence curve, you should put T0 close to (but always before) the peak position. But, this is normally not important at all.

So, what shall I do with T0? In general, you should set T0 to zero and don't care about it.

5. How Should I Treat the Offset Parameter?

The Offset can be fitted like any other parameter. However, in many cases it is difficult for the fitting engine to determine the Offset precisely, especially if there are long lifetime components. (This point is detailed in the Tutorial. Please refer to it.)

This means, the best is if the offset of your measurement data is zero or at least very close to zero.

It is recommended that you *fix* the Offset to zero (or to a rest value close to zero) whenever possible.

In case of photon-counting measurements, you can usually assume that the offset is exactly zero. In case of analog measurements, you should at least minimize it by background subtraction and so on. In the latter case, a small offset may remain sometimes; in such a case, inspect the profile via the HPD-TA Profile Window, read the remaining offset from it and input the number into the TA-Fit Offset box, and fix it.

6. How do I Use Non-Linear Time Axis Calibrations?

Don't worry about this. TA-Fit is always using correct time axis data, regardless whether the calibration is linear or nonlinear. This is fully automatic and does not require any attention from the user.

7. Why is My Chi-square not Unity?

Apart from the obvious and trivial reason of too noisy data (which will yield chi-squares close but not very close to 1), there can be several reasons why chi-square does not converge against a value close to 1.

One possibility is that your data are not "true" photon-counting data. Perhaps they have been taken in analog mode or in binarization (= photon counting without peak detection) mode. In these cases, please refer to Question 8 below.

Another obvious possibility is that your data contain more components than your theory function. This should be easily recognizable by inspecting the residuals curve and the autocorrelation curve (refer to section E2). In this case, increase the "number of components".

Finally, your data may contain *systematic* errors. In real life, there are plenty of possible causes for such measurement errors, and we can't list them all. The most common cases are probably:

You have constant offset (perhaps stray light or camera dark current) on your data. Please refer to Question 5 above.

In photon-counting mode, you used too high count rates. In this case, HPD-TA could not always resolve two photons as such and may have frequently counted two photons as only one. This will cause a distortion at the high-intensity areas of the data, known as "pile-up". You need to repeat your measurement with lower count rates.

You may have a large "shading" on your streak image, causing (typically) a signal too high in the center of the image compared to the edges. In this case, you need to perform a shading correction on your image data. (Refer to the HPD-TA manual.or help file)

8. What is the Statistics of Streak Data?

In this document we mostly assume the ideal case that the fluorescence data are Poissonian (see section C1b). The fitting engine is based on a Gaussian model, which is a good approximation of Poisson statistics in most cases (see section C2a).

The question remains whether the statistics of real streak measurement data *are* close to Poissonian (or Gaussian) or not.

The answer on this question depends on the mode of measurement:

Photon-counting mode

Here again, we need to distinguish two cases:

Photon-counting with peak or center-of-gravity (COG) detection ("true" photon-counting)

Photon-counting by binarization only

Method (a) is currently only available in the so-called PCI versions of HPD-TA. On each single-photoelectron spot a peak or COG detection is performed, leading to a count of exactly one pixel for each detected photon. (For details, refer to the HPD-TA manual or help file) It is well known (see literature in Appendix) that the statistical behavior in this case is quite well approximated by a Poisson distribution. This near-Poissonian behavior of the detection system is regarded as a nice feature since it resembles well the true statistics of the photons themselves.

Method (b) is the only counting method available with the so-called AFG versions of HPD-TA. In this mode, the peak or COG detection is impossible due to technical reasons. For each photon, a counting *spot* is recorded. The height of the spot is set to unity, but the *size* of the spot is larger than one pixel. This means, for each spot a typical number N_{spot} of pixels will be affected. Nevertheless, it can be shown (see literature) that both the number and the individual sizes of those spots still follow a Poissonian statistics quite well. This means, we expect that chi-square fitting is still applicable, but the reduced chi-square will not converge against 1 but against N_{spot}. So, we can still

apply the fitting method and interpret the obtained chi-square values in the usual way, if we don't forget that we have to expect them to converge against N_{spot}.

2. Analog mode

In contrast to the photon-counting methods discussed above, no firm assumptions can be made in case of analog measurements. Although there exist some publications about the statistical behavior of streak tubes in general, it is very hard to derive useful conclusions about the expected statistics of actually measured data. One reason is that it depends simply on too many experimental conditions, mainly the MCP gain (which can be freely chosen by the user), noise characteristics of the CCD device (which is not eliminated by discrimination, in contrast to the photon-counting modes), efficiency of CCD coupling, and so on.

So, generally speaking, in analog mode there are no clear reasons for expecting neither a Poissonian nor a Gaussian statistics. Nevertheless, we believe that chi-square fitting is still a useful method even here, since probably the noise behavior can be at least roughly approximated by a Gaussian (see the remark on the near-universality of Gaussian behavior in section C2a). What cannot be assumed, however, is that the standard deviation will be close to the square root of the count numbers. Accordingly, the weighting factors will not be correct and also chi-square values will not tend to be close to 1, even if the fit is otherwise acceptable.

9. Why does My Iteration Never Stop?

You always can manually stop the fitting process by hitting the "Stop Fitting" button in the small Fitting Progress window.

There are two situations that appear as if the fitting iterations would not come to an end:

After approaching the chi-square minimum quite closely, the algorithm may enter a state of "looping". What happens here is that the algorithm "rides on the edges" of the chi-square minimum in a cyclic fashion without ever hitting its center. In this case, the algorithm infinitely cycles between a few sets of parameters. The period of such a cycle is typically around 3 (i.e. after every third iteration exactly the same parameters are reached). This condition can be detected by eye by looking at the iteration log in the Fitting Trace window. Since the parameters keep continually changing from one iteration to the next, the stop condition specified in "Minimum parameter change" is never true, and so the fitter never stops (until the time-out is reached). In those cases, please simply stop the process manually; the fitting result will usually be good enough that you can use it well.

In some situations, it may happen that the fitter "crawls" very slowly along a valley in the chi-square landscape. (Due to some special condition of that landscape it does not adjust its internal step width efficiently.) Such situations should be extremely rare. Here, it is not exactly true that the fitter will never stop; actually, it will stop, but it may take extremely long. Probably you should stop it manually in such a case. However, unlike in case (a) above, after the manual stop the fit result may still be far from optimum. You should slightly change your start values and/or the fitting range and start over again; most probably the problem will be gone.

10. How Can I Speed Up the Fitting Process?

The time required for a fit depends - among others - on the following factors:

(a) The number of data points of the fitting range

The dependence is linear. If you reduce the fitting range, there are less calculations to do. Of course, the fitting range cannot be reduced arbitrarily, without compromising the quality of the fitting result. You should be aware, however, that it makes no sense to choose a too wide fitting range. (See also Question 1).

Typically, a speed increase obtained this way will be rather small.

(b) The number of data points of the laser curve

This dependence is linear, too. (Of course, it contributes only if deconvolution is ON.) Notice that there is no way to restrict the part of the laser curve which is used for fitting (as analogous to fitting range). The fitter always uses the *whole* laser curve. So, the number of data points must be restricted from the beginning, i.e. when generating the laser profile using an ROI in the streak image. Do not use a larger ROI than necessary; the inclusion of areas where there is no laser signal is useless and only slows down the fitting.

Notice that *both* above points (no.1 and 2) have a linear influence on fitting speed. This means, if you record the streak image with half resolution (e.g. by using a C4880 camera in 2x2 superpixel mode instead of full resolution), the speed improvement will be 4-fold.

(c) Whether the deconvolution is ON or OFF

The fitting process is much slower when deconvolution is ON. The frequent calculation of the convolution integral is computationally expensive. In many cases, you can obtain excellent results even without deconvolution.

(d) The speed of your computer.

This is the point where you perhaps can speed up the fitting processes most easily. Buy a faster machine. On a modern Pentium PC, TA-Fit is so fast that considerations for speed increase are quite obsolete. Also note that TA-Fit spends almost all time in floating-point operations. The floating-point unit of a Pentium class processor is much more efficient than that of older processors, even at similar clock rates.

(e) Background applications

Microsoft Windows 3.x and Windows 95 are multitasking systems, allowing other tasks running in the background while TA-Fit is active. This may degrade the performance of TA-Fit. Check that no background programs are running which you don't really need.

11. How Can I do Anisotropy Analysis?

The current version of TA-Fit does not support fluorescence anisotropy analysis.

Whether we will offer such a feature in future versions will depend on the requirements from our users and potential customers. Consequently, if you are interested in such a feature, please inform HAMAMATSU about your desire.

G. Tutorial

This tutorial guides you through some typical fitting sessions. Important techniques and frequent problems will be emphasized and demonstrated.

Please go through the tutorial *sequentially from first to last lesson*, without skipping any parts.

This is necessary because only in the first lessons the operation procedures are explained in detail, and later lessons will just refer to it. Also, when some data or setups are changed during a lesson, the subsequent lesson will silently assume these settings and rely on them.

You do not need to operate your streak camera for this tutorial. All required data are on disk.

The data used in the tutorial are only *profile* data (no images), since using images does not add extra insight. (For an example involving an image, please refer to the Quick Start section in chapter B.)

All the sample data are not results of real measurements but have been *synthetically generated* by means of simulation software. This approach, obviously, has its advantages and disadvantages. The advantages are mainly: (a) we can generate the data easily; (b) we can study certain points in a "pure form", without being distracted by other phenomena or side-effects; (c) we don't need to be afraid of artifacts such as systematic measurement errors which could - if unrecognized as such - lead to wrong conclusions. The latter point also leads us to the main disadvantage: if we only play with clean, synthetic data, we will learn nothing about those "dirty problems" which will occur only in the laboratory practice. Despite the latter limitation, however, we believe the usage of synthetic data is justified, at least for the beginning.

Note: Don't be afraid if you encounter some error messages during the course of this tutorial. Their occasional appearance is normal. (Refer to chapter F3.)

Also note: The behavior or results of your version of TA-Fit may slightly vary from the descriptions in the Tutorial due to changes in the fitting engine between TA-Fit sub-versions. Please don't contemplate on those differences. They are minor and do not spoil the key points of the lessons.

Lesson 1

Start the fitting session by selecting "Fitting" from the "Analysis" menu.

If you like, adjust the size and position of the Fitting Display window.

Hit the "Options" button in the Fitting Control window. Switch logarithmic display OFF, excitation pulse ON, residuals ON, autocorrelation OFF, Fitting Trace ON. Close the options window by clicking the OK button.

Hit the small "Get" button at the "Data profile" line. Choose "Get from file" and "vertical", then hit OK. In the file selection dialog, change to the tutorial directory. Select the file named "DECAY01.PRF" and hit OK. The decay data should now be displayed (as yellow dots) in the data display (upper part) of the Fitting Display window.

Repeat step 4, but this time use the "Get" button in the "Convolution profile" row. Choose the file "LASER01.PRF". The laser's excitation pulse (white dots) should now be displayed, too.

Set "number of components" to 1 by using the spin buttons (right of the text "Function properties"). This selects mono-exponential fitting. (Leave "Do convolution" OFF for the moment.)

Click with the mouse near the left border inside the black area in the data display of the Display window and drag the yellow cursor to the right. This yellow cursor defines the start value of the fitting range. Position the cursor at around 2 nanoseconds. (The numerical value is also shown in the Fitting Control window, right of the text "Fitting range".). In the same manner, drag the end cursor (violet color) from the right edge of the data display, and put it near 9 nanoseconds.

Define your start values by typing some numbers into the left column of the "Parameters" section of the Fitting Control window. For example: "Offset"=0, "T0 fix"=0, "Amp 0"=1000, "Tau 0"=1. Nail Offset to zero by checking its "Fix" checkbox. We do this because we don't want the fitter to change this parameter. (Note that T0 is *always* fixed.)

Hit the "Do Fitting" button.

The following things should have happened now: The "Fitting Trace" window (sometimes also referred as "log window") has opened and shows a log report of the fitting process. (Also the small "Fitting Progress" window had been opened, but - depending on the speed of your computer - you may not have noticed, since it is closed automatically.)

The trace window should report that the fitting procedure has ended after a few iterations (maybe 10 or less) with a message saying "Parameter tolerance reached. Fit successful." This indicates that nothing went wrong. ("What tolerances?", you may ask. Answer: the tolerances specified in the "Execute fitting" section at the bottom of the Fitting Control window. Refer to section D1j for details, but postpone this for later.)

You can find the *fitting results* (amplitude and lifetime) in the right column of the "Parameters" section in the Fitting Control window. We don't care about the amplitude at the moment. The lifetime should be very close to 2.0. (Actually, this was the value with which the data were synthesized.)

The *fitting function* is drawn in the data display in red color.

Also, you can see *how* the fitting engine had converged on the final values by referring to the trace window. (The last line in that window is identical with the fitting results displayed in the Fitting Control window, at least when no error occurred.)

Note that the first column of the log window is always the iteration counter, and the second column is always the chi-square. The real parameters start from the third column on. Note, too, that those parameters which were "fixed" during the fit are not displayed in the trace window.

Finally, where can you find the output indicating the *quality* of the fit? In the Display Window, between the data display and the residuals display, there is a row of text. The following numbers are shown there: Chi-square (probably around 1.05 in this example); the standard variate Z (around 0.7); the Durbin-Watson parameter (around 1.9); the residuals mean value (close to zero); and the residuals standard variation (close to 1). (Refer to chapter E for details about these parameters.)

The *most important* quality check, however, is the inspection of the *residuals curve* (the curve of yellow dots in the residuals display, below the data display). These spots should be spread around the zero axis randomly, without being skewed or showing obvious oscillations (if there are fluctuations there, they should at least have no low-frequency component).

Play around with the fitting range cursors and hit the "Do Fitting" button several times to see how the choice of the fitting range would influence the result of the lifetime. It won't change much, unless you do one of the following mistakes:

The range is simply too small. When the range gets smaller, the results get less and less reliable.

If you extend the range to the very right end of the data, the result may get significantly wrong and the chi-square may get worse. This is due to the very low count rates there, in our example.

If you put the start cursor too far on the left (at values smaller than about 1.3 ns), you will get large errors in that range (and a much worse chi-square), because we selected a theory function without deconvolution. It does not match the decay data well near the excitation, obviously. (Another reason is that we did not match the "T0" parameter; more about this later.)

Now, please play with the radio button in the "Display theory" section of the control window. "Result" means that the curve resulting from the last fit is displayed. "Start values" means that the theory curve belonging to the current start parameters is displayed. "Don't display" turns off the theory curve display.

Enter some different numbers in the parameters start fields and see how the "start values" curve will change. (The curve display is automatically turned off when you start to enter these fields; you need to turn it on afterwards explicitly again.)

Specifically, now do the following: enter the same values as at the beginning. For T0, enter values of 1, 2, 3 ns and observe the theory curve each time. Obvious, right? T0 is simply the time position of the infinitely narrow excitation pulse (remember, we are still without convolution). So, why did T0 stay at its initial value of 0 during the fit? The reason is that *T0 cannot be fitted* (in contrary to the "Shift" when we use convolution). This point is explained in a bit more detail in chapter F4. (Please refer to it.) The main thing you should remember for now is that you simply set T0 to zero and don't care about it any further. It will have no influence on the lifetimes. Most importantly, do *not* set T0 to a value greater than your fitting start value, or you will get very large (and meaningless) chi-squares. (Just try it once now to see the effect: set the fitting range to the interval from 2.0 to 9.0 ns, and set T0 to 3 ns. Hit "Do Fitting" and see the result. The lifetime is fine, but the curve and the chi-square simply don't make sense.)

Finally, try out the "Initial Guess" button. Hit it once, and it will automatically fill out the start value fields for you, using some "reasonable" numbers. This feature will be useful especially when we are doing convoluted fits where it is sometimes difficult to guess the start values. When you use "Initial Guess", be aware of the following: this function does *not* respect your "Fix" flags! This means, it might change a start value even if that value has been "fixed". If you want it to stay fixed at your original value, you need to type in your value again afterwards.

Now, let's try the same with deconvolution. Check the "Do convolution" checkbox. (Remember, if you should encounter some error message box talking about some floating point exception and the like, just ignore it and go ahead. It's normal and shouldn't worry you.)

Re-enter the same start values as before (Offset=0, Shift=0, Amp0=1000, Tau0=1). (Notice that "T0 fixed" has changed to "Shift" when you turned on the deconvolution.) Display the start value curve (remember how?). Fix the Offset, but *don't* fix Shift. Set the fitting range to the interval from 1.0 to about 9.0 ns. (Note that we now let the fitting range start *before* the rising edge of the decay data.) Hit "Do Fitting" and let it run.

You might notice that the fitting calculation now takes longer than before. The reason is the convolution. The convolution integral must be calculated many times for each iteration.

The resultant lifetime should be the same as before (2.0 ns). In contrast to before, however, it is now possible to fit also the *rising* edge of the decay curve.

For fitting with deconvolution, it is important to understand that it is *essential* that the Shift is determined precisely. This usually means you should *not* fix it. Obviously, the chi-square will be very sensitive to variations of Shift since the functions are very steep there and any wrong Shift will cause large residuals.

In order to get some feeling for the sensitivity of the Shift, please do the following. Against our recommendation, please fix the shift at zero. Set the fitting range to the interval from 0.8 to 9.0. Start fitting. The chi-square will be around 1.16 or so; the residuals curve will look almost fine. Now, fix the shift at 0.01, and repeat. The chi-square has increased to about 1.35 (!) and the residuals are clearly distorted. Larger start values will lead to even larger errors. Now, do the same in the negative direction. In this example, you will probably not see much change at values of -0.01 to -0.02 (despite the fact that the first of them actually yields a value even slightly better than 0), but at values of -0.03 and less the residuals will again look clearly distorted and chi-square will increase again.

The main lesson to take home here is: *even very minute changes in the shift value can have large impacts on the quality of the fit*. Accordingly, it is very important to have a correct value for Shift. And this is turn has mainly two implications: (a) Whenever possible, you should obtain the shift as a *result* of the fit and *not* fix it. (b) When you fit the Shift, you should *include the rising edge* of the decay curve into your fitting range. Due to the large slopes there, the iteration will quickly and firmly lock the Shift parameter in its correct position.

In order to test the latter point, you should do one final experiment: Repeat the above fit (with Shift not fixed), but slowly increase your fitting range start value (yellow cursor). When you put it above the peak of the decay curve (at about 1.2 ns), the fitting engine will have much more difficulties to find a good fit and it may take many more iterations than before. (Sometimes, it may even abort and complain about something. In such a case, change the fitting range slightly and try again.)

What we have learned in Lesson 1:

Basic usage of TA-Fit.

How to define a suitable fitting range

How to define suitable start values

The difference between fitting with and without deconvolution

The meaning of T0

The meaning and importance of Shift

Lesson 2

Load another data set now. For the decay data, load "Decay02.PRF" (recall: "Get" button after "Data profile"). For the laser data, load "Laser02.PRF" ("Get" button after "Convolution profile").

Remember: in case you get an error message (talking about "mathematical divergences"), just ignore it for the moment and quit the message via the OK button.

Look at the data display. The data were synthesized with the same set of parameters as in Lesson 1, the only difference being the width of the laser pulse. The shape of the laser pulse was chosen Gaussian during synthesis, in both cases. The FWHM of the pulse, however, was 0.3 ns in Lesson 1 and is 1.0 ns now.

There seems also to be a change in the amplitude of the fluorescence data, since Decay02.PRF has a lower peak than Decay01.PRF. This change, however, is resulting only from the convolution, which causes the fluorescence counts be dispersed over differently wide ranges.

Set the fitting range to 2.0 to 9.0 ns. We assume that "number of terms" is still 1 (mono-exponential), convolution is still ON, and Offset is still *fixed* to zero. (If not, change it now.)

Hit "Initial Guess". Start fitting. You should again obtain a lifetime around 2.0 ns (probably around 1.98 ns).

Lesson 3

Repeat Lesson 2, this time with "Decay03.PRF" and "Laser03.PRF". The only difference to Lesson 2 and Lesson 1 is again the width of the laser pulse. It is 3.0 ns now.

Remember: be careful when you use "Initial Guess". The function may change even those start values that you have fixed. So, in case it changes the start value of Offset, you need to reset Offset to zero before you start fitting.

The lifetime obtained should again be very close to 2.0 ns. Note that the laser's pulse width has been *larger* than the fluorescence lifetime now.

Note: in case the fitting engine "loops" (i.e. it never stops since it cycles periodically between a set of parameters, a condition which can be identified by watching the log in the "Intermediate Results" window), just terminate the iteration manually.

Lesson 4

Repeat Lesson 3, this time with "Decay04.PRF" and "Laser04.PRF".

Now, there are two differences compared to Lesson 3: The laser's pulse width is now 5.0 ns, but the lifetime has also been reduced to 1.0 ns. So, the lifetime is *5 times shorter than the laser pulse*! (Notice how odd the data look in the display window.)

Nevertheless, the lifetime can be reproduced quite well with an error of only about 5%. (Probably you will obtain a value of around 1.05 ns.)

Lesson 5

Repeat Lesson 4, this time with - you guessed it - "Decay05.PRF" and "Laser05.PRF".

The synthesis parameters have been the same as in Lesson 4, but with a lifetime of only 0.5 ns now. It means, the lifetime is *10 times shorter than the laser pulse*.

This represents an extreme case now, demonstrating the limits of what the fitter can achieve.

You will probably find that the fitting behavior now depends much more critically on the choice of the fitting range (experiment with it), and also "looping" and "divergence errors" tend to occur more frequently.

Experiment with different fitting ranges; don't make them too small. Record the lifetime results you get. You will probably obtain lifetimes varying typically between 0.45 and 0.55 ns. This means, the result is varying around the "true" value of 0.5 ns with an error of about +/- 10%. Considering the extreme ratio of lifetime to laser width, this is not a bad result.

Also notice that our absolute amplitudes of the fluorescence data have been very low. The maximum count number in the last example is less than 100. If we had more counts - meaning better S/N ratio - we could achieve even better results.

Lesson 6

Load "Decay06.PRF" and "Laser06.PRF".

The synthesis parameters are the same as in Lesson 1, with two differences: The lifetime was chosen to be 4.0 ns and the laser exhibits a clear "afterpulse". (Real afterpulses may be the result of a badly tuned laser, of trigger jitter or of an unexpected light reflex from some surface.) Notice how the afterpulse clearly distorts the decay curve, causing a "bump" near 3 ns.

Set the start of the fitting range near the rising point of the decay curve, i.e. around 0.7 or 0.8 ns. (You shouldn't choose the start position too early or too late. A start too early would include many channels which are zero or close to zero, a condition which can cause problems in the fitting calculations and can cause a bad chi-square. A start too late will decrease the precision with which the fitter can determine the Shift value.) Set the end position near the end of the curve.

The fit will yield a lifetime very close to 4.0, with an error of much less than 1 %. Notice how the theory curve follows the bump smoothly. Notice, too, that chi-square is very good (i.e. close to 1) and the residuals look good, too (i.e. they are randomly distributed around zero with no obvious systematic deviation).

Lesson 7

Load "Decay07.PRF" and "Laser07.PRF".

This is the same as Lesson 6, with the only difference that a non-zero Shift between the excitation and the fluorescence has been introduced. (Such shifts are frequently encountered in real life, due to various reasons.)

We mentioned the importance of a precise Shift determination already in Lesson 1. In this example, the Shift introduced during synthesis was 0.1 ns.

Set the fitting range properly (see Lesson 6), and start fitting. As expected, we again obtain a correct lifetime within an error of less than 1 %. In particular, recognize how precisely the Shift is reproduced. Its error is in the few percent range, too. (This is an absolute precision of only about a few ps, which is only a small fraction of one single time channel!)

What we have learned in Lessons 2 - 7:

We have seen how the deconvolution yields good results even in cases where the laser pulse is broad and/or distorted.

Again, we noticed the importance of the Shift parameter.

We used chi-square and the residuals curve to check the quality of the fit result.

Lesson 8

Load "Decay08.PRF" and "Laser08.PRF".

Choose a suitable fitting range (from 0.8 to 9.8 ns), and start fitting.

The fitter stops happily after a few iterations but the result looks very bad. The strange shape of the residuals curve with its large "ringing" is a typical and clear *indication for a second lifetime component*.

Actually, the synthesis parameters for this example were: Tau0=1.0 ns, Tau1=5.0 ns, ratio of Amp0:Amp1=10:1, Shift=0, Offset=0, laser FWHM=0.2 ns (Gaussian).

So, let's try a multi-component fit now. Increase the "number of terms" from 1 to 2 (by hitting the right arrow button once). Don't change anything else.

Hit "Initial Guess". (Ignore potential error messages. If necessary, hit the button once more.) If necessary, reset Offset to zero and let it fixed. Start fitting.

The result you obtain should show the following: Residuals look good now. Chi-square is good, too. Tau0 is reproduced with very high precision of clearly better than 1 %. Tau1 is reproduced with an error of about 2 % (maybe around 4.9 ns). The amplitude ratio is reproduced with an error of about 1 or 2 %.

The result is not bad, but why is the precision of the longer lifetime so clearly worse? There are two reasons: First, the two lifetimes are relatively close (yes, 1 ns and 5 ns can be close for fitting, depending on the conditions). Second, the longer lifetime is comparable with the length of our time window. If we wanted to obtain the long component with a higher precision, we would need a bit longer time window.

In order to illustrate the latter point (the first point will be illustrated in the next Lesson), please do the following experiment: reduce the end of the fitting range stepwise by about 1 ns and repeat the fit each time. As you will notice, the error of Tau1 quickly gets larger. As a side effect, also the precision of Tau0 will be affected (but much less), and that of the amplitude ratio, too.

There is something else we can learn from this data set: restore the original fitting range to 0.8...9.8 ns. Start fitting. The result of Tau1 will be about 4.9 ns (an error of about 2 %). Now, repeat fitting, but un-fix Offset first. You will see that the fitter determined a relatively large offset, although it should be zero. Also, Tau1 is now much worse than before. What happened here is that the Offset "mixed" with the long decay component. If there is such a long component, it can become notoriously difficult for the fitter to distinguish it from offset. So, it is *important that you "fix" the Offset*.

In case of photon-counting data, you probably should fix it always to zero, since the background noise is normally neglectable then. In case of data taken in analog mode, you should always try to obtain data with an offset as low as possible (by performing background subtraction and so on). This is always better than letting the fitter calculate it. If there remains a small offset, you should determine its value (e.g. by inspecting the data with the Analysis/Profile tools of HPD-TA), and then fix the Offset parameter to that value.

Finally, we want to remark that Lesson 8 is an example where the lifetimes could also be extracted quite well *without* deconvolution, because the laser pulse is quite short compared to the shortest lifetime and has no distortion

or afterpulse. Please try this now. (Be aware that you need to increase the start position of the fitting range, of course, to a value larger than about 1.2 ns. Also, don't forget to fix Offset to zero again.) You may notice that fitting without deconvolution is much faster.

Lesson 9

Load "Decay09.PRF" and "Laser09.PRF". The synthesis parameters were: Tau0=0.2 ns, Tau1=5.0 ns; ratio Amp0:Amp1=1:1, Shift=0, Offset=0, laser FWHM=0.4 ns (Gaussian).

Please fit this in the same manner as in Lesson 9 (remember to turn Deconvolution ON again). Even though the laser pulse is now broader than in Lesson 9 (actually, it is now twice as large as the shorter lifetime), you will recognize that the fitter's convergence behavior is better. The precision of the obtained lifetimes is very high.

The reason for the different behavior compared to Lesson 9 is that the two lifetimes are now better separated and less "mixing" of them occurs during the fit.

In order to study this mixing more, please load the files of Lesson 8 again. In order to enhance the effect, please use a low position for the end of the fitting range (perhaps around 7 ns or so). Fit it. The result for the long lifetime is poor, as we already recognized in Lesson 8. Now, repeat the fit, but this time enter a start value of 1.0 ns for Tau0 and *fix it*. You will recognize that the result for the longer lifetime is now much better (though still far from perfect). This technique of fixing some lifetime component (which may be known "a priori", perhaps from a previous measurement or from the literature) is often useful for obtaining better results for the other lifetime components.

Finally, load "Decay09.PRF"/"Laser09.PRF" again and try to fit it *without* deconvolution (don't forget to un-fix Tau0). You may face the difficulty that the fitter does not converge with the start values given by the "Initial Guess" function. In this case, experiment with other start values you enter manually. (Try values for Tau0 which are not too far away from 0.2 ns.)

Remember: When you changed a start value, you can visually inspect the theory function corresponding to the new start value set by selecting "Start values" with the radio button below "Show theory".

You will find that it is impossible to obtain a reasonable value for Tau0 without deconvolution. This is because this shorter lifetime is close (even longer) to the laser pulse width. (Recall it was easy to do fitting without deconvolution in Lesson 8.)

Lesson 10

Load "Decay10.PRF" and "Laser10.PRF". This is a *three*-component case now. The synthesis values were: Tau0=0.2 ns, Tau1=1.0 ns, Tau2=5.0 ns, ratio Amp0:Amp1:Amp2 = 1:1:1, laser FWHM = 0.4 ns (Gaussian).

A three-component fit is clearly more difficult than a fit with fewer components.

First, try a *two*-component fit in the usual way. (A reasonable fitting range is 0.6 to 9.8 ns. And, use "Initial Guess" for obtaining the parameter start values. Also, check that Deconvolution is ON.) The resulting residuals curve clearly indicates the presence of a further component.

Increase "number of terms" to 3, and hit "Initial Guess" again. (Ignore possible error messages.)

Try fitting. You will probably find that the fit does not converge with the start values given by "Initial Guess". This means, you need to adjust the start values manually (this is not unusual in those more critical, multi-component cases). In most cases, a good advice is that the start values of the three lifetimes shall be maximally spread (i.e. no two values should be very close to each other); but don't use too short lifetimes. In the actual example, most probably the main problem with the values from "Initial Guess" is that one of the starting amplitudes is negative. If you simply change the sign of that value, the fitter will probably run well.

After the fit, you will find that the fitter found values for the three lifetimes which are quite close to the real ones (with an error of only a few percent), with the exception that the shortest lifetime is a bit off (error of 10 to 20 %). It will probably be difficult to obtain a better value for this short component. If you fix one of the two components, however, the result may improve significantly. (Try this.)

By the way, you may notice that the lifetimes of the fitting result are not necessarily sorted in ascending or descending order (even if the start values had been sorted). This is no fault but a result of the complicated path the fitting iteration may have taken.

Finally, let's try a *three*-components fit on *two*-components data. Re-load the data from Lesson 9. First, repeat the two-components fit to confirm that everything is fine. Then, increase the number of terms again to 3 and try again. (As above, it may require some fiddling with the start values.)

You won't get a meaningful result. The problem is *underdetermined*. Depending on the details (on the your start values and the fitting range), you will encounter one of the following behaviors:

The fitter terminates regularly (unlikely), but two of the obtained lifetimes are close to each other. We say, the result is nearly *degenerated*. (Such a result is always suspicious and should not be over-interpreted.)

The fitter crawls slowly over many iterations and then terminates with a message (last line in Fitting Trace window) saying that there are too many components. This means the fitter has detected that the case is underdetermined and recommends to use less components. However, there is no guarantee that the fitter will detect this.

Same as case (b), but the fitter aborts with a standard error message ("mathematical divergences"). This happens because during the calculations it is very likely that singularities appear. In most cases, you will be able to spot two lifetimes that are nearly the same, if you consult the Fitting Trace window.

What we have learned in Lessons 8 - 10:

How to recognize the presence of additional lifetime components by means of the residuals curve.

How to fit multi-component data correctly.

If two lifetimes are too close, it may not be possible to resolve them.

If the time window is too short, it may not be possible to measure a long lifetime.

If one parameter is known a priori, fixing it may be a good idea.

If the laser pulse is narrow, a trial without deconvolution is worthwhile.

Contributions from long lifetimes and offset are hard to distinguish.

Offset should be fixed whenever possible.

Fitting with too many parameters is useless.

Lesson 11

In this lesson, we want to continue the discussion of cases with closely neighbored lifetimes, since this belongs to the most difficult cases which will happen in practice.

Load "Decay11.PRF" and "Laser11.PRF". The synthesis parameters were: Tau0=4.0 ns, Tau1=5.0 ns, ratio Amp0:Amp1=1:1, laser FWHM=0.4 ns (Gaussian).

Try a two-components fit. You will find it quite hard to reproduce the true lifetimes. The results depend quite critically even on the fitting range. The fitter tends to spend a long time in the iteration, even though the chi-square is already close to 1. (If it does not terminate, stop it manually.)

Probably you can obtain the "true" values if you choose a slightly unusual fitting range (try a start value of around 1.0 ns). But, in practice this is no help, because other fitting ranges would yield different results and you simply wouldn't know *which* one is more trustworthy. (Note that the chi-square as well as the residuals look equally well in all those cases.)

This means, that this example approximately marks the limits of what the fitter can do.

Nevertheless, a trial with a mono-exponential fit quite clearly indicates the presence of a second component. Try this. After fitting, you can see that the residuals curve looks a bit suspicious (some ringing can be seen). This gets clearer if you consult the *autocorrelation curve*. (Hit the "Options" button and enable "Show autocorrelation".)

Next, try a two-components fit on the data set "Decay12.PRF"/"Laser12.PRF". The only difference compared to the previous one is that Tau1 is 3.0 ns now. You will find that the fitting is much easier now and yields non-ambiguous results. The lifetimes are correct within a few percent.

Repeat the above with the data set "Decay13.PRF"/"Laser13.PRF". The only difference now is that the absolute count numbers are ten times lower. The behavior will be similar as before (the fitter tends to spend a bit more iterations, however), but the errors of the lifetimes increased.

Repeat the above with the data set "Decay14.PRF"/"Laser14.PRF". The only difference is that the absolute count numbers are another ten times lower. Now, the fitter will yield lifetimes with large errors, especially for the longer one. (Typical result: 3.3 and 6.1 ns.) The two parameters "mix", and the first one tends to obtain a bigger contribution (larger amplitude), rendering the second one quite wrong. (When you fix Tau0 to 3.0 ns, you still will obtain a quite good result for Tau1. Try it.)

Repeat the above with the data set "Decay15.PRF"/"Laser15.PRF". The only difference is that the absolute count numbers are again another ten times lower (now 1000 times lower than data set number 12). Now, fitting gets really difficult (the fitter frequently aborts with error messages; if necessary, try to stop it manually before it aborts). Also, the results will depend critically on the start values and fitting range. A typical behavior will also be that one of the components tends to get "eliminated", in such a way either (a) that its amplitude gets very small, or (b) that the lifetime gets extremely short, or (c) that both lifetimes "melt together". Try a single-component fit with this data set, too. It works well with a so-so chi-square and yields an intermediate lifetime (probably around 3.9 ns). You will find that the presence of a second component can hardly be spotted in this case (even with aid of the autocorrelation curve), since the data are simply too noisy. A single-exponential fit is probably the best treatment of such data. (Better, yet, of course, would be another measurement with a longer integration time!)

Lesson 12

As a final example, let's try our luck with a data set which looks very weird, due to a quite poor excitation pulse.

Load "Decay16.PRF"/"Laser16.PRF". The synthesis parameters were: Tau0=0.2 ns, Tau1=5.0 ns; ratio Amp0:Amp1=1:1, laser FWHM = 1.0 ns, strongly asymmetric with a very strong afterpulse.

Fit it in the usual manner. The fit works very well and yields very good results with lifetimes errors of typically much smaller than 1 %.

This example shows that a very broad and distorted laser pulse and a "strange-looking" decay curve are no indications for a difficult fit, supposed that the statistics is good enough, the time window is long enough and the lifetime components are sufficiently separated.

What we have learned in Lessons 11 & 12:

How to use the autocorrelation curve.

Whether two closely related lifetimes can be distinguished depends heavily on the noise.

Even if the data are very distorted by the laser pulse, good results can be obtained.

Appendix

A1. Glossary

Theory function
The mathematical model for the physical system or process under study (here: the fluorescence decay). The theory function may contain a set of free parameters.

Least-square method
A popular method for calculating the best match between experimental data and a theory function by adjusting the free parameters of the latter in such a way that chi-square is minimized.

Chi-square
Defined as the sum of the squares of the weighted residuals. During the fitting process, chi-square is the value which the fitting algorithm tries to minimize. At the end of the fitting process, the final chi-square is a measure for the quality of the fit.

Residuals
The differences between the measured and the calculated data points, i.e. between data and theory function. Residuals are usually weighted with the standard deviation of the data.

Convolution
In a mathematical sense, a certain kind of integral relation between three functions. Throughout this document, it refers to the distortion of fluorescence decay curves due to the non-perfect IRF of the experimental equipment (excitation and detection systems).

Instrument response function (IRF)
The response of a detector system to an idealized, infinitely narrow input (here: narrow in time). Since the IRF is never perfect, it will broaden or otherwise distort the output signal compared to the input signal. In this document, the term is used in a generalized way, including also the contribution from the finite (and possibly distorted) pulse shape of the excitation source.

Deconvolution
The mathematical process of reversing the effect of convolution.

Poisson statistics

Statistical behavior commonly observed when counting independent individual events. Counting photons is an example (in this case, the associated noise is also frequently referred to as photon shot noise). For low count numbers, Poissonian noise is asymmetric with respect to the mean value. For higher count numbers, it becomes symmetrical and can be approximated by Gaussian noise.

Gaussian statistics

Statistical behavior very frequently observed in a wide class of phenomena. It is symmetrical with respect to the mean value. The fitting engine of TA-Fit is using a Gaussian statistics model. For reasonable count numbers, it approximates a Poissonian statistics very well.

A2. The Other Theory Functions

As noted earlier, the theory functions other than the multi-exponential are not officially supported. This means, they are believed to work correctly but have not been tested extensively. Despite of this, you may find situations where you want to use these functions (e.g. for analyzing laser pulses or spectral peaks). If you do so, please do it on your own risk.

This section lists the definitions of these other theory functions.

a. Gauss Function

The Gauss function is defined as

$$F(t) = a_0 + \sum_{i=1}^{K} a_i \cdot \exp\left[-\frac{(t-t_{0i})^2}{2 \cdot w_i}\right] \quad \text{(APP1)}$$

where the a_i are the amplitudes, t_{0i} the centers, and w_i the widths of the components. a_0 is a constant offset. The maximum number K of coefficients is limited to 5. Note that the FWHM is related to the width w via the relation $FWHM = 2 \cdot \sqrt{2 \cdot \ln 2} \cdot w \approx 2.35 \cdot w$.

b. Lorentz Function

The Lorentz function is defined as

$$F(t) = a_0 + \sum_{i=1}^{K} a_i \cdot \frac{{w_i}^2}{4 \cdot (t-t_0)^2 + {w_i}^2} \quad \text{(APP2)}$$

where the a_i are the amplitudes, t_{0i} the centers, and w_i the widths of the components. a_0 is a constant offset. The maximum number K of coefficients is limited to 5. Note that the FWHM is equal to w.

c. Polynomial Function

The polynomial function is defined as

$$F(t) = a_0 + \sum_{i=1}^{K} a_i \cdot (t-t_0)^i \quad \text{(APP3)}$$

where the a_i and a_0 are the polynomial coefficients. Their maximum number K is limited to 18.

A3. File Formats

When you save a fitting set via the "Save" button in the Control Window (specifying a filename of <name>.SET), the following seven files will be written to disk.

<name>.DAT	data profile
<name>.CNV	laser profile
<name>.THE	theory profile
<name>.RES	residuals profile
<name>.AUT	autocorrelation profile
<name>.SET	set file
<name>.FIT	special file for export only

The first five are normal HPD-TA profile files. Please refer to the HPD-TA manual or help file for a description of their file format. (Some of these files may not contain data, if the data were not defined when the set was saved; but the files are always created.) The .SET and .FIT files are special ASCII files (see below).

If you load a set file, TA-Fit will look for the other files belonging to the set *according to the filename* (<name>), so you should *not* rename these files.

Note: the purpose of the set files is to offer a way to restore the fitting control window to a state which is exactly identical to that when the set was saved. When you intend to (mis-)use the set files for reading the data by another (perhaps self-made) program, please keep this in mind. (For example, when you do a fit, then change start values but do not start a new fit before you save the set, the set will contain your new start values but the *old* fit results.)

The structures of the .SET and the .FIT files are explained on the next two pages.

.SET file format

The main purpose of the .SET files is saving & restoring fitting sessions as explained on the previous page. Their file format is best explained by means of an example.

Example:

```
1                                                    version identifier
"c:\hpdta\sample01.prf"                              data source (= fluorescence data) (1)
"None"                                               convolution source (= laser data) (1)
"Multi Exponential",0,0,20,1,4,-1,0                  name of theory function
                                                     index of theory function (2)
                                                     handle number of theory function (3)
                                                     maximum number of parameters (4)
                                                     number of terms (5)
                                                     number of parameters
                                                     flag "convolution possible"
                                                     flag "convolution enabled"
56                                                   current fitting range, start (6)
223                                                  current fitting range, end (6)
"Offset",0,0,221.469                                 parameter no.1: name, start value, fix
flag, result value
"T0",20,0,20                                         parameter no.2: name, start value, fix
flag, result value
"Amp 0",1000,0,1517.538                              parameter no.3: name, start value, fix
flag, result value
"Tau 0",30,0,29.13939                                parameter no.4: name, start value, fix
flag, result value
56                                                   used fitting range, start (6)
223                                                  used fitting range, end (6)
.4711337                                             reduced chi-square
-4.789088                                            standard normal variate (Z)
.1429796                                             Durbin-Watson parameter (DW)
60,1000,.0001                                        Timeout, MaxCycles, MinParameterChange
(7)
1.758295                                             excitation pulse display factor
"Sample no. 17"                                      set description string
```

Remarks:

- If data were loaded from a profile file, the string contains the filename. Otherwise, it is "Get Once from Image" or "Auto Updt from Image". (Compare to section D1a). If no data were defined, the string is "None".
- Currently always 0.
- Handles: 0 = Multi-exponential, 1 = Gauss, 2 = Lorentz, 3 = Polynomial
- Currently always 20.
- In this context, "terms" = "components". (Same as "No. of terms" in Fitting Control window.)
- Normally, you won't use the "current fitting range", since it may be *different* (!) from the fitting range used for the currently displayed fit. You should use "used fitting range". The fitting ranges are given in channel numbers.
- These are the stop conditions (see section D1j).

Other remarks:

- The number of lines "parameter no. n" is given by "current number of parameters".
- Boolean flags are false if zero, true otherwise.
- End-of-line is coded as <CR><LF>.

.FIT file format

The main purpose of the .FIT files is data export to third-party programs (such as Origin). They are ASCII files containing six columns.. The first five columns contain the time axis, decay data, fit result curve, residuals, and autocorrelation respectively. The sixth column contains other parameters (see below). The first line in columns 1 to 5 do not contain numerical data, but the strings "Time", "Data", "Fit", "Res" and "Corr".

Columns are separated by <TAB>, and end-of-line is coded as <CR><LF>.

The meaning of the sixth column is as follows:

line	content
1	<unit string> (= the physical time axis unit, e.g. "ns")
2	fitting range start
3	fitting range end
4	number of autocorrelation channels
5	convolution flag
6	chi-square
7	standard variate (Z)
8	DW value
9	number of components
10	results of fit parameters (starting with Offset, T0/Shift, A0, Tau0, A1, ...)
11	" " "
12	" " "
...	....

Note that all strings are without quote marks (in contrast to the .SET files).

Example:

Time	Data	Fit	Res	Corr	ns
0	0	0	0	0	6
0,02	0	0	0	0	445
0,04	0	0	0	0	331
0,06	0	0	0	0	1
0,08	0	0	0	0	1,137536
0,1	0	0	0	0	2,033016
0,12	0	0	0	1	1,904992
0,14	0	0	0	4,965339E-02	1
0,16	0	0	0	8,628935E-02	0
0,18	0	0	0	6,035784E-02	-,0050033
0,2	0	0	0	3,889206E-02	1001,753
0,22	0	0	0	4,339474E-02	1,986684
0,24	0	0	0	6,399346E-02	
0,26	0	0	0	0,1119589	
0,28	0	0	0	3,712217E-02	
0,3	0	0	0	-2,094121E-02	
0,32	0	0	0	-8,010389E-02	
0,34	0	0	0	-4,275264E-03	
0,36	0	0	0	-1,197044E-02	
0,38	0	0	0	-1,061169E-02	
0,4	0	0	0	-7,219172E-03	
0,42	0	0	0	-2,671745E-02	
0,44	0	2,360436E-03	-2,360436E-02	-1,563076E-02	
0,46	0	1,327243E-02	-0,1327243	-2,583158E-02	
0,48	0	0,0295183	-0,295183	-5,999221E-02	
....	....	....	....	....	

A5. Bibliographical Reference

A selected list of publications on various topics covered in this document:

(on least-square method, chi-square fitting, etc.:)

Bevington, Philip R.
"Data Reduction and Error Analysis for the Physical Sciences"
New York: McGraw-Hill, 1969

Press, William H., Flannery, Brain P., Teukolsky, Saul A., Vetterling, William T.
"Numerical Recipes in C"
Cambridge University Press, 1994

Jacobs, David A.H.
"The State of the Art in Numerical Analysis"
London: Academic Press, 1977

McKinnon, A.E., Szabo, A.G., Miller, D.R.; J. Phys. Chem. **83**, 1564-1570 (1977)

Grinvald, A., Steinberg, I.Z.; Anal. Biochem. **59**, 583-598 (1974)

(Marquardt's original paper:)

Marquardt, D. W.; J. Soc. Ind. Appl. Math., (SIAM J.) **11**, 431-441 (1963)

(comparison of different techniques for lifetime analysis:)

O'Connor, D.V., Ware, W.R., Andre, J.C.; J. Phys. Chem. **83**, 1333-1343 (1979)

(on DW parameter, including tables of DW limit values:)

Durbin, J., Watson, G.S.; Biometrika **37**, 409-428 (1950)
Durbin, J., Watson, G.S.; Biometrika **38**, 159-178 (1951)

(on streak photon counting:)

Watanabe, M., Koishi, M., Fujiwara, M., Takeshita, T., Cieslik, W.;
J. Photochem. Photobiol. A: Chem., **80**, 429-432 (1994)

A5. Version History

This section lists the major changes between the different versions of TA-Fit. You need to consult it only if you upgraded from a previous version and want to get a quick overview of the changes.

Note: you can find the version number of your current TA-Fit on the front page of this document and by invoking "Info"/"About" from the program's main menu (TA version is on top, Fit version at the bottom).

TA v 2.1.2 / Fit v 1.0

The "Fix" checkbox behind the parameter "T0 fix" is permanently checked now.

"Quick fit" in the Options window has been eliminated.

The text on the stop button of the Fitting Progress window has been corrected.

Many small corrections in the manual

TA v 2.1.4 / Fit v 1.03

In case of deconvolution, amplitudes are now normalized by the integral over the laser pulse. This yields the same order or magnitudes for the amplitudes regardless whether deconvolution is on or off. The previous chapter F4 of the manual has been deleted.

The previous "Intermediary Results" window has been renamed to "Fitting Trace" window.

TA v 2.2.0 / Fit v 1.03

A new file format (.FIT) is implemented which helps reading fitting results by other programs.

An Origin graph template with macro is provided which allows easy printing of fitting results. This makes use of the new .FIT file types. See the new Appendix A4 for details.

"Recall ROI" buttons have been added in the Fitting Control window which provide the same functionality as in HPD-TA.

Nomenclature: The names of several parameters and options have been changed for better consistency. The changes are:
in the Fitting Control window: "No. of terms" --> "No. of components"; "Max Cycles" --> "Max iterations"
in the Options window: "Show conv. prf." --> "Excitation pulse"; "Show residuum" --> "Residuals"; "Show autocor." --> "Autocorrelation"; "Show log window" --> "Fitting Trace"

"Auto update from image" did not work. This bug is fixed.

The values of final chi-square displayed in the Fitting Display window and in the Fitting Trace window sometimes differed slightly. This bug is fixed.

In contrast to the description in chapter 3a, the first channel of autocorrelation was not excluded in the Fitting Display window. As a consequence, the autozoom of the autocorrelation trace did not produce nice results. This bug is fixed.

The user-defined description string was not saved with the set and consequently could not be restored when loading a set. This bug is fixed. The .SET file format has been extended for this purpose.

TA-Fit tended to issue error messages in situations where it was unnecessary. This bug is fixed.

TA v 3.0.0 / Fit v 1.10

Like HPD-TA, TA-Fit has been ported to 32-bit code. It now runs under Windows NT as well as Win95. It does not run under Windows 3.x anymore.

The "Fitting Trace" window has been equipped with scroll bars.

The "Initial Guess" function yielded the wrong sign for the start value of Shift. This bug is fixed.

Under certain conditions (which depended on the data), TA-Fit overwrote part of its own data. After that happened, TA-Fit then refused to fit data even if the same data could be fitted without problems before. The only help to escape from that condition was to terminate and re-start HPD-TA. This bad bug is fixed.

TA v 3.0.2 / Fit v 1.11

TA-Fit now remembers the user's choice in the "Get profile for fitting" dialog.

TA v 6.4 / Fit v 1.11

The Template and Script for Origin has been debugged and made compatible with Origin v7.0.

TA v 8.1.0/ Fit v 1.20a

Introduced global fitting. Fitting of negative values and absorption data is possible

Index

P

Q

R

S

T

U

V

W

Z