Advanced fitting algorithms for analysing positron annihilation lifetime spectra

Carlos Pascual-Izarra, Aurelia W. Dong, Steven J. Pas, Anita J. Hill, Ben J. Boyd, Calum J. Drummond

This is a preprint of a paper published in NIM A.

The preprint is available at:
http://PAScual.sourceforge.net

The published article is available at:
http://dx.doi.org/10.1016/j.nima.2009.01.205

If you want to cite this work, please cite the published paper:

DOI: 10.1016/j.nima.2009.01.205
Advanced Fitting Algorithms for Analysing Positron Annihilation Lifetime Spectra

Carlos Pascual-Izarra*1, Aurelia W Dong1,2, Steven J Pas3,4, Anita J Hill*3,4, Ben J Boyd2 and Calum J Drummond1,4

1.- CSIRO Molecular and Health Technologies, Private Bag 10, Clayton, VIC 3169 Australia
2.- Drug Delivery, Deposition and Dynamics, Monash Institute of Pharmaceutical Sciences, Monash University (Parkville Campus), 381 Royal Parade, Parkville, VIC 3052, Australia.
3.- ARC Centre of Excellence for Electromaterials Science, Department of Materials Engineering and School of Chemistry, Monash University, Clayton VIC 3800, Australia.
4.- CSIRO Materials Science and Engineering, Private Bag 33, Clayton, VIC 3169 Australia

Authors e-mail addresses: CPI: carlos.pascual@cells.es; AWD: aurelia.dong@csiro.au; SJP: steven.pas@csiro.au; AJH: anita.hill@csiro.au; BJB: ben.boyd@vcp.monash.edu.au; CJD: calum.drummond@csiro.au

Corresponding authors Carlos Pascual-Izarra <carlos.pascual@cells.es> and Anita J Hill <anita.hill@csiro.au>

Abstract
The most common way to analyse PALS spectra involves fitting a parameter-dependent model to the experimental data. Traditionally, this fit involves *local* nonlinear optimisation routines that depend on a reasonable initial guess for the searched parameters. This, together with the fact that very different sets of parameters may yield indistinguishably good fits for a given experimental spectrum, gives rise to ambiguities in the data analysis in most but the simplest cases. In order to alleviate these difficulties, a computer program named *PAScual* was developed that incorporates two advanced algorithms to provide a robust fitting tool: on the one hand, it incorporates a *global* nonlinear optimisation routine based on the Simulated Annealing algorithm and, on the other hand, it yields information on the reliability of the results by means of a Markov Chain Monte-Carlo Bayesian Inference method. In this work the methods used in *PAScual* are described and tested against both simulated and experimental spectra, comparing the results with those from the well established program LTv9. The examples focus on the type of complex data that results from the study of self-assembled amphiphile materials containing co-existing aqueous and hydrocarbon regions.

**Keywords**: Bayesian inference, Simulated annealing, phytantriol, optimization, PAScual

**Introduction**

Positron Annihilation Lifetime Spectroscopy (PALS) is based on measuring the lifetimes of positrons injected into a material [1,2]. When a positron enters a sample, it may undergo different processes leading to its annihilation with an electron from the medium. In some materials, the annihilation might be mediated by the formation of a bound state with an electron, called positronium (o-Ps or ortho-positronium if the electron and positron spins are parallel and para-positronium, p-Ps, when the spins are anti-parallel) [3] [4]. The fact that both the probability and lifetime associated with the different annihilation processes depend on the physico-chemical properties of the medium can be used to
characterise a material. Specifically, the lifetime and relative amount of o-Ps annihilations are sensitive to structural and chemical changes in the material and have been used to characterise free volume [2].

A PALS spectrum consists of a histogram of measured annihilation times of individual positrons in the material and, therefore, it is a decay curve, \( y(t) \), formed by a number (or a continuum) of exponential decay components each with a different intensity and lifetime \((I_i , \tau_i)\), convoluted with the instrumental resolution of the detection system \((R)\) and subject to experimental noise \((B)\):

\[
y(t) = R(t) \otimes \sum_i I_i e^{-t/\tau_i} + B
\]

(1)

The objective of a PALS analysis program is to solve the inverse problem of extracting, from the experimental data, the lifetimes and relative weights of the components forming the spectrum. This solution can be attempted either by a direct deconvolution of the spectrum or by fitting a theoretical model to the experimental data. Some existing programs such as CONTIN [5,6] or MELT [7] follow the direct deconvolution approach, while POSITRONFIT [8], PALSfit [9,10] or LTv9 [11,12] use the model fitting method.

Regardless of using one or the other approach, the main difficulty in solving the inverse problem is that it is a very ill-posed one [7], making the solution non-trivial and ambiguous: very small variations in the experimental data (such as those caused by random experimental noise or data collection artifacts) lead to very large variations in the solution. Conversely, many (even infinite) different solutions may exist to a given problem that are equally valid in terms of their consistency with the experimental data. In order to deal with this difficulty, the different programs use various regularization strategies such as the Maximum Entropy principle in MELT [7] or a preference for smooth solutions (CONTIN [5]) or using a model with fixed number of components as in POSITRONFIT [8] and LTv9 [11] or introducing more constrained models specifically tailored to the system under study (such as the ETLA model [13] for polymers). Even when these regularizations alleviate to a certain extent the problem, they cannot eliminate all of the ambiguity in the results and, in practice, the user needs to decide amongst a set of solutions. The bias from the user can be very explicit (as in the choice of the number of components to
model) or more indirect (e.g., the initial guessing for the fitting parameters or the choice of the regularization weight in MELT/CONTIN).

In this work a program named PAScual [14] is presented that implements (and improves) some of the existing approaches and also introduces two advanced model fitting algorithms (Simulated Annealing and Markov Chain Monte-Carlo Bayesian Inference) that address the problem of the user bias on the results. Figure 1 shows an illustrative screen capture of PAScual’s main interface window, and more details are given in the supporting information.

PAScual is a model fitting program that provides a choice of various fitting algorithms (the details are given in section 3). When using its local optimisation routines, PAScual behaves in a very similar way to that of programs like LTv9 or PALSfit: first, the user loads one or more spectra, chooses the region of the spectrum to be fitted and decides how many components to include in the analysis; then the user provides the program with initial guesses on the various fitting parameters and launches the fit which, on a modern computer, would typically take only a fraction of a second of calculation time per spectrum; the user would then examine the solution(s) and, if not satisfied, modify the initial guesses and/or number of components and launch the fit again. Ideally, this iterative process should continue until it is clear that any effect on the results caused by user decisions are either ruled out or taken into account.

While this approach is fine in some situations (mostly when dealing with relatively well known samples or when performing a preliminary screening), in some other cases it may be tedious and be limited by the skill, judgement and patience of the user. It is in these cases that the Simulating Annealing (SA) and the Markov Chain Monte-Carlo Bayesian Inference (MCMC-BI) algorithms are an interesting alternative and/or complement. Both SA and MCMC-BI require much longer calculation times than the local optimisation routines but the solution is robust and independent of the initial guesses. The MCMC-BI automatically explores the distribution of solutions that are compatible with the experimental data and is able to provide statistically meaningful information of such distribution (it is even possible to reconstruct the solution space near the optimum). These characteristics drastically
change the way the analysis is performed: the analysis can be run unattended even for unknown samples and the computer takes care of the exploration of the possible solutions, providing confidence estimations that take into account the quality of the experimental data.

In the following Sections the model and the optimisation algorithms used in \textit{PAScual} are described and some examples of analysis both of real and simulated spectra are discussed, validating \textit{PAScual} against existing codes. Finally, a detailed study is presented on the complexities of analysing PALS data from self-assembled amphiphiles, setting the grounds for decisions made when analysing phytantriol/water results published elsewhere [15]. The supporting information for this work provides further technical details on the \textit{PAScual} code as well as a description of the program’s features that may be of interest to potential users.

\textbf{Theoretical Model}

As mentioned, \textit{PAScual} solves the inverse problem by fitting a model to the experimental data provided. The model consists in a simulation of the PALS spectrum from a set of free parameters that are automatically varied until the calculated spectrum matches the experimental one. The simulated spectrum is essentially a sum of an arbitrary number of individual component that can be expressed as:

\[ f_k = B_k + \sum_i^N \sum_k^N M_{ki} a_i \]

where \( f_k \) is the calculated number of counts in the \( k \)th channel of the spectrum, \( B_k \) is the background (commonly assumed to be constant over all the channels, but not necessarily so) and \( a_i \) are the relative probabilities of each component of the spectrum. It is interesting to note that percent intensities \( I_i = 100 \times a_i / \left( \sum_j a_j \right) \) are used instead of \( a_i \) for the interaction with the user. \( N \) is the number of counts over the region that is fitted (not counting the background) and can be calculated from the experimental spectrum \( (Y_k) \) as: \( N = \sum_k (Y_k - B_k) \). The elements denoted by \( M_{ki} \) form a matrix whose columns are the component vectors. The explicit formula for each component depends on the type of decay and the
function describing the instrumental resolution. The current version of PAScual [14] (at the moment of writing it is v1.3), implements the simple case of components formed by single exponentials convoluted with a Gaussian, as in reference [8]:

\[
M_{ki} = \frac{1}{2} [\Psi(\sigma, \tau_i, t_k) - \Psi(\sigma, \tau_i, t_{k+1})], \text{ where}
\]

\[
\Psi(\sigma, \tau, t_k) = \exp\left(\frac{\sigma^2}{2\tau^2} - \frac{t}{\tau}\right) \left[1 - \text{erf}\left(\frac{\sigma}{\sqrt{2}\tau} - \frac{t}{\sqrt{2}\sigma}\right)\right] - \text{erf}\left(\frac{t}{\sqrt{2}\sigma}\right),
\]

where \(\tau_i\) is the characteristic decay time for the \(i\)th component, \(\sigma = \text{FWHM}/(8\ln 2)^{1/2}\) is the standard deviation of the instrumental function and \(t_k\) is the time at the boundary between channel \(k-1\) and channel \(k\). The value of each \(t_k\) depends on the settings of the instrument, but in general it is safe to assume a linear response from the spectrometer and, therefore, the channel to time conversion can be safely done as \(t_k = (k-k_0)g\), where \(g\) is the gain or “channel width” of the spectrometer (typically measured in ps/channel) and \(k_0\) is the channel offset (often called “time zero channel” or “channel-0”), which is not necessarily an integer number. While this version only includes support for components of the type described by eq. (2), other shapes including composite resolution functions or Gaussian distributions of lifetimes [11] can easily be added without affecting the discussion.

**Optimisation methods**

I. General characteristics of the methods used. The model described by Equations (2) and (3) ultimately depends on the following parameters: the \(a_i\) and \(\tau_i\), \(\sigma\), the background, the channel width and the channel offset. These parameters may either be well known from theory or previous measurements or only approximately known or not known at all. The objective of the program is to recover the information on the parameters that are not well known by finding a set of them that leads to a simulated spectrum matching the experimental data. This is accomplished by choosing an objective function that quantifies the “matching” or goodness of fit and optimising that function with respect to the unknown
parameters. Different optimisation methods may use different objective functions but most will, at least, express the final “goodness of fit” by using the $\chi^2$ estimator:

$$\chi^2 = \sum_k \left( \frac{Y_k - f_k}{\Delta Y_k} \right)^2,$$

(4)

where $Y_k$ is the number of counts in channel $k$ for the experimental spectrum and $\Delta Y_k$ the corresponding uncertainty in the measurement. In the case of the PALS spectra, where each channel can be considered an independent measurement following Poisson statistics, the uncertainty for each channel is well approximated by $\Delta Y_k = \max\{2, Y_k^{1/2}\}$. As a fit tends to its optimum solution (in terms of $\chi^2$), the value of $\chi^2$ tends towards its mean value, which is given by the degrees of freedom of the fit: $\kappa-n_F$, where $\kappa$ is the number of fitted channels and $n_F$ is the number of free parameters in the fit. For a more convenient comparison of different fits, a normalised form of the $\chi^2$ is used: $\chi^2/(\kappa-n_F)$.

Similarly to programs such as POSITRONFIT or LTv9, the user can choose between fitting a given parameter or assigning it a fixed value. In addition, PAScual also offers the possibility of setting boundaries to free parameters, ensuring that the solution lies within these boundaries. This allows the user to instruct the program to not explore non-physical regions of the solution space (e.g. negative intensities or lifetimes larger than 142 ns). Moreover, it can be a powerful tool to force the program into fitting relatively weak components that would otherwise be ignored because of the influence of the more intense components. Of course, this kind of tool must be used with care to avoid unwanted bias in the solution.

Another fitting option available to the user is the possibility of linking parameters of different spectra. This is similar to the “global parameters” in LTv9 but offers more flexibility, since it allows the user to establish the link only for selected parameters of selected spectra (as opposed to all the spectra). The spectra linked by common parameters (either directly or through links formed by intermediary spectra) will form a so-called “fitting set” that is fitted as a whole to ensure a congruent solution. As before, this tool is very powerful but it must be used with care: if there are too many independent parameters
compared with the number of linked parameters, the dimensionality of the solution space may be too large to be solved. As a rule of thumb, only those spectra with most of their free parameters in common with each other should be fitted in a linked set.

In *PAScual*, when the user loads and sets the spectrum parameters, the spectra are automatically distributed in fitting sets (if no common parameters have been used, each spectrum will form its own separate “fitting set”). Each fitting set will be analysed in turn and the user can choose whether the solutions from one analysis should be used to initialise the following one or not. This allows the automated fitting of large numbers of spectra regardless of whether the measurements are related or not: in the case of related measurements in which one can reasonably expect smooth variations of the parameters from one fitting set to another, it makes sense to initialise the current fit using the last fit results, but if the premise is not true, relating the fits might not be desirable. Similarly, using non-connected analyses may be useful in avoiding artificial trends in the results produced by a progressive refinement of the fit over various spectra. A further advantage is obtained by this flexibility: each fitting set can be initialised with totally independent choices (e.g. on the number of components, on the initial guesses or boundaries for the parameters, etc.). This, combined with the possibility of loading the same spectrum as many times as desired, provides an efficient way of testing different approaches to the analysis of a given spectrum in a single run.

The main characteristic of *PAScual* is, perhaps, the possibility of choosing and/or combining various optimisation methods (local, SA, and MCMC-BI). The details for each one are discussed in the following sections.

**II. Local optimisation.** The multivariate non-linear optimisation algorithms are considered local if they only explore solutions relatively close to the initial guess, with the risk of getting trapped in local optima instead of finding the global optimum. In return, these algorithms tend to be much faster than global ones.

*PAScual* relies on two routines for local optimisation provided by the Scipy module [16]: one is an implementation of the Levenberg-Marquardt Algorithm (LMA [17]) and the other is the L-BFGS-B
algorithm [18]. Both algorithms are based on minimising the $\chi^2$ value shown in eq. 4. The LMA is relatively robust and fast and provides an estimation of the covariance matrix for the result, but it does not perform a constrained optimisation (this algorithm is the same one used by POSITRONFIT and LTv9). The L-BFGS-B, in turn, is slightly slower and does not calculate the covariance matrix, but it supports box-type parameter limits. By default, when performing a local optimisation, PASCual first attempts a fit using LMA and then checks if the solution lies within the acceptable boundaries in which case, the optimisation is complete. If the result from the LMA is not acceptable, a sequence of L-BFGS-B + LMA + L-BFGS-B is performed: the first L-BFGS-B provides a reasonable bounded initial guess to the LMA; this one refines the L-BFGS-B search (and, since it is initialised already close to an optimum, it is quite probable that it does not jump too far away); the final L-BFGS-B ensures that the solution is indeed still within the limits (if the LMA did not jump out of the bounds, the convergence is very fast).

In the absence of bounds, this scheme works just as in the LTv9 or POSITRONFIT (giving virtually identical results) while, if bounds are required, it handles them.

III. Global optimisation: Simulated Annealing. The Simulated Annealing is a general-purpose global optimisation algorithm [19] which employs a Monte-Carlo calculation. Its adaptability has led to its application across very diverse fields which only have in common that a global optimisation is required (see, e.g., references [20-23]). Previous successful experience in applying SA to the analysis of data from other nuclear spectroscopies [24-26], encouraged us to try a similar approach for PALS. PASCual is, to our knowledge, the first code using SA (and MCMC-BI) to solve the inverse problem for PALS.

To understand the SA algorithm one should consider the multidimensional solution space formed by the combination of all allowed values of the free model parameters. The SA will explore that space in a series of steps, starting from a given point, $s$, which represents a state of the values of the model parameters. In each step of the SA, a transition to a neighbouring state $s'$ is proposed and the algorithm decides whether or not to accept the transition following a probabilistic criterion based on the relative value of the objective function in the current and proposed states. In this case, the $\chi^2$ is chosen as the
objective function and the Metropolis-Hastings criterion [27] is used for calculating the probability of acceptance of new states:

$$\alpha(s \rightarrow s') = \min\left\{1, \exp\left(-\frac{\Delta \chi^2}{T}\right)\right\},$$

(5)

where \( \alpha \) is the probability of acceptance for the transition, \( \Delta \chi^2 = \chi^2_{s'} - \chi^2_s \) is the change in the objective function if moving from \( s \) to \( s' \) and \( T > 0 \) is the so-called control parameter. The Metropolis criterion implies that every proposed transition leading to a decrease of the objective function will be accepted \((\Delta \chi^2 < 0 \Rightarrow \alpha = 1)\), but it also gives a certain acceptance probability to those proposed transitions that lead to an increase in the objective function. The value of the control parameter is varied from very high values in the initial stages of the SA to very low values towards the end. This variation is the key to the ability of SA to escape from local minima, since initially most of the proposed transitions are accepted (even those that lead to large increases on the objective function) while, when \( T \) reaches very low values, practically only those transitions that decrease the \( \chi^2 \) get accepted.

In our implementation of the SA, the transitions are proposed by slightly modifying the current state \( s \) by adding two numbers to one of the free parameters, one of which is determined from the current \( s \) itself and the other is a random component. More details about the transition mechanism are given in Subsection V. Regarding the variation of the control parameter, \( T \), its value is initially increased until an effective randomisation of the initial state has occurred (thus completely losing the memory of the initial guesses). From that moment on, \( T \) decreases whenever a certain number of accepted transitions is reached. The decrease of \( T \) follows a geometric progression towards 0.

An interesting property of SA is that it can be proven that, given the above conditions, as the number of accepted transitions before stopping the SA increases, the probability of having reached the global optimum tends to 1. In practical situations, though, the finite calculation time imposes a limit in the number of transitions and therefore, while the SA is a very well-behaved global optimisation algorithm, it is not infallible. Moreover, while SA is very efficient in getting close to the global optimum ignoring
local optima, its convergence towards the optimum is relatively slow once it is close to it. These characteristics make it a good complement to a local optimisation algorithm: the SA can be used to get a robust approach to the surroundings of the global optimum at which point a faster local optimisation can be safely launched to refine the fit.

**IV. Confidence bars determination: Bayesian Inference.** Even if the local optimisation algorithm (or the SA) succeeds in avoiding the local minima, the result is just a single point of the solution space which is very close to the global optimum. When dealing with real-world noisy data, there are many different possible near-optimal solutions which are all compatible within the noise level of the data and the tolerances of the experimental parameters. Therefore, the SA or local optimisation does not provide information about the uncertainty associated with the result. Even when, as seen in the LMA case, the covariance matrix can sometimes be estimated, its calculation is not always robust for these kinds of problems and, therefore, should only be used as a qualitative indicator. The MCMC-BI algorithm addresses the problem of characterising the distribution of near-optimal solutions that can result from fitting a given experimental spectrum. This distribution, which is known as the *posterior* distribution, can be expressed in terms of probability as: \( p(s \mid Y) \), i.e., the conditional probability associated with each possible solution given the experimental data. By calculating the posterior distribution, one can extract any estimator that characterises the near-optimal solutions, for example (but not limited to) the mean value for any parameter or its standard deviation. The posterior distribution is calculated using the Bayes’ theorem [28]:

\[
p(s \mid Y) = \frac{p(Y \mid s)p(s)}{p(Y)},
\]

(6)

where \( p(s) \) is the *prior* distribution and represents the *a priori* knowledge (if any) about the solution (note that it does not depend at all on the experimental data). The prior may include, for example, information about boundaries on the fitting parameters or correlations between fitting parameters. The denominator, \( p(Y) \), does not depend on the fitting parameters at all and can be regarded as a scaling constant that disappears when imposing the normalisation of \( p(s \mid Y) \). Next, there is a need to calculate
$p(Y|s)$, which is the *likelihood* distribution of the data. In the case of a PALS spectrum with Poisson noise, the likelihood distribution is:

\[
P(Y | s) = \prod_k \frac{\exp\left(-\frac{1}{2} \chi^2_k \right)}{\sqrt{2\pi\Delta Y_k}}
\]  

(7)

To estimate $p(Y|s)$, a Markov chain Monte-Carlo method is used [29]. In this context, a Markov Chain is a sequence of states formed in such a way that the probability of accepting a new state depends exclusively on the new and current states. The Metropolis-Hastings algorithm is used to generate a Markov chain whose equilibrium distribution is a good sample of the likelihood distribution around the optimum. If the proposal of transitions is symmetrical (i.e. the probability of proposing a transition from $s$ to $s'$ is the same as the probability of proposing a transition from $s'$ to $s$), the Metropolis probability of accepting a proposed transition is:

\[
\alpha(s \rightarrow s') = \min\left\{1, \exp\left(\frac{p(Y | s')}{p(Y | s)}\right)\right\}
\]  

(8)

In order to properly estimate the distribution, a certain number of initial steps of the Markov chain are discarded to ensure that the equilibrium distribution has been reached. The remaining states in the Markov chain constitute a sample of the likelihood distribution from which statistical information can be inferred. Since the accuracy of the information is better the larger the sample, generating long enough Markov chains is the most computationally demanding aspect of the MCMC-BI algorithm.

The choice of the Metropolis-Hastings criterion for both the MCMC-BI and the SA algorithms implies that the algorithm needed to generate the Markov chain for the MCMC-BI is a special case of the SA, in which the control parameter is fixed at a value $T=2$. Therefore, if the MCMC-BI is run after an optimisation, it can be understood simply as a very long walk around the optimum, sampling a lot of potential solutions. Due to the way the Markov chain is constructed, the states that are more likely to be good solutions are more represented in the sample so one can infer the statistical significance of the solution found.
It is important to note that the MCMC-BI is a completely different approach to that used in MELT (even if both make use of Bayes' theorem). In the case of MELT, the Bayesian Inference is used in the context of finding an optimal regularisation to a direct deconvolution method [7], whereas in PAScual Bayes' theorem is used to calculate the posterior probability distribution from the likelihood distribution estimated by a Monte-Carlo Markov Chain method.

V. Optimised Monte-Carlo transition algorithm. Both the SA and the MCMC-BI algorithms need to propose transitions from the current state to a new one. There are almost no fundamental restrictions in how the candidate state is chosen except that, for MCMC-BI, the transition must be symmetrical (otherwise eq. (7) would require an additional factor). However, the efficiency of both SA and MCMC-BI is greatly affected by the scheme used for proposing transitions: since the computationally-expensive part of both algorithms lies in the calculation of the $\chi^2$ (which is done regardless of the proposed state being accepted or not), it is convenient to propose transitions that have a large probability of being accepted. When the current state is near the optimum, this generally means proposing new states that are not too far from the current one and, if possible, in a direction that reduces the $\chi^2$. On the other hand, if one proposes states that are too close to the current one, many more steps will be needed to sample the solution space appropriately.

The transition scheme used in PAScual is similar to that introduced in reference [30] and consists of modifying one or more free parameters using the following Gaussian random distribution: $r'=N(r+u dr, \sigma_{dr})$, where $r$ and $r'$ are, respectively, the current and new values of the free parameter, $u=1$ for SA and $u=0$ for MCMC-BI and $N(x,y)$ is a normal distribution centred in $x$ with standard deviation $y$. The quantities $dr$ and $\sigma_{dr}$ are calculated assuming that, for a very small perturbation of $r$ (denoted by $dr$), $f_k$ is approximately linear and $r$ is not correlated with the other parameters. In this situation, the value of the $\chi^2$ after the perturbation is:
\[ \chi^2 = \sum_{k} \left( \frac{Y_k - \left( f_k + \frac{\partial f_k}{\partial r} dr \right)}{\Delta Y_k} \right)^2, \]  

(9)

where the partial derivative of \( f_k \) with respect to \( r \) is calculated analytically for certain parameters and numerically for others. By minimising eq. (9) with respect to \( dr \), one can find the optimum value for \( dr \) (and its associated uncertainty, \( \sigma_{dr} \)):

\[
dr = \frac{Z_r}{S_{2,r}} \quad \text{and} \quad \sigma_{dr} = \left( \frac{S_{0,r}}{S_{0,r} S_{2,r} - S_{1,r}^2} \right)^{1/2}, \quad \text{where} \]

(10)

\[
Z_r = \sum_k \left( \frac{(Y_k - f_k)}{\Delta Y_k^2} \frac{\partial f_k}{\partial r} \right) \quad \text{and} \quad S_{n,r} = \sum_k \left( \frac{1}{\Delta Y_k^2} \left( \frac{\partial f_k}{\partial r} \right)^n \right) \]

(11)

The benefits of calculating \( dr \) and \( \sigma_{dr} \) for every single step of the Monte-Carlo may not compensate for the computational cost involved. Therefore, these quantities are recalculated only after a certain number of steps which is dynamically adjusted depending on the relative variations of \( dr \) and \( \sigma_{dr} \) with respect to their preceding values.

**Results and Discussion**

**I. Test with experimental spectra.** In order to show the suitability of PAScual as a general tool for analysis of PALS data, experimental data from a variety of materials representative of the different typical applications of PALS were analysed. Table 1 shows the lifetimes obtained for free positrons in annealed aluminium, as well as for o-Ps in a set of three well studied polymers: polymethyl methacrylate (PMMA), polycarbonate (PC) and polysulfone (PSF).

In all cases, the contribution of the annihilations in the \(^{22}\text{Na} \) positron source was taken into account (which amounted to only 3.7% of the total). Apart from the source correction, a single component analysis was performed for aluminium while for the polymers, three components were fitted, and the
longest-lived one (i.e., the one associated with o-Ps) is compared to literature values as reported in Table 1. The results obtained agree well with those found in the literature [31-34].

Apart from the results quoted in Table 1, further validation of the performance of PAScual for analysing complex experimental spectra can be found in [15].

II. Test with computer-generated spectra. By using computed generated spectra instead of experimental ones the program can be validated in well controlled scenarios. Different series of generated spectra were used in order to test various aspects of the PAScual program.

The first tests were aimed at establishing a general comparison with existing codes as well as to check the performance of the global optimisation algorithm. In order to do so, the conditions proposed by Shukla et al [7] for testing MELT (which were also used by Kansy [11] to validate LTv9) were reproduced. They consisted of four series of spectra, labelled A to D, whose parameters are given in Table 2. The spectra have been analysed with both PAScual and LTv9 for reference. In the case of PAScual, the fit was performed using both local optimisation and Simulated Annealing. In the case of limiting PAScual to performing just a local optimisation, the results were indistinguishable from those obtained by using LTv9, which is not surprising given that no constraints were used and, therefore, both programs use very similar optimisation algorithms. The errors quoted in Table 2 correspond to the standard deviations of the parameters found in the analysis of the different spectra in each series. From Table 2 it is evident that the simulated annealing algorithm is less accurate but more precise than the local optimisation. This is not surprising given the already discussed characteristics of the SA algorithm: the slow convergence towards the optimum during its final stage is responsible for the relatively poor accuracy, while its ability to find the region of the global optimum regardless of the initial guess accounts for the narrower variability in the results from different spectra. The combination of both algorithms is a good option when the local optimisation yields erratic results due to the lack of good initial guesses.

Although the best method for estimating the uncertainties associated with the results is to study the variability in the parameters obtained from various independent spectra (as done in Table 2), in many
situations experimental constraints such as sample degradation or geometrical limitations may prevent the acquisition of a sufficient number of spectra comprising enough events. In such cases, one needs to estimate the uncertainties in the results from a single or a few fits. On the one hand, the local optimisation method allows the calculation of the covariance matrix from which the uncertainties in the parameters can be estimated provided that the solution space can be approximated by a parabola near the optimum [9]. On the other hand, the MCMC-BI algorithm allows the reconstruction of the solution space in the general case, regardless of it being well behaved, at the expense of longer calculation times. Figure 2 shows various projections of a 9-dimensional solution space sampled by the MCMC-BI method. The sampling consists in 9 million of near-optimal solutions for the fit of a 3 component spectrum (nine free parameters were used for this fit: 3 intensities, 3 lifetimes, channel offset, FWHM and background). In each panel, the frequency of sampled solutions as a function of two chosen free parameters is shown, revealing the interrelations among them. In this case the solution space is mostly unimodal within the vicinity of the optimum. The extension of this “vicinity” (i.e., the 9-dimensional volume actually sampled by the MCMC-BI) is defined by the randomness allowed by the statistical noise in the analysed spectrum.

Comparing Figures 2a and 2b, it can be observed that the intensities for the two shorter components are heavily correlated while the intensity for the third component (which would be associated with o-Ps in a real spectrum) is quite independent. Likewise, the correlation between the intensity and the corresponding lifetime is much weaker in the case of the longest-lived component, as shown by comparing Figures 2c and 2d. The relatively good behaviour of the fit of the longest-lived component is very relevant to the analysis of spectra from any material where the o-Ps components are the ones used to gain information about the nanostructure of the sample.

The ability to access and visualise the solution space also opens the possibility of choosing from a range of statistical descriptors for the solution. The average and standard deviation over all the sampled solutions are used as results and uncertainties, respectively, but other estimators with different meanings
can be used. For example, one could be interested in the solution yielding the best $\chi^2$ value amongst all the sampled states, or the mode in any of the projections of Figure 2. Finally, in some complex cases, the sampled solution space can be used to filter out non-physical solutions and then recompute the results from the filtered space, as it is often done when analysing data from high energy physics [35].

Another very powerful tool implemented in PAScual is the possibility of fitting different spectra simultaneously (i.e., in the same fitting set), by imposing the condition that one or more parameters be common among the spectra. Table 3 compares the results from independently fitting one single spectrum from each of the B, C and D series (described in Table 2) using three different choices for common parameters. It becomes evident that using common parameters greatly improves the overall accuracy compared to performing independent fits. This is especially significant for the case of the D spectrum, which is the most problematic to fit as already seen from Table 2.

As the last test, a situation was reproduced similar to the analysis of a series of spectra from a thermal phase transition experiment in a self-assembled amphiphile system. For this series the spectra were generated using values for the lifetimes and intensities that roughly match those observed for real spectra from the phytantriol/water system (88/11 w/w %), when varying temperature from 0 K to 325 K. A total of 250 spectra were created (5 for each temperature, in steps of 1 K). The parameters used for constructing the spectra, as a function of the temperature are:

$$\tau_1 = 125 \text{ ps}, \quad \tau_2 = 400 \text{ ps}, \quad \tau_3 = \begin{cases} 886 \text{ ps} + 6.91 \text{ ps} \cdot K^{-1} \times T & \text{for } T < 298 K \\ 2774 \text{ ps} + 0.544 \text{ ps} \cdot K^{-1} \times T & \text{for } T > 298 K \end{cases} ,$$

$$I_1 = 20 , \quad I_2 = 100-(I_1+I_3) , \quad I_3 = 7.52 + 0.469 \text{ K}^{-1} \times T ,$$

$$B = 250 \text{ counts} , \quad k_0=100 , \quad \text{FWHM}=270 \text{ ps} , \quad g=58 \text{ ps} , \quad A=5\times10^5 \text{ counts}$$

The so generated spectra were analysed in series using different approaches in the way the data are handled. The first approach is to fit the spectra independently and average the results for each temperature (Figure 3 shows the results obtained with this approach). The second approach is to group the spectra for each temperature and fit them using common parameters for everything but $B$ and $k_0$. The
third method for analysing consists of summing the spectra for each temperature and analysing the sum. In all cases, the fitting region of interest was limited from 5 channels to the left of the maximum to 800 channels to the right of the maximum, and the results from each fit were used to automatically initialise the next fit. Local optimisation was used in all three approaches and, in the case of the second one, optimisation using MCMC-BI was also performed to obtain a better description of the uncertainties, as shown in Figure 4.

Since the spectra are computer generated and, therefore, the “real” values of the parameters are known, it is possible to quantify the overall quality of the results for the different approaches to the problem. For example, the average absolute deviation from the known value of $\tau_3$ is 17.1 ps for each of the three approaches (when using local optimisation) as well as for the results using LTv9 on the same series of spectra. The average absolute deviation from the known value of $I_3$ is 0.07% for all methods (and for LTv9 as well). When using Bayesian Inference, the results were slightly better: 16.8 ps and 0.07% for the average absolute deviations of $\tau_3$ and $I_3$, respectively. Also, the estimation of the errors is better in the case of the MCMC-BI: for example, the error bars for $\tau_3$ in Fig 3 (local optimisation) are, on average, 54 ps, and they vary quite a lot from value to value, while in Figure 4 (MCMC-BI) they are more uniform and their average of 24 ps is a lot closer to the dispersion of the data from the “real” values (which is, as seen before, about 17 ps).

The fact that all three methods are similarly accurate indicates that the choice can be made based on their other relative strengths: whenever it is possible, it is recommended that one fits spectra independently and averages the results using the standard deviation as an estimation of the uncertainties. If this is not possible (because of the lack of sufficient data), the use of common parameters seems the best alternative since it is flexible and yields very similar results. The analysis of the summed spectra is the last option (to be used when there is only data for one or two spectra) because it offers similar results to the “common parameters” approach and is less reliable in situations where some parameters may change from one spectrum to another.
An important observation from Figure 4 is that in a situation where the “real” values are unknown, one could be tempted to concede significance to apparent patterns in the data that are not really present. For example, the somewhat downwards trend of $\tau_3$ in the region between 296 K and 310 K could be thought to be reflecting a structural change occurring in the sample (when, in fact, the data were constructed using a slightly positive slope for the lifetimes in that region). In relation to this, it could be hypothesised that using the results from the previous fit to initialise the next one could create spurious trends in the results. However, when repeating the fits without that option (i.e., by initialising all the fits with the same guess), the results are identical to those of Figure 4, ruling out the aforementioned hypothesis in this case. It is therefore important to be cautious when interpreting similar data from non-generated spectra and only trust trends that are clearly above the uncertainties associated with the fitted parameters.

III. Deciding the number of components in the fit. One of the most important aspects influencing the correctness of the PALS fit results is the choice of the correct number of components used in the model. On the one hand, adding extra components introduces enough degrees of freedom in the model to fit any arbitrary spectrum although the solutions, while mathematically correct, may not be physically sound. On the other hand, sometimes the data may be a composition of components too close to be resolved within the finite statistics of the experimental spectra and, consequently, one single component in the model might be used to account for them. The traditional approach to this problem is to use as few components as necessary to obtain a reasonable fit in terms of both the $\chi^2$ parameter and the residuals of the fit (a non stochastic pattern in the residuals may indicate a missing component in the model) while accounting for any component that may be expected from physical considerations.

In order to help in the decision of the adequate number of components for a given system, it is very convenient to gather information about the expected behaviour of the fit in the different possible scenarios. This is possible by using computer-generated spectra that consider the various hypotheses and then testing the different fitting approaches. By comparing the results of the fit with the known parameters used for the generation, the effects of the options in the analysis can be better understood.
As noted in the previous section, the behaviour of the fit depends heavily on several factors such as
the statistics in the spectra, the number of components and their distributions, the number of channels,
etc. For this reason, it is convenient to generate spectra in conditions close to those in the experimental
problem that one is trying to understand.

As an example, a self-assembled amphiphile system (phytantriol/water) has been chosen (see [15] for
more details) For this particular system it is important to decide whether to use 3 or 4 components in the
analysis of the spectra. Fitting with a four component model would account for two separate o-Ps
components (associated with the annihilation of o-Ps in water and in lipid), while a three component
model would force the fit to accommodate all o-Ps contributions in a single component.

In order to test the hypothesis that o-Ps annihilation can occur within water channels as a separate
phase in the amphiphiles, a series of spectra was generated with the following parameters:

\[ \tau_1 = 125 \text{ ps}, \tau_2 = 400 \text{ ps}, \tau_3 = 1800 \text{ ps}, \tau_4 = 3000 \text{ ps}, \]
\[ I_1 = 20, I_2 = 55, I_3 = x \cdot 25, I_4 = (1-x) \cdot 25, \text{ where } x \text{ was varied from 0 to 1} \]
\[ B = 125 \text{ counts}, k_0 = 100, \text{ FWHM} = 270 \text{ ps}, g = 58 \text{ ps} A = 2.5 \times 10^6 \text{ counts} \]

The third component corresponds roughly to the o-Ps in water, while the fourth component represents
approximately the o-Ps in pure phytantriol. By varying the relative intensity of these two components
along the generated spectra, every possible situation from a pure "phytantriol" spectrum to a pure
"water" spectrum is explored.

Each spectrum of this series was fitted first using four components (see Figure 5) and then using only
three components (Figure 6). From Figure 5 it can be observed that the average of the fitted intensities
follows the simulated values, although the variability is rather large. However, in the extreme cases
where either the third or the fourth component are nearly non-existent (i.e., roughly for \( x < 0.15 \) and \( x > 0.85 \)), the accuracy in the result for both \( I_3 \) and \( I_4 \) is much improved. This allows one to conclude that
fitted o-Ps components with very small associated intensities (\(< 3\%\)) can be effectively neglected.
Another important conclusion is that the results obtained when fitting o-Ps lifetimes of relatively weak
components (less than 10 \% for the corresponding intensity) may not be very accurate for this type of
spectra. This is a consequence of the fact that, for small intensities, the net effect of the component in the $\chi^2$ is rather limited and therefore, the fit is insensitive to even large deviations in the corresponding lifetime.

Figure 6 can be used to understand the effect of fitting with less components than those actually present in the spectra. It can be observed that the fitted lifetime follows, as expected, the weighted average of $\tau_3$ and $\tau_4$ while the fitted intensity is close (within 2%) to the sum of $I_3$ and $I_4$. The largest deviations from the simulated values occur, obviously, for the cases in which the third and fourth components have roughly the same intensity ($x \approx 0.5$), since the spectrum is not dominated by one or the other and therefore more difficult to fit with just three components (the $\chi^2$ is also larger for $x \approx 0.5$).

Finally, it is worth mentioning that if the lifetime of either the third or the fourth component is approximately known, the corresponding parameter can be tightly constrained or even fixed in a four component fit, drastically reducing the variability in the results, especially for the intensity values.

**Conclusions**

A novel approach to the analysis of positron annihilation lifetime spectroscopy (PALS) data has been presented that combines both traditional well-established methods with new ones in order to provide an efficient, flexible and robust (user independent) analysis tool.

The advanced tools and algorithms developed have been implemented in PAScual, an easy-to-use software package that is freely available [14] and customisable (being free and open-source). The suitability of PAScual to solve a wide range of different general problems, using both real and artificially generated data has been demonstrated. When using the local optimisation routines, PAScual performs at least as well as existing codes such as LTv9 or POSITRONFIT, while offering also the possibility of performing bounded fits and of using the simulated annealing and MCMC-BI algorithms either independently or by combining them with the local optimisation.

Special emphasis has been placed on checking the performance of PAScual for the analysis of complex data such as that resulting from self-assembled amphiphile materials.
Acknowledgements

We thank Dr Nuno P Barradas for providing access to unpublished notes on the optimisation of the MCMC algorithm. This work was supported by the Australian Research Council (ARC) through DP0667189 and the award of a Federation Fellowship to CJD. SJP acknowledges an ARC APD award.

Supporting Information Available. Further technical information about the PAScual code is provided. The following additional features of PAScual are also discussed: flexible input/output formats, automatic parameter guessing, relative regions of interest, spectra simulation tool and, finally, the (Rectangular) Tao-Eldrup calculator for automated lifetime-to-free volume conversion. Also the so-called pseudo-continuous mode of operation is discussed (which allows results in the form of continuous intensity versus lifetime curves instead of discrete component analysis). This material is available at the on-line web page of this article.

References.


Figure captions

Figure 1. Screen capture of the main graphical user interface of *PAScual*. The left panels show the multiple spectra management and visualisation tools and the right panel shows the parameter input options.

Figure 2. Various projections of the solution space near the optimum for spectrum F0.5, as sampled by the MCMC-BI algorithm (using $10^6 \times n_F = 9 \times 10^6$ samples). The intensity scale represents the non-normalised posterior distribution in arbitrary units.

Figure 3. Evolution of the second and third components as a function of temperature for simulated data. The lines represent the values of the parameters used to generate the simulated spectra. The circles are the average of the results from local optimisation for 5 spectra at each temperature (the error bars represent their standard deviation). Each simulated spectrum had an area of $5 \times 10^5$ counts.

Figure 4. (Similar to Figure 4) Evolution of the second and third components as a function of temperature for simulated data. The lines represent the values of the parameters used to generate the simulated spectra. The circles are the results of fitting the generated spectra with MCMC-BI. Each point is the result of a simultaneous fit of 5 spectra of the same temperature using common parameters. The error bars were calculated by the MCMC-BI algorithm. Each simulated spectrum had an area of $5 \times 10^5$ counts.
**Figure 5.** Results for the third and fourth components (from fits using a four component model), as a function of their relative simulated weights ($x = I_3^{(\text{sim})}/I_4^{(\text{sim})}$). The circles represent the fit results and the solid lines indicate the "true" simulated values. The dashed lines in the right panels represent the bounding limits for the parameter in the fit.

**Figure 6.** Effect of a missing component in the fit model. The same simulated spectra of Figure 6 are analysed in this case using only three components. The circles represent the fit results. The solid lines indicate the total sum of the third and fourth simulated component intensities (left) and the weighted average of the simulated lifetimes for the same components (right).
## Tables:

### Table 1. Characteristic lifetimes for selected materials.

<table>
<thead>
<tr>
<th>Material</th>
<th>Lifetime [ps] $^a$</th>
<th>Literature [ps] $^{a,b}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al</td>
<td>163 (2)</td>
<td>165 (2)</td>
</tr>
<tr>
<td>PMMA</td>
<td>1891 (13)</td>
<td>1770-1870</td>
</tr>
<tr>
<td>PC</td>
<td>2097 (15)</td>
<td>2121 (16)</td>
</tr>
<tr>
<td>PSF</td>
<td>2002 (8)</td>
<td>2018</td>
</tr>
</tbody>
</table>

$^a$ Uncertainties are quoted in parenthesis when available.

$^b$ The literature values correspond to those in references [31-34].
Table 2. Comparison of local (LTv9 and PAScual using the local fitting mode) and global (PAScual using only Simulated Annealing) optimisation methods for the analysis of four series of simulated spectra.

<table>
<thead>
<tr>
<th>Series</th>
<th>Method</th>
<th>( \tau_1 ) [ps]</th>
<th>( \tau_2 ) [ps]</th>
<th>( \tau_3 ) [ps]</th>
<th>( \tau_4 ) [ps]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( I_1 ) [%]</td>
<td>( I_2 ) [%]</td>
<td>( I_3 ) [%]</td>
<td>( I_4 ) [%]</td>
</tr>
<tr>
<td>A</td>
<td>Simulation {10}</td>
<td>100</td>
<td>250</td>
<td>600</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>LTv9/ PAScual(_{\text{LOCAL}})</td>
<td>99 (5)</td>
<td>249 (18)</td>
<td>613 (43)</td>
<td>1010 (19)</td>
</tr>
<tr>
<td></td>
<td>PAScual(_{\text{SA}})</td>
<td>93 (2)</td>
<td>221 (4)</td>
<td>554 (15)</td>
<td>985 (6)</td>
</tr>
<tr>
<td>B</td>
<td>Simulation {5}</td>
<td>150</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LTv9/ PAScual(_{\text{LOCAL}})</td>
<td>149 (7)</td>
<td>249 (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAScual(_{\text{SA}})</td>
<td>147 (7)</td>
<td>247 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Simulation {5}</td>
<td>150</td>
<td>220</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LTv9/ PAScual(_{\text{LOCAL}})</td>
<td>147 (10)</td>
<td>219 (5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PAScual(_{\text{SA}})</td>
<td>141 (9)</td>
<td>214 (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Simulation {5}</td>
<td>150</td>
<td>190</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The results from a local optimisation with PAScual, are identical to those from LTv9 (within less than 0.5 ps in lifetimes and 0.05 % in intensities). Standard deviations among the series given in parenthesis.

The remaining simulation parameters are: FWHM=270 ps, g=58ps, B=100 counts, $A=2\times10^7$ counts for the A series and $2\times10^6$ counts for the B,C and D series.
Table 3. Comparison between fits using a different number of common parameters when fitting three related spectra.

<table>
<thead>
<tr>
<th>Common parameters in the fit&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Spectrum &lt;sup&gt;b&lt;/sup&gt;</th>
<th>$I_1$ [%]</th>
<th>$I_2$ [%]</th>
<th>$\tau_1$ [ps]</th>
<th>$\tau_2$ [ps]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Simulation)</td>
<td>B</td>
<td>50</td>
<td>50</td>
<td>150</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>50</td>
<td>50</td>
<td>150</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>50</td>
<td>50</td>
<td>150</td>
<td>190</td>
</tr>
<tr>
<td>None</td>
<td>B</td>
<td>43.6</td>
<td>56.4</td>
<td>141</td>
<td>245</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>35.4</td>
<td>64.6</td>
<td>132</td>
<td>212</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>15.5</td>
<td>84.5</td>
<td>111</td>
<td>179</td>
</tr>
<tr>
<td>FWHM, $\tau_1$</td>
<td>B</td>
<td>41.7</td>
<td>58.3</td>
<td>138</td>
<td>243</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>39.2</td>
<td>60.8</td>
<td>138</td>
<td>214</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>34.3</td>
<td>65.7</td>
<td>138</td>
<td>186</td>
</tr>
<tr>
<td>FWHM, $\tau_1, I_1, I_2$</td>
<td>B</td>
<td>46.2</td>
<td>53.8</td>
<td>145</td>
<td>247</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>46.2</td>
<td>53.8</td>
<td>145</td>
<td>219</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>46.2</td>
<td>53.8</td>
<td>145</td>
<td>191</td>
</tr>
</tbody>
</table>

<sup>a</sup> The background and the channel offset were free and not common in all cases.

<sup>b</sup> A single spectrum was chosen from each of the B, C and D series of Table 2 and fitted using local optimisation.
a) $L(a_1, a_2)$

b) $L(a_2, a_3)$

c) $L(a_1, \tau_1)$

d) $L(a_3, \tau_3)$