PDSFit

User Manual

Dinar Abdullin

Pablo Rauh Corro

Tobias Hett

Olav Schiemann

Bonn 2023

Table of contents

1 Introduction

PDSFit[1] is a software that allows the extraction of inter-spin distance distributions and, in some cases, angular distributions from primary time traces acquired by pulsed dipolar EPR spectroscopy (PDS).[2–5] This data analysis is performed *via* the model-based simulation or fitting of the time traces, using a parametrized geometric model of the spin system and a parametrized model of the PDS background. The features that determine the application field of PDSFit are listed below:

- PDSFit can be applied to time traces acquired with the pulsed electron-electron double resonance (PELDOR or DEER)[6,7] and relaxation-induced dipolar modulation enhancement (RIDME)[8,9] techniques.
- PDSFit can be applied to the time traces with or without orientation selectivity.
- PDSFit can be applied to spin systems consisting of up to two spin centers with (highly) anisotropic *g*-factors.
- PDSFit can be applied to spin systems with nonzero exchange coupling, if the exchange coupling constant falls into the weak exchange limit.

This manual is structured as follows. The underlying theory, the parametrized models, and main components of PDSFit are described Section 2. Sections 3 and 4 describe the installation and execution of the program, respectively. At execution, a configuration file is required which is described in Section 5. The output of the program is described in Section 6. Finally, the examples of the data analysis with PDSFit are given in Section 7.

2 Program description

2.1 Geometric models of a spin system

The geometric model used in PDSFit is depicted in Figure 1. The model consists of two coordinate frames associated with the *g*-tensor principle axes of the two spins, denoted as spin A and spin B. The reference coordinate system of the model is set to coincide with the *g*-tensor principle axes of spin A. The position and orientation of the *g*-tensor of spin B are described by three spherical coordinates (r, ξ, φ) and three Euler angles (α, β, γ), respectively. By default, the Euler angles are defined according to the ZXZ convention, but users can choose alternative conventions as well. To account for the conformational flexibility of the spin system, all six parameters are allowed to have either a uniform distribution or a Gaussian distribution. This allows each distribution $P(p)$ to be described using two parameters: a mean value $\langle p \rangle$ and a width Δp, where p represents r, ξ, φ, α, β, or γ. In case of a Gaussian distribution, Δp is defined as the full width at half maximum (FWHM). For the angular parameters, von Mises distributions are used, which have the same mean and FWHM as Gaussian distributions but account for the periodicity of angles. Overall, the model is described by 12 parameters, i.e., six mean values ($\langle r \rangle$, $\langle \xi \rangle$, $\langle \varphi \rangle$, $\langle \alpha \rangle$, $\langle \beta \rangle$, and $\langle \gamma \rangle$) and six widths (Δr, $\Delta \xi$, $\Delta \varphi$, $\Delta \alpha$, $\Delta \beta$, and $\Delta \gamma$). Since information about possible correlations between the geometric parameters is usually not available, it is assumed that they are uncorrelated.

Some spin systems, such as spin labels attached to biomolecules, can have several distinct conformational ensembles.[10] In this case, the geometric model can be extended by approximating the distributions of r, ξ, φ, α, β, and γ with multimodal uniform or Gaussian distributions, in which each mode represents one of the conformational ensembles. Since each *n*-modal distribution is described by *n* means, *n* widths, and $n - 1$ relative weights, the corresponding geometric model will have $6 \times (3n - 1)$ parameters. Although PDSFit sets no limit for the largest *n*, $n > 2$ would result in a quite large number of model parameters, which will pose a challenge for the fitting procedure described below.

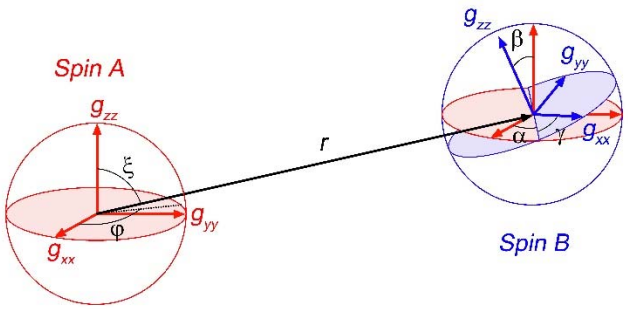


Figure 1. Geometric model of a spin system considered in PDSFit.

2.1.1 Symmetry considerations

Due to symmetry considerations, the parameters of the above-described geometric model cannot be uniquely defined. There are at least three different types of symmetry that are relevant for the analysis of PDS time traces. The first type is the symmetry of the magnetic tensors, i.e. isotropic, axial, or orthorhombic *g*- and hyperfine tensors. Depending on this symmetry, it may be necessary to consider all five angles (ξ, φ, α, β, γ) or only a subset of them. Table 1 provides a summary of the different spin center symmetries and the corresponding angles, along with the range each angle can adopt. If a particular angle is irrelevant for the PDS simulations, its value is set to zero. In PDSFit, users have the ability to choose which angles to include in the analysis and which angles to set to zero, thus allowing for customization of the PDS data analysis based on the specific spin system. The second type of symmetry is the inversion symmetry of the *g*-tensors. This symmetry implies that PDS time traces remain unchanged when one of the *g*-tensor axes of at least one of the spins is rotated by 180°.[5,11] As a result, there are 16 symmetry-related sets of angles that produce identical PDS time traces. PDSFit automatically calculates all these symmetry-related sets of angles and provides them as part of the output data. Additionally, the values of the geometric parameters depend on how the spins are assigned. If the two spins in the spin system are distinct, their assignment to spins A and B can be uniquely determined. However, if both spins correspond to the same species, there are two different ways to assign spins A and B. In such cases, PDSFit simulates the PDS time traces using one of the assignments but also calculates the parameters of the equivalent geometric model corresponding to the second assignment and provides them together with the symmetry-related parameters mentioned in the previous paragraph.

Table 1. Values and ranges of parameters ξ, φ, α, β, and γ used to simulate the PELDOR time traces of a two-spin system with isotropic, axial, and rhombic *g*-tensors of spins A and B.

Spin A	Spin B	ξ	φ	α	β	γ
isotropic	isotropic	0°	0°	0°	0°	0°
axial	isotropic	[0°, 90°]	0°	0°	0°	0°
axial	axial	[0°, 90°]	0°	[0°, 90°]	[0°, 90°]	0°
rhombic	isotropic	[0°, 90°]	[0°, 90°]	0°	0°	0°
rhombic	axial	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]	0°
rhombic	rhombic	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]	[0°, 90°]

2.2 Mathematical description of a PDS time trace

The simulation of the PDS time traces is based on the theory originally developed for PELDOR.[2,12] This theory considers a disordered sample containing a large number of molecules with a particular spin system. The PDS time trace of such a sample, $V(t)$, is given by the product of two contributions, a background $B(t)$ and a form factor $F(t)$:

$$V(t) = B(t) \cdot F(t). \quad (1)$$

In PELDOR, $B(t)$ describes the decay of the signal caused by the dipole-dipole interaction between electron spin centers located on different molecules. Therefore, $B(t)$ is often called the intermolecular component of the PELDOR time traces. The shape of $B(t)$ depends on the spatial distribution of electron spin centers in the sample. In the case of a homogeneous distribution, $B(t)$ is a monoexponential decay function.[13] Deviations from the homogeneous distribution lead to a $B(t)$ which can be described better by a stretched exponential function.[14] In RIDME, the shape of $B(t)$ is governed by spin and spectral diffusion, which makes it more complicated compared to the PELDOR background. Keller *et al.* derived a theoretical expression for the five-pulse RIDME background, which corresponds to the product of monoexponential and Gaussian decays.[15] However, experimental studies have emphasized that the background decay is often better described by one or two stretched exponential decays,[15] as well as polynomials.[16–19] The summary of all mentioned mathematical models of PDS background is given in Table 2.

The form factor $F(t)$ describes the modulation of the measured echo intensity by dipolar frequencies which correspond to the dipole-dipole interaction between the electron spin centers within individual molecules. Therefore, $F(t)$ is often called the intramolecular component of the PDS time traces. $F(t)$ is given by

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{4\pi}\int_0^{+\infty} P(r)dr \int_0^{\pi} P(\xi)\sin(\xi)d\xi \int_0^{2\pi} P(\varphi)d\varphi \int_0^{2\pi} P(\alpha)d\alpha \int_0^{\pi} P(\beta)\sin(\beta)d\beta$$
$$\times \int_0^{2\pi} P(\gamma)\,d\gamma \int_0^{\pi} \sin(\xi_{B_0})d\xi_{B_0} \int_0^{2\pi} d\varphi_{B_0}\,\lambda(1-\cos(\nu_{dd}t)). \quad (2)$$

Table 2. Background models and their parameters.

Background model	Function	Parameters
exponential decay	$B(t) = \exp(-kt)$	k
stretched exponential decay	$B(t) = \exp\left(-kt^{d/3}\right)$	k, d
2nd order polynomial	$B(t) = 1 + c_1 t + c_2 t^2$	c_1, c_2
3rd order polynomial	$B(t) = 1 + c_1 t + c_2 t^2 + c_3 t^3$	c_1, c_2, c_3
4th order polynomial	$B(t) = 1 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4$	c_1, c_2, c_3, c_4
Keller's exponential decay	$B(t) = \exp(-k_1 t - k_2 t^2)$	k_1, k_2

$F(0)$ is the PDS signal intensity at $t = 0$. ξ_{B_0} and φ_{B_0} are the polar and azimuthal angles, respectively, which describe the orientation of the magnetic field $\vec{B}_0$. $r, \xi, \varphi, \alpha, \beta,$ and γ are defined in Figure 1. The distributions of these parameters are normalized to 1:

$$\int_0^{+\infty} P(r)dr = 1, \quad \int_0^{\pi} P(\xi)\sin(\xi)d\xi = 1, \quad \int_0^{2\pi} P(\varphi)d\varphi = 1,$$

$$\int_0^{2\pi} P(\alpha)d\alpha = 1, \quad \int_0^{\pi} P(\beta)\sin(\beta)d\beta = 1, \quad \int_0^{2\pi} P(\gamma)\,d\gamma = 1.$$

ν_{dd} is a dipolar frequency. The equations used for ν_{dd} can be found in Section 2.2.1. λ is the probability with which each single spin pair contributes to the dipolar modulation of the PDS time trace. The equations for λ in PELDOR and RIDME are given in Section 2.2.2. In Equation 2, λ is averaged over all geometries of the spin pair and all orientations of the spin pair with respect to the magnetic field $\vec{B}_0$ (powder averaging), yielding the value of modulation depth parameter Λ. Note that Λ is also determined as a decay of $F(t)$ from $F(t = 0)$ to t where the dipolar oscillation is completely damped.

If the spin centers have a weak exchange coupling, and the corresponding exchange coupling constant J has a distribution $P(J)$, Equation 2 needs to be extended to

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{4\pi}\int_{-\infty}^{+\infty} P(J)dJ \int_0^{+\infty} P(r)dr \int_0^{\pi} P(\xi)\sin(\xi)d\xi \int_0^{2\pi} P(\varphi)d\varphi \int_0^{2\pi} P(\alpha)d\alpha$$
$$\times \int_0^{\pi} P(\beta)\sin(\beta)d\beta \int_0^{2\pi} P(\gamma)\,d\gamma \int_0^{\pi} \sin(\xi_{B_0})d\xi_{B_0} \int_0^{2\pi} d\varphi_{B_0}\,\lambda(1 - \cos(\{\nu_{dd} + J\}t)). \quad (3)$$

2.2.1 Dipolar frequency

Depending on the *g*-factors of spins A and B, different equations must be used for ν_{dd}. For spins with isotropic *g*-factors, it is given by

$$\nu_{dd} = \frac{\mu_0}{4\pi h}\frac{g_A g_B \beta_e^2}{r^3}[1 - 3\cos^2(\theta)], \quad (4)$$

where μ_0 is the vacuum permeability, β_e is the Bohr magneton, h is the Planck constant, r is the inter-spin distance, and θ is the angle between the inter-spin vector $\vec{r}$ and the applied static magnetic field $\vec{B}_0$. If one of the spins has an anisotropic *g*-tensor, e.g. spin A, Eq. (1) has to be replaced by the more general equation[17,18,20]:

$$\nu_{dd} = \frac{\mu_0}{4\pi h}\frac{g_{A,eff} g_B \beta_e^2}{r^3}\left[1 - 3\left(\frac{\hat{g}_A \hat{g}_A^T}{g_{A,eff}^2}\frac{\vec{B}_0}{B_0} \cdot \frac{\vec{r}}{r}\right)\cos(\theta)\right], \quad (5)$$

where $\hat{g}_{\mathrm{A}}$ and $g_{\mathrm{A},eff}$ are the g-tensor and the effective g-values of the anisotropic spin A, respectively, index T denotes matrix transposition, and B_0 is the amplitude of $\vec{B}_0$. Finally, if both spins have an anisotropic g-tensor, ν_{dd} is given by

$$\nu_{dd} = \frac{\mu_0}{4\pi h}\frac{g_{\mathrm{A},eff}g_{\mathrm{B},eff}\beta_e^2}{r^3}\left[\left(\frac{\hat{g}_{\mathrm{A}}\hat{g}_{\mathrm{A}}^T}{g_{\mathrm{A},eff}^2}\frac{\vec{B}_0}{B_0}\cdot\frac{\hat{g}_{\mathrm{B}}\hat{g}_{\mathrm{B}}^T}{g_{\mathrm{B},eff}^2}\frac{\vec{B}_0}{B_0}\right) - 3\left(\frac{\hat{g}_{\mathrm{A}}\hat{g}_{\mathrm{A}}^T}{g_{\mathrm{A},eff}^2}\frac{\vec{B}_0}{B_0}\cdot\frac{\vec{r}}{r}\right)\left(\frac{\hat{g}_{\mathrm{B}}\hat{g}_{\mathrm{B}}^T}{g_{\mathrm{B},eff}^2}\frac{\vec{B}_0}{B_0}\cdot\frac{\vec{r}}{r}\right)\right] \quad (6)$$

where $\hat{g}_{\mathrm{B}}$ and $g_{\mathrm{B},eff}$ are the g-tensor and the effective g-values of the anisotropic spin B, respectively. The latter two equations consider that the quantization axes of anisotropic spin centers are not necessarily collinear with $\vec{B}_0$. Consequently, ν_{dd} becomes dependent on the relative orientation of one or both g-tensors with respect to each other and to the orientation of the inter-spin vector $\vec{r}$. Thus, when spin A is anisotropic, ν_{dd} depends not only on r and θ, but also on angles ξ and φ. When spin B is anisotropic too, ν_{dd} additionally depends on α, β, and γ.

2.2.2 Modulation depth parameter

For a single orientation of the two-spin system in the applied magnetic field $\vec{B}_0$, λ is given by

$$\lambda = \frac{p_{det}(\omega_A)\cdot p_{pump}(\omega_B) + p_{det}(\omega_B)\cdot p_{pump}(\omega_A)}{F(0)}. \quad (7)$$

ω_A and ω_B are the resonance angular frequencies of spins A and B, respectively. $p_{det}(\omega_A)$ and $p_{det}(\omega_B)$ are the probabilities that the spins A and B with resonance frequencies ω_A and ω_B, respectively, are excited by the detection pulses. $p_{pump}(\omega_A)$ and $p_{pump}(\omega_B)$ are the probabilities that the spins A and B are flipped by the pump pulse (PELDOR) or the spontaneous relaxation (RIDME). $F(0)$ is the amplitude of the PDS time trace at $t = 0$ and is given by

$$F(0) = \frac{1}{4\pi}\int_0^{\pi}\sin(\xi_{B_0})d\xi_{B_0}\int_0^{2\pi}d\varphi_{B_0}\ \{p_{det}(\omega_A) + p_{det}(\omega_B)\}, \quad (8)$$

where ξ_{B_0} and φ_{B_0} are the polar and azimuthal angles, respectively, which describe the orientation of the magnetic field $\vec{B}_0$ in the reference coordinate system. If several different resonance angular frequencies ω_A and ω_B correspond to the same orientation of the magnetic field $\vec{B}_0$, e.g., due to the hyperfine interaction of spin A and/or B with nuclear spins, Equations 7 and 8 are replaced by

$$\lambda = \frac{\sum_{\omega_A}\sum_{\omega_B} w_{\omega_A}w_{\omega_B}\left\{p_{probe}(\omega_A)\cdot p_{pump}(\omega_B) + p_{probe}(\omega_B)\cdot p_{pump}(\omega_A)\right\}}{F(0)} \quad (9)$$

and

$$F(0) = \frac{1}{4\pi}\int_0^{\pi}\sin(\xi_{B_0})d\xi_{B_0}\int_0^{2\pi}d\varphi_{B_0}\sum_{\omega_A}\sum_{\omega_B} w_{\omega_A}w_{\omega_B}\{p_{det}(\omega_A) + p_{det}(\omega_B)\}, \quad (10)$$

where w_{ω_A} and w_{ω_B} are the relative weights of different ω_A and ω_B values, respectively. The sum of all w_{ω_A}, as well as the sum of all w_{ω_B}, equals 1.

The equations for $p_{det}(\omega)$ and $p_{pump}(\omega)$ depend on the PDS pulse sequence. In the case of four-pulse PELDOR with rectangular pulses, they can be calculated as follows[21]

$$p_{det}(\omega) = \frac{\omega_{1,\pi/2}}{\Omega_{\pi/2}} \sin\left(\Omega_{\pi/2} t_{\pi/2}\right) \frac{\omega_{1,\pi}^4}{4\Omega_\pi^4} \{1 - \cos(\Omega_\pi t_\pi)\}^2, \qquad (11a)$$

$$p_{pump}(\omega) = \frac{\omega_{1,pump}^2}{2\Omega_{pump}^2} \left\{1 - \cos\left(\Omega_{pump} t_{pump}\right)\right\}, \qquad (11b)$$

where

$$\omega_{1,\pi/2} = \frac{\pi/2}{t_{\pi/2}}, \qquad \Omega_{\pi/2}^2 = \omega_{1,\pi/2}^2 + (\omega - \omega_{det})^2,$$

$$\omega_{1,\pi} = \frac{\pi}{t_\pi}, \qquad \Omega_\pi^2 = \omega_{1,\pi}^2 + (\omega - \omega_{det})^2, \qquad (11c)$$

$$\omega_{1,pump} = \frac{\pi}{t_{pump}}, \qquad \Omega_{pump}^2 = \omega_{1,pump}^2 + (\omega - \omega_{pump})^2.$$

$t_{\pi/2}$ and t_π are the lengths of the probe $\pi/2$- and π-pulses, respectively, and t_{pump} is the length of the pump pulse, ω_{det} and ω_{pump} are the angular frequencies of the detection and pump pulses, respectively.

In the case of five-pulse RIDME with rectangular pulses, $p_{det}(\omega)$ and $p_{pump}(\omega)$ can be calculated using the following equations[8,22]:

$$p_{det}(\omega) = \frac{\omega_{1,\pi/2}^3}{\Omega_{\pi/2}^3} \sin^3\left(\Omega_{\pi/2} t_{\pi/2}\right) \frac{\omega_{1,\pi}^4}{4\Omega_\pi^4} \{1 - \cos(\Omega_\pi t_\pi)\}^2, \qquad (12a)$$

$$p_{pump} = \frac{1}{2}\left\{1 - \exp\left(-\frac{T_{mix}}{T_1}\right)\right\}, \qquad (12b)$$

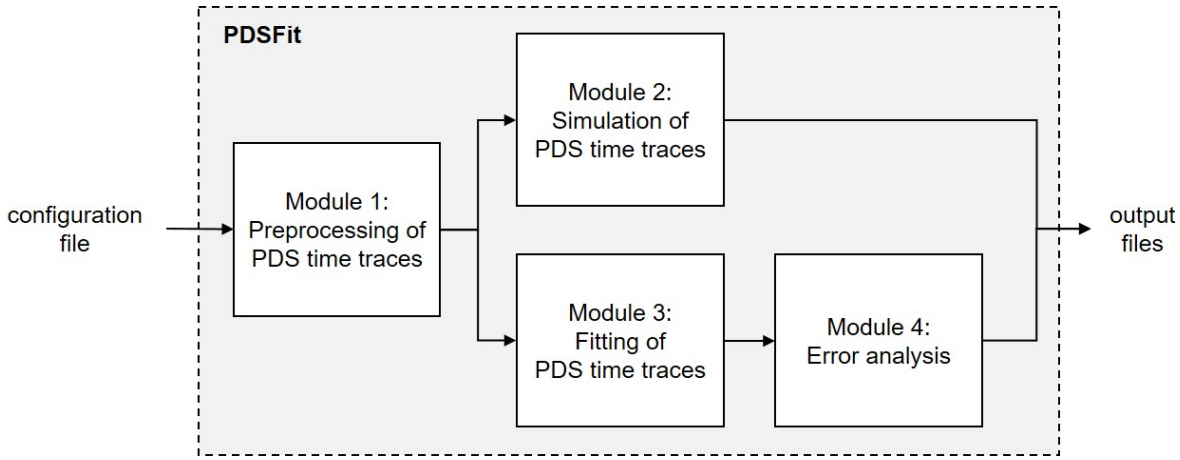
where $t_{\pi/2}$, t_π $\omega_{1,\pi/2}$, $\omega_{1,\pi}$, $\Omega_{\pi/2}$, and Ω_π are defined as above, T_{mix} is the length of the mixing inter-pulse interval in the RIDME pulse sequence, and T_1 is the longitudinal (spin-lattice) relaxation time of the flipped spins. Note that in RIDME p_{pump} does not depend on the resonance frequency ω of the flipped spin. However, at temperatures comparable to the electron Zeeman energy (usually liquid helium temperatures), p_{pump} depends on the effective g-factor of the flipped spin, g_{eff}, as given by[17]

$$p_{pump} = \frac{2f}{(1+f)^2}\left\{1 - \exp\left(-\frac{T_{mix}}{T_1}\right)\right\}, \text{ where } f = \exp\left(-\frac{g_{eff}\beta B_0}{k_B T}\right), \qquad (13)$$

where β is the Bohr magneton, k_B is the Boltzmann constant, and T is the temperature of the RIDME experiment. In the high-temperature limit, Equation 13 transforms into Equation 12b.

2.3 The block diagram of PDSFit

The block diagram of PDSFit is shown in Scheme 1. PDSFit is a console application that runs via Terminal (Linux, macOS) or Command Prompt (Windows). When starting the program, a configuration file (Section 5) must be provided that specifies all input data of the program. When executed, PDSFit reads out the configuration file, including the specified PDS time traces. In the first module of the program, the loaded time traces undergo automatic preprocessing, which includes phase correction, zero-point correction, normalization, and determination of the noise level (module 1). The preprocessed time traces are then either simulated (module 2) or fitted (module 3) using the geometric model of the spin system and a PDS background model. The fitting is followed by an error analysis (module 4), which evaluates the errors of the optimized model parameters. At the end, all results of the PDSFit analysis are output in the form of graphic and data files into a user-defined directory (Section 6). In the following each module of PDSFit is described in more detail.



Scheme 1. The block diagram of PDSFit.

2.4 Module 1: Preprocessing of PDS time traces

The preprocessing of the PDS time traces includes the following steps:

1) The phase of each PDS time traces is adjusted to ensure that the entire signal is contained in the in-phase (real) component, while the quadrature (imaginary) component is a horizontal line.[23] In practice, slight nonlinearity is sometimes observed for the quadrature component, especially at the beginning of the time trace. If the PDS time trace contains ESEEM artifacts, this method may fail, and the zero point must be set manually.
2) The time axis of each PDS time trace is shifted to set the correct zero point. The zero point is determined using a method based on the calculation of the first moment.[24] If the PDS time trace is very noisy, this method may fail, and the zero point must be set manually.

3) The in-phase component of each PDS time trace is normalized to 1.
4) The standard deviation of the noise, σ_N, in each PDS time trace is estimated using its quadrature component.[25] If the quadrature component of the PDS time trace is not available, σ_N must be set manually in the configuration file.

2.5 Module 2: Simulation of PDS time traces

The simulation of PDS time traces is based on Equation 1 and involves the simulation of the form factor $F(t)$ and the background $B(t)$ based on the user-defined geometric model of the spin system and the user-defined background model, respectively. $F(t)$ is calculated based on Equation 2, in which the distributions $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$ are defined by the user. In addition, the distribution of the exchange coupling constant, $P(J)$, can be included in the simulation via Equation 3. If included, $P(J)$ is also approximated by either a uniform distribution or a Gaussian distribution, resulting in two additional simulation parameters: a mean value $\langle J \rangle$ and a width ΔJ. The integrals in Equation 3 are solved numerically via the Monte-Carlo method, which consists of the following steps:

- N_{MC} Monte-Carlo samples are generated by randomly picking values of r, ξ, φ, α, β, γ, J, and ξ_{B_0} from the sampling distributions $P(r)$, $P(\xi)\sin(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)\sin(\beta)$, $P(\gamma)$, $P(J)$, $\sin(\xi_{B_0})$, respectively, while the values for φ_{B_0} are generated by picking values from a uniform distribution on the interval [0, 2π]. Using sampling distributions instead of uniform random samples prevents the consideration of parameter values with probabilities close to zero or zero, thereby increasing the efficiency of Monte-Carlo integration. In this work, N_{MC} was set to 10^6.
- The values of ν_{dd} and λ are calculated for each Monte-Carlo sample. Besides the integration variables, these calculations use the EPR parameters of the spin centers and the experimental parameters of the PDS pulse sequence, which allow simulating the EPR spectra of the spin centers and the excitation profiles of the microwave pulses, respectively (Sections 2.2.1 and 2.2.2).
- $F(t)$ is calculated as the sum over N_{MC} Monte-Carlo samples:

$$\frac{F(t)}{F(0)} = 1 - \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \lambda_i \left\{ 1 - cos\left(\left(\nu_{dd,i} + J_i \right) t \right) \right\}. \qquad (14)$$

$B(t)$ is calculated using one of the functions listed in Table 2. The corresponding background parameters (last column in Table 2) can either be defined by the user or be optimized using the Nelder-Mead algorithm[26] to minimize the mean squared difference (MSD) between the experimental PDS time trace, $V_{exp}(t)$, and the corresponding simulated PDS time trace, $V_{sim}(t)$:

$$MSD(\vec{b}) = \sum_{j=1}^{N_t} \{V_{exp}(t_j) - V_{sim}(t_j)\}^2 = \sum_{j=1}^{N_t} \{V_{exp}(t_j) - F(t_j)B(t_j, \vec{b})\}^2 \quad (15)$$

where $\vec{b}$ is the vector consisting of the background parameters, and the sum over *j* denotes the sum over the N_t time points in the PDS time trace.

Often, the calculated value of the modulation depth parameter Λ exceeds the experimental value due to, e.g., the imperfection of the pump pulse in the PELDOR experiment or non-ideal sample conditions such as incomplete spin labeling or incomplete metal loading. To reproduce this effect in the simulations, PDSFit uses an additional parameter η to scale the modulation depth parameter. For each simulated PDS time trace, η can be either set to a constant or can be optimized alongside with the background parameters. In the latter case, the MSD is calculated as

$$MSD(\vec{b}, \eta) = \sum_{j=1}^{N_t} \{V_{exp}(t_j) - F(t_j, \eta)B(t_j, \vec{b})\}^2, \text{ where } \frac{F(t_j, \eta)}{F(0, \eta)} = 1 - \eta\left(1 - \frac{F(t_j)}{F(0)}\right) \quad (16)$$

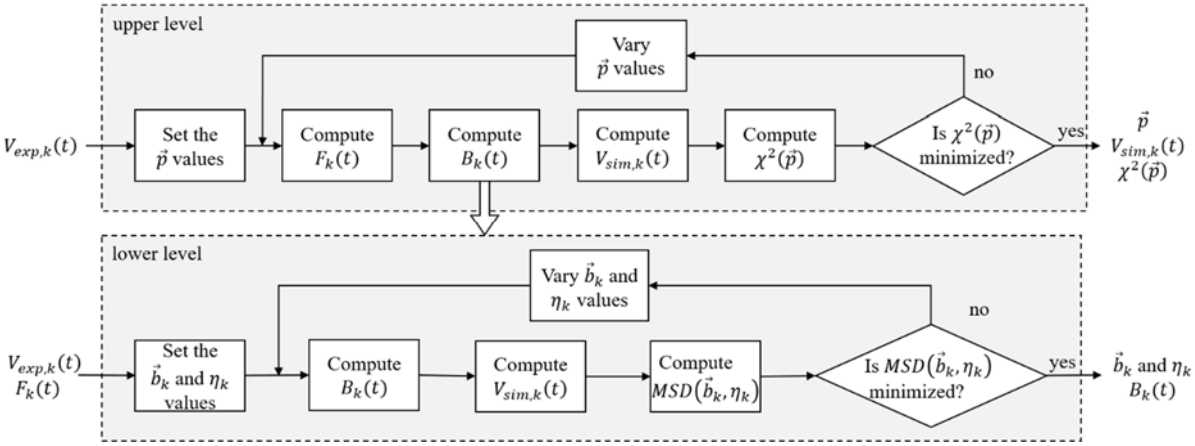
2.6 Module 3: Fitting of PDS time traces

When fitting PDS time traces, the parameters of the user-defined geometric model of a spin system and the user-defined PDS background model are used as fitting parameters and are optimized to achieve the best match between the simulated and experimental PDS time traces. The optimization of the fitting parameters has two levels called the upper level and lower level (Scheme 2). At the upper level, the parameters of the geometric model and, optionally, $P(J)$ are optimized using a genetic algorithm.[27] This algorithm is iterative and typically requires a significant number of optimization steps (500 - 1000). At each step, the genetic algorithm considers multiple sets of fitting parameter values and identifies the set that yields the smallest χ^2 between the simulated and experimental PDS time traces:

$$\chi^2(\vec{p}) = \sum_{k=1}^{N_V} \sum_{j=1}^{N_t} \frac{\left(V_{exp,k}(t_j) - V_{sim,k}(t_j)\right)^2}{\sigma_{N,k}^2}, \quad (17)$$

where $\vec{p}$ is the vector consisting of the parameters of the geometric model. The sums over *k* and *j* denote the sum over different PDS time traces and the sum over the time points in the k^{th} PDS time trace, respectively. $V_{exp,k}(t)$ and $V_{sim,k}(t)$ represent the k^{th} experimental PDS time trace and the corresponding simulated PDS time trace, respectively. $\sigma_{N,k}$ is the noise level (standard deviation) of the k^{th} experimental PDS time trace. The calculation of $V_{sim,k}(t)$ follows the same procedure as described in the previous section, including the optimization of the background parameters and η for each PDS time trace (see Equation 16). This determines the lower level of the optimization, in

which the background parameters are optimized for every set of geometric model parameters evaluated by the genetic algorithm. The reason for optimizing the background parameters separately from the geometric model parameters is that a gradually decaying background can be effectively fitted using a local optimization algorithm such as the Nelder-Mead algorithm and, therefore, there is no need to use a computationally expensive global optimization algorithm, like the genetic algorithm, for optimizing the background parameters.



Scheme 2. Block diagram of the fitting module of PDSFit. $V_{exp,k}(t)$ and $V_{sim,k}(t)$ are the k^{th} experimental PDS time trace and the corresponding simulated PDS time trace, respectively. $F_k(t)$ and $B_k(t)$ are the two components of $V_{sim,k}(t)$ and denote the form factor and background, respectively. $\vec{p}$ and $\vec{b}_k$ are the vectors consisting of the parameters of the geometric model and the background parameters for the k^{th} time trace, respectively. η_k is the scaling factor of the modulation depth of the k^{th} time trace. MSD and χ^2 are defined by Equations 16 and 17, respectively.

The genetic algorithm is well suited for optimizing the geometric model parameters because it is a derivative-free method, it does not require initial guesses for the fitting parameters, and it is efficient in terms of computation time when dealing with a large number of fitting parameters. However, like any other algorithm, it has several drawbacks. Firstly, the genetic algorithm has several intrinsic parameters that influence its performance. The values of these parameters which have been found to yield good results in previous studies are summarized in Section 2.6.1. Secondly, the genetic algorithm is a stochastic algorithm, utilizing randomness to search for the global χ^2-minimum. As a result, the total number of optimization steps required to reach the global χ^2-minimum can vary between different optimization runs, especially when encountering deep local χ^2-minima. In such cases, reaching a local χ^2-minimum may necessitate a substantial number of additional steps or, in some cases, the algorithm may fail to find the global χ^2-minimum. To address this issue, PDSFit performs the entire optimization process multiple times (Figure 2a). In the performed tests, 10 repetitions were sufficient to find the global χ^2-minimum,

even in the presence of deep local χ^2-minima. Lastly, it has been observed that the genetic algorithm often finds a solution very close to the global χ^2-minimum within a few hundred optimization steps. However, due to the inherent stochastic nature of the algorithm, achieving precise convergence to the global χ^2-minimum may require a larger number of steps. To accelerate this final step, the Nelder-Mead algorithm can be applied immediately after the genetic algorithm (Figure 2b). In the conducted tests, 500 steps of the Nelder-Mead optimization were sufficient to reach the "bottom" of the χ^2-minimum.

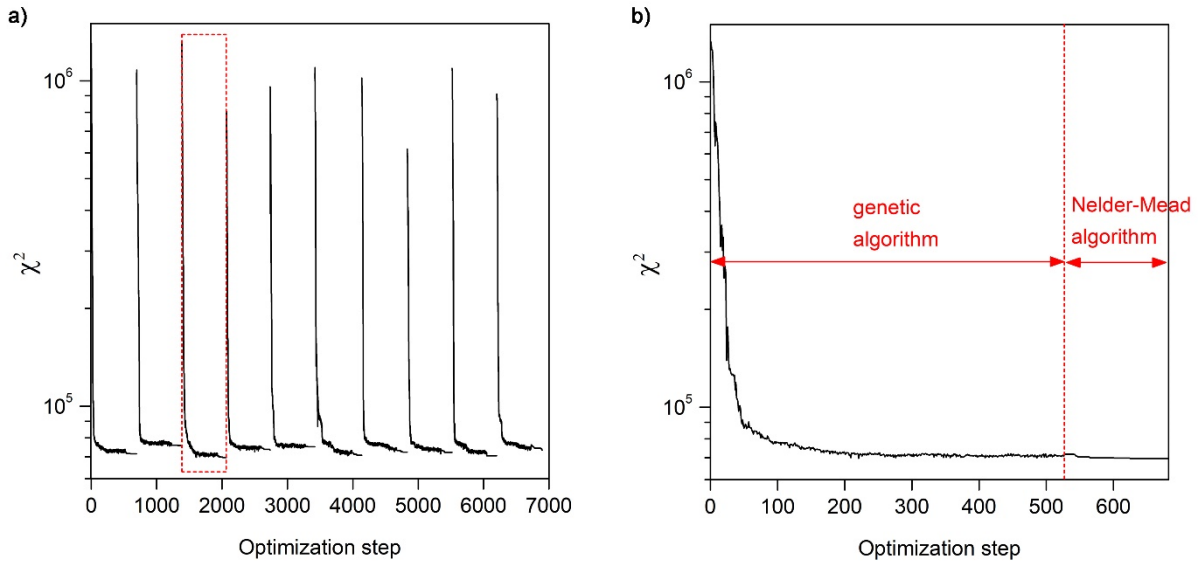


Figure 2. An example of χ^2 recorded as a function of optimization step. **a)** χ^2 is shown for 10 consecutive optimizations. The minimal χ^2-value was obtained in optimization run no. 3, which is indicated by a red rectangle. **b)** The magnified picture of the optimization run no. 3, which included 527 iterations of the genetic algorithm, followed by 155 iterations of the Nelder-Mead algorithm.

2.6.1 Genetic algorithm

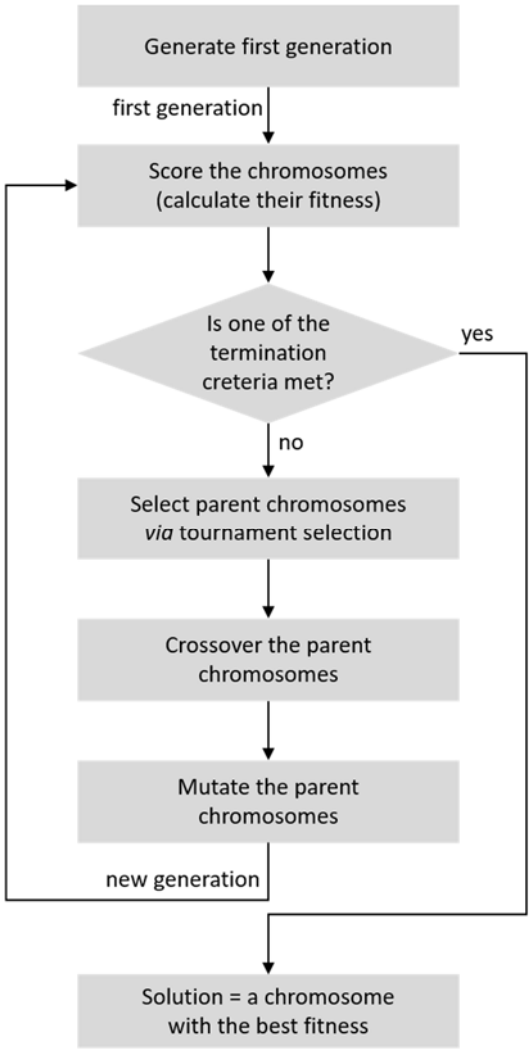
The genetic algorithm (GA) is an optimization algorithm that mimics the processes of Darwin's natural evolution. A description and various implementations of the GA can be found elsewhere.[28,29] Here, we present a concise description of the GA implemented in PDSFit.

In the GA, a specific terminology is used. Each fitting parameter is called a gene. A set of genes that corresponds to a complete set of fitting parameters is referred to as a chromosome. A set of chromosomes is called a generation. Each chromosome in the generation corresponds to a potential solution to the optimization problem. The accuracy of each solution is assessed using the χ^2-value (Equation 17), which determines the fitness of the corresponding chromosome. Consequently, the chromosomes with the best fitness are the chromosomes with the lowest χ^2-values. The procedure that describes the calculation of fitness is called scoring.

The implementation of the GA in PDSFit is outlined in Scheme 3. The evolutionary cycle of the GA begins by creating N_c chromosomes, which form the first generation. The genes of the initial chromosomes are randomly generated using the ranges specified by the used for the corresponding fitting parameters. Subsequently, all chromosomes of the first generation undergo scoring, which yields the fitness for each chromosome. Adhering to the principles of evolution, the fitness of chromosomes in the current generation influences their likelihood of producing offspring and contributing to the next generation. Therefore, in order to generate an offspring, the following three procedures are repeated N_c / 2 times:

1) Two parent chromosomes are determined via tournament selection. In this process, two pairs of chromosomes are randomly selected from the current generation, and from each pair, the chromosome with the highest fitness is chosen. This ensures that chromosomes with better fitness have a higher chance of being selected as parents for the next generation.
2) The genes of the parent chromosomes undergo crossover with a probability p_c. The crossover is responsible for the gene exchange between chromosomes and is therefore crucial for the convergence of the GA to the global χ^2-minimum. The crossover is done either around one random gene in the gene sequence of the parent chromosomes, which is known as a single-point crossover, or for M randomly chosen genes of the parent chromosomes, which is known as a uniform crossover. M is defined as a product of the total number of genes in a single chromosome, N_g, and the probability of gene exchange, p_{ge}. The ratio between the single-point crossover and the uniform crossover is denoted as r_c.
3) The genes of the parent chromosomes undergo mutation with a probability p_m. The mutation introduces diversity into the generation, preventing the convergence of the GA into local χ^2-minima. The mutation is also performed in two different ways: either as a uniform mutation or as a creep mutation. In the case of the uniform mutation, a mutated gene is set to a random value within the entire range specified for this gene. In the case of the creep mutation, a mutated gene is set to a random value within the range which is centered about the previous gene value and which has a width of κ % of entire range specified for this gene. The ratio between the creep mutation and the uniform mutation is denoted as r_m.

Once the new generation with N_c offspring chromosomes was created, the next evolutionary cycle begins, starting with the scoring and mating described above. The whole process is repeated a number of times until one of the termination criteria is met. The GA terminates either when the best fitness (the lowest χ^2-value) does not change by more than δ % in $N_{G,const}$ consecutive generations or when the maximum number of generations, $N_{G,max}$, is reached.



Scheme 3. Block diagram of the genetic algorithm implemented in PDSFit.

From the above follows that the GA has 10 intrinsic parameters, namely N_c, p_c, r_c, p_{ge}, p_m, r_m, κ, δ, $N_{G,const}$, and $N_{G,max}$. These parameters influence the performance of the GA. Unfortunately, there are no simple rules that allow determining the optimal values of these parameters and, therefore, these values have to be found empirically. Based on the previous tests,[27,30–33,17,34,35] a reasonable performance of the GA can be achieved using the following values of the intrinsic parameters: $N_c \geq 10 \times N_g$, where N_g is the number of genes in a single chromosome and, simultaneously, the number of fitting parameters optimized by the GA, $p_c = 0.5$, $r_c = 1$, $p_{ge} = 0.5$, $p_m = 0.05$, $r_m = 1$, $\kappa = 10$ %, $\delta = 5$ %, $N_{G,const} = 500$, and $N_{G,max} = 1000$.

2.7 Module 4: Error analysis

The fitting procedure of PDSFit yields optimized values for the parameters that describe the geometric model of a spin system and the PDS background model. However, the uncertainties of the obtained parameter values are unknown and require an error analysis. Since the parameter

space is usually large and the fitting of PDS time traces is time consuming, common approaches for error analysis such as bootstrap[36] or Bayesian[37] analyses are not feasible within a reasonable timeframe. Therefore, PDSFit performs the error analysis using a specialized approach, which assumes that the contributions to χ^2 from optimized $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, and $P(\gamma)$, and even from individual modes in these distributions, are uncorrelated near the global minimum of χ^2. This allows recording χ^2 values for different subspaces of the geometric model parameters rather than examining the entire parameter space. In the case of the unimodal distributions, each subspace consists of two parameters, a mean value and a corresponding width. In the case of the multimodal distributions, each subspace comprises three parameters: a mean value, a width, and a relative weight describing a single mode in the corresponding distribution. The parameters of the geometric model that do not belong to the selected subspace are set to their optimized values, while the background parameters and η are optimized in accordance with Equation 16. As a result, two- or three-dimensional χ^2-surfaces, referred to as error surfaces, are obtained (Figure 3). In the next step in the error analysis, these error surfaces are used to determine the confidence intervals of the fitting parameters, which are determined as ranges in which the χ^2 values do not exceed a threshold, $\Delta\chi^2$, applied above the χ^2 minimum, χ^2_{min} (Figures 3). $\Delta\chi^2$ is composed of two contributions. The first contribution takes into account errors related to the noise in the experimental data and possible discrepancies between the actual spin system and its geometric model. It is calculated based on a chi-squared distribution using a *p*-value of 95% (2σ confidence level).[36] The second contribution takes into account the numerical error, which is mostly determined by the accuracy of the Monte-Carlo integration. It is calculated as 2σ of a Gaussian distribution, obtained when calculating χ^2 for 10^4 identical sets of the optimized fitting parameters. In the last step of the error analysis, the difference between the optimized value of each fitting parameter and the lower and upper bounds of the corresponding confidence interval is calculated, yielding the parameters' errors. Our test showed that the uncertainty intervals are often asymmetric with respect to the optimized values of fitting parameters (Figures 3). Therefore, the parameter errors are calculated as asymmetric errors.

Several remarks regarding the error analysis need to be made. First, it should be noted that the error surfaces calculated for geometric model parameters can be also used to determine the errors of the PDS background parameters and η. Given that the background parameters and η are optimized for each point on the error surfaces, PDSFit identifies all corresponding values that yield χ^2-values below the threshold. These values are then converted into 2σ confidence intervals and asymmetric errors in the same way as it is done for the geometric model parameters. Second, the errors of $\langle\xi\rangle$, $\langle\varphi\rangle$, $\langle\alpha\rangle$, $\langle\beta\rangle$, and $\langle\gamma\rangle$ are determined within the range [0°, 90°] in PDSFit, but extend

beyond the specified ranges according to the periodicity and symmetry (Section 2.1.1). Finally, due to the assumptions used to perform the error analysis, parameter errors obtained by PDSFit should be considered as lower bounds of the actual parameter errors.

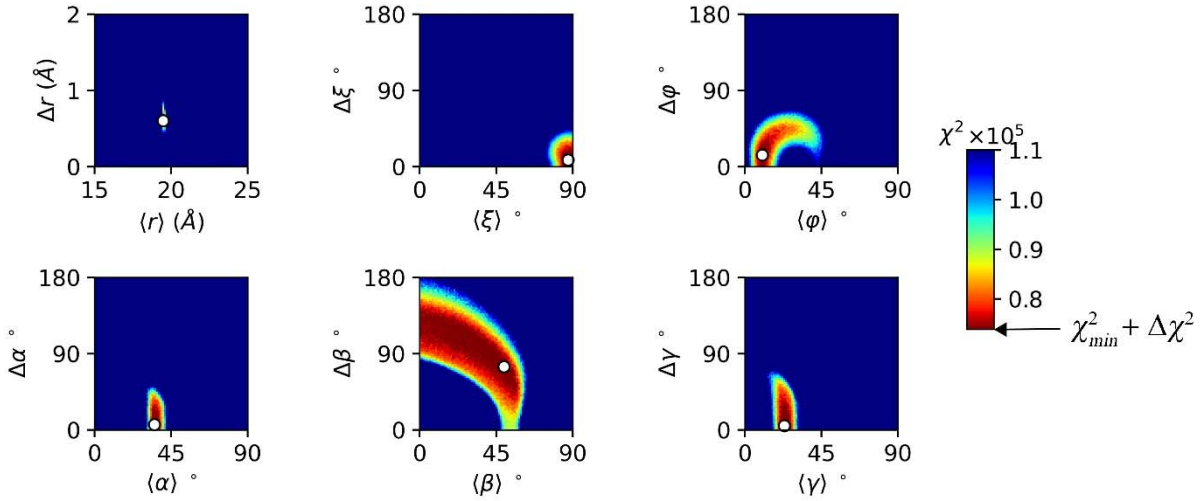


Figure 3. An example of error surfaces obtained for the fitting parameters. Dark red regions correspond to the 2σ confidence intervals, which lie below the threshold $\chi^2_{min} + \Delta\chi^2$. The corresponding optimized values of the fitting parameters are depicted as white circles.

2.8 Technical details

PDSFit is written in Python 3.8 using linbraries numpy, scipy, matplotlib, libconf, mpi4py, and pyinstaller. It is free of charge and can be distributed under GNU General Public License v3.0.

The program supports parallel computing and, therefore, its performance can be significantly improved when using a large number of CPUs. When fitting PDS time traces with PDSFit, the optimal number of CPUs is equal to the number of chromsomes considered simulteneously by the genetic algorithm (Section 2.6.1). For the examples given in Section 7, the program was run on an HPC-cluster with 64-192 CPUs and 5 GB RAM per CPU.

3 Installation

3.1 Using the executable

The precompiled PDSFit executables are available for Windows 10 and Linux CentOS 7 operating systems. These executables can be dowloaded from

https://github.com/dinarabdullin/PDSFit/releases.

When downloaded, one needs to unzip the executable into the user directory (referred to as user_directory), after which it will be ready for use. If PDSFit will be run on a cluster, we recommend to install the prorgam in accordance with Section 3.2.

3.2 Using the source code

1) Download the PDSFit source code from

   https://github.com/dinarabdullin/PDSFit

   and unzip it into the user directory (referred to as user_directory). If needed, rename the folder with the PDSFit source code to "PDSFit".

2) Make sure that Python 3.8 is installed and set as a default Python version. To check the default Python version, open Terminal (Linux, macOS) or Command Prompt (Windows) and type:

   ```
   python3 --version (Linux, macOS)
   python --version (Windows)
   ```

3) Navigate to the directory with the source code:

   ```
   cd user_directory/PDSFit
   ```

4) Install the python libraries required for PDSFit. The best practice to do this is to use a virtual environment. To create a virtual environment, type

   ```
   python3 -m venv pdsfit_env (Linux, macOS)
   python -m venv pdsfit_env (Windows)
   ```

   Here, pdsfit_env is the name of the virtual environment. Next, activate the virtual environment

   ```
   source pdsfit_env/bin/activate
   ```

   and install the required python libraries to the virtual environment:

   ```
   pip install -r requirements.txt
   ```

5) (optional) To enable running PDSFit on a computer cluster, such as an HPC-cluster, make sure that OpenMPI is installed.

6) (optional) To convert the source code into an executable, the python library pyinstaller is used:

   ```
   pyinstaller --onefile PDSFit.py
   ```

This command generates the PDSFit.spec file in the current directory. Open this file with a text editor and replace empty brackets in exe = EXE(…) with [('W ignore', None, 'OPTION')]. Then, save and close the PDSFit.spec file, and run the following command:

```
pyinstaller --onefile PDSFit.spec
```

Copy the PDSFit executable, PDSFit (Linux, macOS) or PDSFit.exe (Windows), from user_directory/PDSFit/dist to user_directory/PDSFit.

4 Running the program

4.1 Using the executable

4.2.1 On a PC

To run PDSFit on a PC, perform the following steps:

1) Make sure that PDSFit was installed in accordance with Section 3.1 or Section 3.2 (steps 1-4 and 6).
2) Prepare the configuration file (referred to as configuration_file_name.cfg). The intructions can be found in Chapter 5.
3) Open Terminal (Linux, macOS) or Command Prompt (Windows) and navigate to the directory in which the configuration file is located (referred to as cofiguration_file_directory):

```
cd configuration_file_directory
```

4) Run the program:

```
user_directory/PDSFit/PDSFit configuration_file_name.cfg (Linux, macOS)
user_directory/PDSFit/PDSFit.exe configuration_file_name.cfg (Windows)
```

4.2 Using the source code

4.1.1 On a PC

To run PDSFit on a PC, perform the following steps:

1) Make sure that PDSFit was installed in accordance with Section 3.1 (steps 1-4).
2) Make sure that the virtual environment pdsfit_env is activated. For this, open Terminal (Linux, macOS) or Command Prompt (Windows) and type:

```
cd user_directory/PDSFit
python3 -m venv pdsfit_env (Linux, macOS)
python -m venv pdsfit_env (Windows)
```

3) Prepare the configuration file (referred to as configuration_file_name.cfg). The intructions can be found in Chapter 5.
4) Navigate to the directory in which the configuration file is located (referred to as cofiguration_file_directory):

```
cd configuration_file_directory
```

5) Run the program:

```
python3 user_directory/PDSFit/PDSFit.py configuration_file_name.cfg (Linux, macOS)
python user_directory/PDSFit/PDSFit.py configuration_file_name.cfg (Windows)
```

4.1.2 On a cluster

Depending on a cluster and its management system, there are many possibilities to run PDSFit. Below, the intructions are provided to run PDSFit on the Linux cluster with the SLURM management system:

1) Make sure that PDSFit was installed in accordance with Section 3.1 (steps 1-5).
2) Prepare the configuration file (referred to as configuration_file_name.cfg). The intructions can be found in Chapter 5.
3) Prepare the SLURM file (refered to as slurm_file_name.slurm). An example of the SLURM file is given below:

```
#!/bin/bash
#SBATCH --nodes=4
#SBATCH --mincpus=32
#SBATCH --mem-per-cpu=5000
#SBATCH --output=output.out

module load Python/3.8.6-GCCcore-10.2.0
module load OpenMPI/4.0.5-GCC-10.2.0

source user_directory/PDSFit/pdsfit_env/bin/activate

mpirun -np 128 python -m mpi4py.futures user_directory/PDSFit/PDSFit.py \
configuration_file_directory/configuration_file_name.cfg --mpisupport=1
deactivate
```

   This SLURM file requests 4 nodes with 32 CPUs per node, which corresponds to 128 CPUs in total. Therefore, the '-np' option of OpenMPI is set to 128.

4) Open Terminal and navigate to the directory, in which the slurm file is located (referred to as slurm_file_directory):

```
cd slurm_file_directory
```

5) Run the program:

```
sbatch slurm_file_name.slurm
```

5 Configuration file

The PDSFit configuration file is an ASCII file with the *.cfg* extension that contains all input data of the program. This file can be created and edited using common text editors, such as Notepad++. Since the amount of input data in the configuration file is usually quite large, it is recommended to use the configuration files from the "examples" folder as a template for creating new configuration files. The configuration file has a defined structure that consists of 11 sections. Each section is described below.

5.1 Operation mode

First of all, the operation mode of PDSFit must be specified. There are two main operation modes, the simulation mode (Section 2.5) and the fitting mode (Section 2.6), and one additional operation mode called error analysis mode. The latter mode allows the error analysis (Section 2.7) to be performed separately and at any later time after the fitting procedure. PDSFit can be run in one of these three modes at a time. The operation mode is specified as follows:

```
mode = 0;
```

Source: PDSFit/examples/example/config_sim.cfg

All options for mode are list below:

mode = 0 activates the simulation mode;

mode = 1 activates the fitting mode;

mode = 2 activates the error analysis mode.

5.2 Experimental parameters

The file paths of the PDS time trace(s) and the parameters of PDS experiments are specified under keyword experiments:

```
experiments = (
   {
       name = "offset XX";
       filename = "offsetXX.txt";
       technique = "4pELDOR-rect";
       magnetic_field = 3.3415;
       detection_frequency = 93.999600;
       detection_pulse_lengths = [7, 14];
       pump_frequency = 93.930000;
       pump_pulse_lengths = [14];
       noise_std = 0.0016;
   },
   …
);
```

Source: PDSFit/examples/example1/config_sim.cfg

Table 3. Supported PDS techniques and corresponding parameters.

Technique		Parameters	
Keyword	**Description**	**Keyword**	**Description**
"4pELDOR-rect"	Four-pulse ELDOR with a rectangular pump pulse	magnetic_field	Magnetic field in [T]
		detection_frequency	Detection frequency in [GHz]
		detection_pulse_lengths	The lengths of detection π/2- and π-pulses in [ns]
		pump_frequency	Pump frequency in [GHz]
		pump_pulse_lengths	Pump pulse length in [ns]
"4pELDOR-chirp"	Four-pulse ELDOR with a chirp pump pulse	magnetic_field	Magnetic field in [T]
		detection_frequency	Detection frequency in [GHz]
		detection_pulse_lengths	The lengths of detection π/2- and π-pulses in [ns]
		pump_frequency	Pump frequency in [GHz]
		pump_pulse_lengths	Pump pulse length in [ns]
		pump_frequency_width	The frequency sweep width of a pump pulse in [GHz]
		pump_pulse_rise_times	The rise time of a pump pulse in [ns]
		critical_adiabaticity	The critical adiabaticity of a pump pulse
"5pRIDME-rect"	Five-pulse RIDME with rectangular pulses	magnetic_field	Magnetic field in [T]
		detection_frequency	Detection frequency in [GHz]
		detection_pulse_lengths	The lengths of π/2- and π-pulses in [ns]
		mixing_time	The length of a mixing inter-pulse interval in [μs]
		temperature	The temperature of a RIDME experiment in [K]

Each individual PDS time trace must be enclosed in curly brackets. Within the curly brackets, the following data is provided:

name	The name used to denote the PDS time trace in PDSFit.
filename	The path to an ASCII file which contains the PDS time trace. The first, second, and third columns in this file must contain time points in nanoseconds, the in-phase (real) component of the PDS signal, and the quadrature (imaginary) component of the PDS signal, respectively.
technique	The PDS technique used to record the PDS time trace. The list of supported PDS techniques is given in Table 3.
experimental parameters	Depending on technique, different experimental parameters must be specified. The list of parameters for each supported PDS technique is given in Table 3.

zero_point (optional)	The zero-point of the PDS time trace. It has to be provided only if the PDS time trace is very noisy, and the automatic determination of the zero point fails (Section 2.4).
phase (optional)	The phase of the PDS time trace. It has to be provided only if the automatic optimization of the signal phase fails (Section 2.4).
noise_std (optional)	The standard deviation of noise in the PDS time trace. It needs to be provided only if the quadrature component of the PDS signal is not available (Section 2.4).

5.3 EPR parameters of a spin system

In PDSFit, the EPR spectrum of each electron spin is determined by the Zeeman interaction with the applied static magnetic field, the hyperfine interaction with nuclear spins, and inhomogeneous line broadening. The first two interactions are described by *g*- and *A*-tensors, whose three principal axes are assumed to be collinear. The unresolved anisotropy of the *g*- and *A*-tensors can be taken into account via *g*- and *A*-strains, respectively. *A*-tensor and *A*-strain can be specified for several nuclear spins, including several identical nuclear spins. The abundance of different nuclear isotopes can also be taken into account. The inhomogeneous line broadening is described by a peak-to-peak linewidth. All of these EPR parameters are provided keyword spins:

```
spins = (
   {
       g = [2.0104, 2.0073, 2.0033];
       gStrain = [0.0004, 0.0003, 0.0001];
       n = (1);
       I = (1.0);
       Abund = (1.0);
       A = ([8.0, 6.0, 96.0]);
       AStrain = ([0.0, 0.0, 12.0]);
       lwpp = 22.4;
       T1 = 0;
       g_anisotropy_in_dipolar_coupling = 0;
   },
   {
       g = [2.0104, 2.0073, 2.0033];
       gStrain = [0.0004, 0.0003, 0.0001];
       n = (1);
       I = (1.0);
       Abund = (1.0);
       A = ([8.0, 6.0, 96.0]);
       AStrain = ([0.0, 0.0, 12.0]);
       lwpp = 22.4;
       T1 = 0;
       g_anisotropy_in_dipolar_coupling = 0;
   }
);
```

Source: PDSFit/examples/example1/config_sim.cfg

Here, the EPR parameters of spins A and B (Figure 1) are given one after the other and enclosed in curly brackets. In addition to the EPR parameters, the T_1 relaxation times of both spins need to be specified for the RIDME data analysis. Moreover, both spins are explicitly assigned to "isotropic" or "anisotropic", which determines the equation PDSFit will use to calculate dipolar frequencies (Section 2.2.2). The notations used in spins are explained below:

g — *g*-factor. g must contain three values, g = [g_{xx}, g_{yy}, g_{zz}].

gStrain — *g*-strain (the unresolved anisotropy of the *g*-factor). If nonzero, gStrain must have three values, gStrain = [Δg_{xx}, Δg_{yy}, Δg_{zz}]. Otherwise, notation gStrain = [] must be used.

n — The number of equivalent nuclear spins coupled to the electron spin. Depending on the number of different types of equivalent nuclear spins, the following notations must be used:

n = [] when no nuclear spins are coupled to the electron spin;

n = [n_1] when n_1 identical nuclear spins coupled to the electron spin;

n = [n_1, n_2] when two groups of identical nuclear spins, consisting of n_1 and n_2 spins, respectively, are coupled to the electron spin;

etc.

I — The nuclear quantum number(s) of the nucleus (nuclei) coupled to the electron spin. I is related to n as follows:

I = [] if n = [];

I = [I_1] if n = [n_1];

I = [I_1, I_2] if n = [n_1, n_2];

etc.

Abund — The natural abundance of the nuclear isotope(s) coupled to the electron spin. Abund is related to n as follows:

Abund = [] if n = [];

Abund = [$Abund_1$] if n = [n_1];

Abund = [$Abund_1$, $Abund_2$] if n = [n_1, n_2];

etc.

A — The hyperfine coupling constant (*A*-constant) in [MHz]. Even if an A-tensor is axial or isotropic, all three principal components of this tensor must be provided. A is related to n as follows:

A = [] if n = [];

A = [A_{1xx}, A_{1yy}, A_{1zz}] if n = [n_1];

A = [A_{1xx}, A_{1yy}, A_{1zz}, A_{2xx}, A_{2yy}, A_{2zz}] if n = [n_1, n_2];

etc.

AStrain A-strain in [MHz] (the unresolved anisotropy of the A-tensor). If nonzero, AStrain is related to n as follows:

AStrain = [] if n = [];

AStrain = [ΔA_{1xx}, ΔA_{1yy}, ΔA_{1zz}] if n = [n_1];

AStrain = [ΔA_{1xx}, ΔA_{1yy}, ΔA_{1zz}, ΔA_{2xx}, ΔA_{2yy}, ΔA_{2zz}] if n = [n_1, n_2]; *etc*.

When the A-strain is zero, notation AStrain = [] must be used.

lwpp Peak-to-peak linewidth in [MHz].

T1 Longitudinal (T_1) relaxation time in [μs]. This parameter is required for the analysis of RIDME time traces only. Otherwise, it can be set to 0.

g_anisotropy_in_dipolar_coupling

If set to 0, the g-anisotropy of the spin center is neglected when calculating the dipolar frequencies. If set to 1, the g-anisotropy is taken into account when calculating the dipolar frequencies (Section 2.2.1).

5.4 PDS background parameters

The mathematical model of the PDS background(s) and corresponding parameters are specified under keyword background:

```
background:
{
    model = "exp";
    parameters = {
       decay_constant : {optimize = 1; range = [0,   1  ]; value = 0.05};
       scale_factor   : {optimize = 1; range = [0.8, 1.0]; value = 1   };
    };
};
```

Source: PDSFit/examples/example1/config_sim.cfg

where

model The PDS background model. The list of supported background models can be found in Table 4.

parameters The list of PDS background parameters. η (Section 2.5) is provided alongside with the background parameters. Each line in background_parameters corresponds to a single background parameter or η and is denoted by the corresponding keyword

(Table 4). In addition, each line contains keywords optimize, range, and value. Keyword optimize is used to specify whether the corresponding parameters will be optimized or not. If optimize = 1, the corresponding parameter will be optimized for each PDS time trace, using the values specified in value and range as an optimization range and an initial guess, respectively. If optimize = 0, the corresponding parameter will be set to a fixed value specified under value. In this case, range is irrelevant and has to be set to range = [].

Table 4. Background models and their parameters.

Background model		Background parameters			
Keyword	**Description**[a]	**Keyword**	**Description**[a]	**Range**	**Initial guess**
"exp"	exponential decay	decay_constant	k	[0, 1]	0.05
		scale_factor	η	[0, 1]	1
"stretched_exp"	stretched exponential decay	decay_constant	k	[0, 1]	0.05
		dimension	d	[1, 5]	3
		scale_factor	η	[0, 1]	1
"polynom2"	2nd order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"polynom3"	3rd order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		c3	c_3	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"polynom4"	4th order polynomial	c1	c_1	[-2, 2]	0
		c2	c_2	[-2, 2]	0
		c3	c_3	[-2, 2]	0
		c4	c_4	[-2, 2]	0
		scale_factor	η	[0, 1]	1
"keller"	Keller's exponential decay	k1	k_1	[-1, 1]	0
		k2	k_2	[-1, 1]	0
		scale_factor	η	[0, 1]	1

[a] For definitions, see Table 2.

5.5 Simulation parameters

If the the simulation mode is selected (Section 5.1), simulation parameters must be specified. In accordance to Section 2.5, the simulation parameters include the geometric parameters of a spin system, the parameters of $P(J)$, the parameters of PDS backgrounds, and η. Since the PDS background parameters and η are specified separately (Section 5.4), they are not considered here. The values of the geometric parameters and of $P(J)$ parameters are specified under keyword simulation_parameters:

```
simulation_parameters:
{
    r_mean      = 1.94;
    r_width     = 0.03;
    xi_mean     = 90;
    xi_width    = 10;
    phi_mean    = 180;
    phi_width   = 5;
    alpha_mean  = 180;
    alpha_width = 20;
    beta_mean   = 45;
    beta_width  = 20;
    gamma_mean  = 0;
    gamma_width = 20;
    rel_prob    = 1;
    j_mean      = 0;
    j_width     = 0;
};
```

Source: PDSFit/examples/example1/config_sim.cfg

In simulation_parameters, all parameters are divided into 15 categories associated with $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, $\Delta \gamma$, w, $\langle J \rangle$, and ΔJ (Table 5). Depending on the modality of distributions used for the geometric parameters and J, one or more simulation parameters may correspond to the same category. In the given example, the simulation parameters correspond to the unimodal distributions, resulting in one parameter per category. In contrast, if the parameters' distributions are n-modal ($n \geq 2$) distributions, n - 1 values must be provided for rel_prob, and n values must be provided for each of the other 14 categories. Multiple parameter values corresponding to a single category must be separated by commas and enclosed in round brackets. In the case of the bimodal distributions ($n = 2$), simulation_parameters will look like:

```
simulation_parameters:
{
    r_mean      = (21.5, 26.9);
    r_width     = (0.5, 4.9);
    xi_mean     = (30, 58);
    xi_width    = (18, 3);
    phi_mean    = (0, 0);
    phi_width   = (0, 0);
    alpha_mean  = (0, 0);
    alpha_width = (0, 0);
    beta_mean   = (31, 10);
    beta_width  = (1, 2);
    gamma_mean  = (0, 0);
    gamma_width = (0, 0);
    rel_prob    = 0.46;
    j_mean      = (0, 0);
    j_width     = (0, 0);
};
```

Source: PDSFit/examples/example3/config_sim.cfg

Table 5. 15 categories of the simulation/fitting parameters.

Keyword	Definition	Units	Ranges
r_mean	$\langle r \rangle$ (Chapter 2.1)	Å	[15, 160] Å
r_width	Δr (Chapter 2.1)	Å	[0, 30] Å
xi_mean	$\langle \xi \rangle$ (Chapter 2.1)	degree	[0°, 90°]
xi_width	$\Delta \xi$ (Chapter 2.1)	degree	[0°, 180°]
phi_mean	$\langle \varphi \rangle$ (Chapter 2.1)	degree	[0°, 90°]
phi_width	$\Delta \varphi$ (Chapter 2.1)	degree	[0°, 180°]
alpha_mean	$\langle \alpha \rangle$ (Chapter 2.1)	degree	[0°, 90°]
alpha_width	$\Delta \alpha$ (Chapter 2.1)	degree	[0°, 180°]
beta_mean	$\langle \beta \rangle$ (Chapter 2.1)	degree	[0°, 90°]
beta_width	$\Delta \beta$ (Chapter 2.1)	degree	[0°, 180°]
gamma_mean	$\langle \gamma \rangle$ (Chapter 2.1)	degree	[0°, 90°]
gamma_width	$\Delta \gamma$ (Chapter 2.1)	degree	[0°, 180°]
rel_prob	w (Chapter 2.1)	-	[0, 1]
j_mean	$\langle J \rangle$ (Chapter 2.5)	MHz	[-10, 10] MHz
j_width	ΔJ (Chapter 2.5)	MHz	[0, 20] MHz

5.6 Fitting parameters

If the the fitting mode is selected (Section 5.1), fitting parameters must be specified. In accordance to Section 2.6, the fitting parameters can include the geometric parameters of a spin system, the parameters of $P(J)$, the parameters of PDS backgrounds, and η. Since the PDS background parameters and η are specified separately (Section 5.4), they are not considered here. The fitting parameters that correspond to the geometric parameters and the parameters of $P(J)$ are specified under keyword fitting_parameters:

```
fitting_parameters:
{
    r_mean      : {optimize = (1); range = ([15.0, 25.0]); value = ()   };
    r_width     : {optimize = (1); range = ([ 0.0,  2.0]); value = ()   };
    xi_mean     : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    xi_width    : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    phi_mean    : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    phi_width   : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    alpha_mean  : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    alpha_width : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    beta_mean   : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    beta_width  : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    gamma_mean  : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    gamma_width : {optimize = (1); range = ([ 0.0, 90.0]); value = ()   };
    rel_prob    : {optimize = (0); range = ();             value = (1.0)};
    j_mean      : {optimize = (0); range = ();             value = (0.0)};
    j_width     : {optimize = (0); range = ();             value = (0.0)};
};
```

Source: PDSFit/examples/example1/config_fit.cfg

In analogy to simulation_parameters (Section 5.5), fitting_parameters consist of 15 parameter categories accociated with $\langle r \rangle$, Δr, $\langle \xi \rangle$, $\Delta \xi$, $\langle \varphi \rangle$, $\Delta \varphi$, $\langle \alpha \rangle$, $\Delta \alpha$, $\langle \beta \rangle$, $\Delta \beta$, $\langle \gamma \rangle$, $\Delta \gamma$, w, $\langle J \rangle$, and ΔJ

(Table 5). Each line in fitting_parameters corresponds to a single category and contains its keyword, followed by a colon and curly brackets. Inside the brackets, three additional keywords, namely optimize, range, and value, are specified. Keyword optimize allows to include/exclude the parameters of the each category to/from the fitting parameters. The range in which the selected fitting parameter will be optimized is specified under keyword range. Finally, under keyword value one can provide fixed value for parameter that was excluded from the fitting parameters.

Depending on the modality of distributions used for the geometric parameters of the spin system and J, one or more parameters may correspond to the same category. This, in turn, determines the number of values that must be specified under optimize, range, and value. In the case the unimodal distributions (see the example above), each category includes a single parameter. Consequently, keyword optimize can be specified as follows:

optimize = (1) The corresponding parameter will be included in the fitting parameters.

optimize = (0) The corresponding parameter will be excluded from the fitting parameters.

The corresponding options for range are:

range = ([*a, b*]) If optimize = (1), the corresponding parameter will be optimized within the range [a, b].

range = () If optimize = (0), no ranges are required.

The corresponding options for value are:

value = () If optimize = (1), no value is required.

value = (*c*) If optimize = (0), the corresponding parameter will be set to a fixed value, c.

In the case of the bimodal distributions, each category (except for rel_prob) includes two parameters, and fitting_parameters looks like:

```
fitting_parameters:
{
    r_mean      : {optimize = (1,1); range = ([20.0,  25.0], [25.0,  30.0]); value = ()        };
    r_width     : {optimize = (1,1); range = ([ 0.0,   4.0], [0.00,  12.0]); value = ()        };
    xi_mean     : {optimize = (1,1); range = ([ 0.0,  90.0], [ 0.0,  90.0]); value = ()        };
    xi_width    : {optimize = (1,1); range = ([ 0.0, 180.0], [ 0.0, 180.0]); value = ()        };
    phi_mean    : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
    phi_width   : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
    alpha_mean  : {optimize = (1,1); range = ([ 0.0,  90.0], [ 0.0,  90.0]); value = ()        };
    alpha_width : {optimize = (1,1); range = ([ 0.0, 180.0], [ 0.0, 180.0]); value = ()        };
    beta_mean   : {optimize = (1,1); range = ([ 0.0,  90.0], [ 0.0,  90.0]); value = ()        };
    beta_width  : {optimize = (1,1); range = ([ 0.0, 180.0], [ 0.0, 180.0]); value = ()        };
    gamma_mean  : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
    gamma_width : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
    rel_prob    : {optimize = (1);   range = ([ 0.0,  1.0]);                  value = ()        };
    j_mean      : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
    j_width     : {optimize = (0,0); range = ();                              value = (0.0, 0.0)};
};
```

Source: PDSFit/examples/example3/config_fit.cfg

Here, the optimize keyword of each category has two values (except for rel_prob), i.e., one value per parameter. If optimize = (1,1) is specified, both parameters will be included in the fitting parameters and will be optimized in the ranges specified in range. If optimize = (0,0) is specified,

both parameters will be excluded from the fitting parameters and set to the fixed values specified in value. Finally, if optimize = (1,0) or optimize = (0,1), one parameter will be included in the fitting parameters and optimized in the range specified in range, while the other parameter will be excluded from the fitting parameters and set to the fixed value specified in value.

The above syntax can be easily extrapolated to cases where the parameters' distributions have higher modalities.

5.7 Fitting settings

If the the fitting mode is selected (Section 5.1), the technical details of the fitting procedure are provided under keyword fitting_settings:

```
fitting_settings:
{
    goodness_of_fit = "chi2";
    optimization_method = "ga";
    parameters : {
       number_of_runs = 10,
       generation_size = 128,
       parent_selection = "tournament",
       crossover_probability = 0.3,
       mutation_probability = 0.05,
       crossover_probability_increment = 0,
       mutation_probability_increment = 0,
       single_point_crossover_to_uniform_crossover_ratio = 1,
       creep_mutation_to_uniform_mutation_ratio = 1,
       exchange_probability = 0.5,
       creep_size = 0.1,
       maximum_number_of_generations = 1000,
       maximum_number_of_generations_with_constant_score = 500,
       accepted_score_variation = 0.05,
       maximal_number_of_nma_iterations = 500
    };
};
```

Source: PDSFit/examples/example1/config_fit.cfg

where

goodness_of_fit	Goodness-of-fit parameter. Curretly, only χ^2 (Equation 17), denoted as "chi2", is available.
optimization_method	Optimization method used for fitting PDS time traces. Currently, only the genetic algorithm, which can be extended by the Nelder-Mead algorithm, is available (Sections 2.6 and 2.6.1). This method is denoted as "ga".
parameters	Intrinsic parameters of the optimization method. The intrinsic parameters of the GA are listed Table 6.

Table 6. Intrinsic parameters of the genetic algorithm extended by the Nelder-Mead algorithm.

Keyword	Description	Suggested value
number_of_runs	The number of optimization runs (Section 2.6)	≥ 10
generation_size	The number of chromosomes per generation, N_c (Section 2.6.1)	≥ 10 × N_g (N_g is the number of fitting parameters optimized by the genetic algorithm)
parent_selection	The method used for parent selection (Section 2.6.1)	"tournament" (tournament selection)
crossover_probability	Crossover probability, p_c (Section 2.6.1)	0.5
mutation_probability	Mutation probability, p_m (Section 2.6.1)	0.05
single_point_crossover_to_uniform_crossover_ratio	The ratio between the single-point crossover and the uniform crossover, r_c (Section 2.6.1)	1.0
creep_mutation_to_uniform_mutation_ratio	The ratio between the creep mutation and the uniform mutation, r_m (Section 2.6.1)	1.0
exchange_probability	The probability of gene exchange, p_{ge}, by the uniform crossover (Section 2.6.1)	0.5
creep_size	The creep size, κ, by the creep mutation (Section 2.6.1)	0.1 (= 10 %)
maximum_number_of_generations	The maximum number of generations, $N_{G,max}$ (Section 2.6.1)	> 1000
maximum_number_of_generations_with_constant_score	The maximum number of consecutive generations, $N_{G,const}$, for which the χ^2-value does not change by more than δ (Section 2.6.1)	500
accepted_score_variation	δ (Section 2.6.1)	0.05 (= 5 %)
maximal_number_of_nma_iterations	The number of optimization steps performed by the Nelder-Mead algorithm (Section 2.6)	500

5.8 Error analysis parameters

If the the fitting mode or the error analysis mode is selected (Section 5.1), the error analysis will be perfomed in which the error surfaces will be recorded for different subpaces of fitting parameters specified under keyword error_analysis_parameters:

```
error_analysis_parameters = (
    {
        names = ("r_mean", "r_width");
        components = (1, 1);
        ranges = ();
```

```
    },
    {
        names = ("xi_mean", "xi_width");
        components = (1, 1);
        ranges = ();
    },
    {
        names = ("phi_mean", "phi_width"");
        components = (1, 1);
        ranges = ();
    },
    {
        names = ("alpha_mean", "alpha_width");
        components = (1, 1);
        ranges = ();
    },
    {
        names = ("beta_mean", "beta_width");
        components = (1, 1);
        ranges = ();
    },
    {
        names = ("gamma_mean", "gamma_width");
        components = (1, 1);
        ranges = ();
    }
);
```

Source: PDSFit/examples/example1/config_fit.cfg

Here, each subset of fitting parameters, for which an error surface will be calculated, is enclosed in curly brackets. Within the curly brackets, the following parameters must be specified:

names	The categories of fitting parameters that form a subset. The categories should be identical to the ones specified in fitting_parameters (Table 5).
components	The components (modes) of $P(r)$, $P(\xi)$, $P(\varphi)$, $P(\alpha)$, $P(\beta)$, $P(\gamma)$, or $P(J)$ to which the fitting parameters of a subset correspond. If all distributions are unimodal, the values in components must equal 1 (alternatively, components can be left empty).
ranges	Ranges in which the fitting parameters of a subset will be varied when calculating the error surface. If empty, the ranges specified in fitting_parameters will be used.

To skip the error analysis, error_analysis_parameters should be specified as follows:

```
error_analysis_parameters = ();
```

5.9 Error analysis settings

If the the fitting mode or the error analysis mode is selected (Section 5.1), the technical details of the error analysis are provided under keyword error_analysis_settings:

```
error_analysis_settings:
{
    confidence_interval = 2;
    samples_per_parameter = 100;
    samples_numerical_error = 10000;
    filepath_optimized_parameters = "";
};
```

Source: PDSFit/examples/example1/config_fit.cfg

where

confidence_interval	Confidence level in [σ] used to calculate the errors of the optimized fitting parameters (Section 2.7).
samples_per_parameter	The number of samples per parameter used to record the error surfaces (Section 2.7).
samples_numerical_error	The number of samples used to estimate the contribution of the numerical error to the χ^2-treshold (Section 2.7).
filepath_fitting_parameters (optional)	The full path to the PDSFit file fitting_parameters.dat which contains the optimized values of fitting parameters (optimized geometric model). This file path must be only provided when the error analysis mode is selected (Section 5.1).

5.10 Calculation settings

The technical details of the PDS simulations are specified under keyword calculation_settings:

```
calculation_settings:
{
    integration_method = "monte_carlo";
    number_of_montecarlo_samples = 1000000;
    distribution_types : {
       r = "normal",
       xi = "vonmises",
       phi = "vonmises",
       alpha = "vonmises",
       beta = "vonmises",
       gamma = "vonmises",
       j = "normal"
   };
   excitation_treshold = 1e-9;
   euler_angles_convention = "ZXZ";
};
```

Source: PDSFit/examples/example1/config_sim.cfg

where

integration_method — Numerical integration method used in the simulation of PDS time traces. Currently, only the Monte-Carlo method (Section 2.5), which is denoted as "monte_carlo", is supported.

mc_sample_size — The number of Monte-Carlo samples, N_{MC} (Section 2.5). This parameter determines the precision of the numerical integration performed in Eq. 14. For high precision, N_{MC} must be set as high as possible. On the downside, the calculation time increases linearly with N_{MC}, which sets some limits on its practically realizable values. Our test showed that $N_{MC} = 10^6$ results in a reasonable compromise between the precision and the calculation time.

distributions — Distribution types used for r (keyword r), ξ (keyword xi), φ (keyword phi), α (keyword alpha), β (keyword beta), γ (keyword gamma), and J (keyword j). Supported distribution types include:
"uniform" – uniform distribution;
"normal" – Gaussian (or normal) distribution;
"vonmises" – von Mises distribution (should be used only for angles).

excitation_threshold — A threshold used for $p_{det}(\omega)$ and $p_{pump}(\omega)$ (Section 2.2.1), below which the probabilities are considered to be zero.

euler_angles_convention — A convention used for the Euler angles α, β, and γ (Figure 1). By default, the ZXZ convention (keyword "ZXZ") is used.

5.11 Output settings

Output settings are specified under keyword output:

```
output:
{
    directory = "";
    save_data = 1;
    save_figures = 1;
};
```

Source: PDSFit/examples/example1/config_sim.cfg

where

directory	Output directory. When initialized by an empty string, the output directory will be identical to the directory of the configuration file.
save_data	If set to 1, all data files generated by PDSFit will be saved in directory. Otherwise, this flag should be set to 0.
save_figures	If set to 1, all graphics files generated by PDSFit will be saved in directory. Otherwise, this flag should be set to 0.

6 Output data

The output data of PDSFit is stored as data files and graphics files. The detailed description of all output files is given below.

6.1 Simulation output

6.1.1 Data files

Filename: logfile
Description: This file contains the messages displayed by PDSFit at execution.
Content: Progress report.

Filename: time_trace_{name}.dat
Description: This file contains an experimental PDS time trace and a corresponding simulated time trace. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Time axis in [µs].
Column 2: The in-phase component of an experimental PDS time trace.
Column 3: A simulated PDS time trace.

Filename(*s*)*:* background_{name}.dat
Description: This file contains an experimental PDS time trace and a corresponding simulated background. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Time axis in [µs].
Column 2: The in-phase component of an experimental PDS time trace.
Column 3: A simulated PDS background.

Filename: form_factor_{name}.dat
Description: This file contains the form factors corresponding to experimental and simulated PDS time traces. The form factors are obtained through the division of the experimental and simulated PDS time traces by the corresponding simulated background. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Time axis in [µs].
Column 2: The form factor of an experimental PDS time trace.
Column 3: The form factor of a simulated PDS time trace.

Filename: dipolar_spectrum_{name}.dat
Description: This file contains the dipolar spectra corresponding to an experimental PDS time trace and a simulated PDS time trace. They are obtained via Fourier transform of the form factors. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Frequency axis in [MHz].

Column 2: A dipolar spectrum corresponding to an experimental PDS time trace.
Column 3: A dipolar spectrum corresponding to a simulated PDS time trace.

Filename: dipolar_angle_distr_{name}.dat
Description: This file contains the dipolar angle distribution calculated for the experimental PDS time trace. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Angle axis in degree.
Column 2: A dipolar angle distribution.

Filename: epr_spectrum_{name}.dat
Description: This file contains the simulated EPR spectrum of a spin system. The spectrum is calculated in the frequency domain using the magnetic field value(s) specified in experiments. The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Frequency axis in [GHz].
Column 2: Simulated EPR spectrum.

Filename: detection_bandwidth_{name}.dat
Description: This file contains the excitation profile of a detection pulse sequence (Section 2.2.2). The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Frequency axis in [GHz].
Column 2: Excitation probabilities.

Filename: pump_bandwidth_{name}.dat (only for PELDOR experiments)
Description: This file contains the inversion profile of the pump pulse (Section 2.2.2). The name of the PDS time trace is specified in experiments (Section 5.2).
Content: Column 1: Frequency axis in [GHz].
Column 2: Inversion probabilities.

Filename: background_parameters.dat
Description: This file contains the optimized and/or fixed values of the PDS background parameters.
Content: Column 1: The keywords of background parameters (Table 4).
Column 2: The names of PDS time traces to which background parameters correspond.
Column 3: Optimization flags.
Column 4: The optimized and/or fixed values of background parameters.
Column 5: The negative asymmetric errors of optimized background parameters.
Column 6: The positive asymmetric errors of optimized background parameters.

6.1.2 Graphics files

Filename: time_traces.png

Description: This file depicts simulated PDS time traces together with corresponding experimental PDS time traces.

Filename: backgrounds.png

Description: This file depicts simulated PDS backgrounds together with corresponding experimental PDS time traces.

Filename: form_factors.png

Description: This file depicts form factors corresponding to simulated and experimental PDS time traces.

Filename: dipolar_spectra.png

Description: This file depicts dipolar spectra corresponding to simulated and experimental PDS time traces.

Filename: dipolar_angle_distr.png

Description: This file depicts dipolar angle distributions determined for experimental PDS time traces.

Filename: bandwidths.png

Description: This file depicts the EPR spectrum of a spin system together with the excitation profiles of detection and pump pulses.

6.2 Fitting output

6.2.1 Data files

Filename: logfile

Description: Same as in Section 6.1.1.

Filename: score.dat

Description: This file contains a goodness of fit, e.g., χ^2, as a function of optimization step. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Content:
Column 1: Optimization steps.
Column 2: Goodness of fit values, e.g., χ^2 values (Section 5.7).

Filename: score_all_runs.dat

Description: This file contains a goodness of fit, e.g., χ^2, as a function of optimization step for multiple fitting runs.

Content: Same as in score.dat.

Filename: score_run{*i*}.dat

Description: This file contains a goodness of fit, e.g., χ^2, as a function of optimization step for fitting run no. *i* (*i* = 1 – number_of_trials, see Table 6).

Content: Same as in score.dat.

Filename: fitting_parameters.dat

Description: This file contains the optimized values of fitting parameters. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Content:
Column 1: Categories to which fitting parameters and/or fixed parameters correspond (Table 5).
Column 2: The components of parameters' distributions to which fitting parameters and/or fixed parameters correspond.
Column 3: Optimization flags.
Column 4: The optimized and/or fixed values of fitting parameters.
Column 5: The negative asymmetric errors of optimized fitting parameters.
Column 6: The positive asymmetric errors of optimized fitting parameters.

Filename: fitting_parameters_all_runs.dat

Description: This file contains the optimized values of fitting parameters obtained in multiple fitting runs.

Content: Same as in fitting_parameters.dat.

Filename: fitting_parameters_ run{*i*}.dat

Description: This file contains the optimized values of fitting parameters obtained in fitting run no. *i* (*i* = 1 – number_of_trials, see Table 6).

Content: Same as in fitting_parameters.dat.

Filename: symmetry_related_solutions.dat

Description: This file contains the symmetry-related sets (Section 2.1.1) of the fitting parameters stored in fitting_parameters.dat.

Content:

Column 1:	Categories to which fitting parameters and/or fixed parameters correspond (Table 5).
Column 2:	The components of parameters' distributions to which fitting parameters and/or fixed parameters correspond.
Column 3:	The optimized and/or fixed values of fitting parameters stored in fitting_parameters.dat
Column 4 – Column *n* (*n* = 18 or 34)	The symmetry-related sets of fitting parameters listed in Column 3. The column names, given as "S_1: R_1, S_2: R_2", provide information on how S_1 and S_2 were assigned to spin A or spin B and which transformations R_1 and R_2 were applied to the principal *g*-axes of S_1 and S_2, respectively. Symbols *I*, *Rx*, *Ry*, and *Rz* denote no transformation, 180° rotation about the g_{xx}-axis, 180° rotation about the g_{yy}-axis, and 180° rotation about the g_{zz}-axis, respectively. The last element of each column contains the goodness of fit value that corresponds to the given parameter set.

Filename: time_trace_{name}.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: background_{name}.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: background_parameters.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: form_factor_{name}.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: dipolar_spectrum_{name}.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: dipolar_angle_distr_{name}.dat

Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: epr_spectrum_{name}.dat
Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: detection_bandwidth_{name}.dat
Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: pump_bandwidth_{name}.dat (only for PELDOR experiments)
Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: detection_bandwidth_{name}.dat
Description: Same as in Section 6.1.1. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

6.2.2 Graphics files

Filename: score.png
Description: This file depicts the goodness of fit, e.g., χ^2, as a function of optimization step. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: score_all_runs.png
Description: This file depicts the goodness of fit, e.g., χ^2, as a function of optimization step for multiple fitting runs.

Filename: time_traces.png
Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: backgrounds.png
Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: form_factors.png
Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: dipolar_spectra.png
Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: dipolar_angle_distr.png

Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: bandwidths.png

Description: Same as in Section 6.1.2. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

6.3 Error analysis output

6.3.1 Data files

Filename: error_surface_{*k*}.dat

Description: This file contains an error surface for k^{th} parameter subset specified in error_analysis_parameters (Section 5.8). If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Column 1 – Column *n* (*n* = 1, 2, or 3)	An *n*-dimensional grid for k^{th} parameter subset specified in error_analysis_parameters. Each column name contains the two identifiers of the corresponding fitting parameter (see names and components in Section 5.8).
Column *n* + 1:	χ^2 values.

Filename error_surface_2d_{*k*}.dat

Description: This file contains the projection of an *n*-dimensional (*n* > 2) error surface onto a two-dimensional subspace of fitting parameters. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Column 1:	The values of the first parameter.
Column 2:	The values of the second parameter.
Column 3:	χ^2 values.

Filename error_surface_1d_{*k*}.dat

Description: This file contains the projection of an *n*-dimensional (*n* ≥ 2) error surface onto a one-dimensional subspace of fitting parameters. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Column 1:	Parameter values.
Column 2:	χ^2 values.

6.3.2 Graphics files

Filename: error_surfaces.png

Description: This file depicts the error surfaces of fitting parameters. The optimized values of fitting parameters are depicted as white dots. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: error_surfaces_all_runs.png

Description: This file depicts the same error surfaces as error_surfaces.png and additionally overlays those with the optimized values of fitting parameters obtained in multiple fitting runs (depicted as white markers of different shape).

Filename: error_surfaces_1d.png

Description: This file depicts the one-dimensional error surfaces of fitting parameters. The optimized values of fitting parameters are depicted as white dots. The confidence intervals of fitting parameters are shown by gray shades. If the fitting procedure has been repeated several times, the data corresponds the best fitting run only.

Filename: error_surfaces_1d_all_runs.png

Description: This file depicts the same error surfaces as error_surfaces_1d.png and additionally overlays those with the optimized values of fitting parameters obtained in multiple fitting runs (depicted as white markers of different shape).

7 Examples

7.1 Example 1: Orientation-selective PELDOR time traces

This example is described in Section 3.1 of the manuscript on PDSFit.[1] The PELDOR time traces and configuration file config_fit.cfg used for this example can be found in /PDSFit/examples/example1. In addition, this directory contains the configuration file config_sim.cfg that enables the simulation of the time traces using the predefined geometric model of the spin system under consideration.

7.2 Example 2: Orientation-selective PELDOR time traces in the presence of exchange interaction

This example is described in Section 3.2 of the manuscript on PDSFit.[1] The PELDOR time traces and configuration file config_fit.cfg used for this example can be found in /PDSFit/examples/example2. In addition, this directory contains the configuration file config_sim.cfg that enables the simulation of the time traces using the predefined geometric model of the spin system under consideration.

7.3 Example 3: Orientation-selective PELDOR time traces of a spin system with one anisotropic spin center and two conformational ensembles

This example is described in Section 3.3 of the manuscript on PDSFit.[1] The PELDOR time traces and configuration file config_fit.cfg used for this example can be found in /PDSFit/examples/example3. In addition, this directory contains the configuration file config_sim.cfg that enables the simulation of the time traces using the predefined geometric model of the spin system under consideration.

7.4 Example 4: RIDME time trace of a spin system with one anisotropic spin center

This example is described in Section 3.4 of the manuscript on PDSFit.[1] The RIDME time trace and configuration file config_fit.cfg used for this example can be found in /PDSFit/examples/example4. In addition, this directory contains the configuration file config_sim.cfg that enables the simulation of the time trace using the predefined geometric model of the spin system under consideration.

7.5 Example 5: Orientation-selective PELDOR and RIDME time traces of a spin system with two anisotropic spin centers

This example is described in Section 3.5 of the manuscript on PDSFit.[1] The PDS time traces and configuration file config_fit.cfg used for this example can be found in /PDSFit/examples/example5a (RIDME) and /PDSFit/examples/example5b (PELDOR). In addition, these directories contain the configuration file config_sim.cfg that enables the simulation of the time traces using the predefined geometric model of the spin system under consideration.

8 References

[1] D. Abdullin, P. Rauh Corro, T. Hett, O. Schiemann **2023**, submitted.

[2] O. Schiemann, T. F. Prisner, *Q. Rev. Biophys.* **2007**, *40*, 1.

[3] G. Jeschke, *eMagRes* **2016**, *5*, 1459.

[4] P. P. Borbat, J. H. Freed, *eMagRes* **2017**, 465.

[5] C. R. Timmel, J. R. Harmer (Eds.) *Structure and Bonding*, *Vol. 152*, Springer Berlin Heidelberg, **2013**.

[6] A. D. Milov, K. M. Salikhov, M. D. Shchirov, *Fiz. Tverd. Tela* **1981**, *23*, 975.

[7] R. E. Martin, M. Pannier, F. Diederich, V. Gramlich, M. Hubrich, H. W. Spiess, *Angew. Chem. Int. Ed.* **1998**, *37*, 2833.

[8] L. V. Kulik, S. A. Dzuba, I. A. Grigoryev, Y. Tsvetkov, *Chem. Phys. Lett.* **2001**, *343*, 315.

[9] S. Milikisyants, F. Scarpelli, M. G. Finiguerra, M. Ubbink, M. Huber, *J. Magn. Reson.* **2009**, *201*, 48.

[10] D. Abdullin, G. Hagelueken, O. Schiemann, *Phys. Chem. Chem. Phys.* **2016**, *18*, 10428.

[11] I. Tkach, S. Pornsuwan, C. Höbartner, F. Wachowius, S. T. Sigurdsson, T. Y. Baranova, U. Diederichsen, G. Sicoli, M. Bennati, *Phys. Chem. Chem. Phys.* **2013**, *15*, 3433.

[12] A. D. Milov, A. G. Maryasov, Y. D. Tsvetkov, *Appl. Magn. Reson.* **1998**, *15*, 107.

[13] A. D. Milov, A. B. Ponomarev, Y. Tsvetkov, *Chem. Phys. Lett.* **1984**, *110*, 67.

[14] A. D. Milov, Y. D. Tsvetkov, *Appl. Magn. Reson.* **1997**, *12*, 495.

[15] K. Keller, M. Qi, C. Gmeiner, I. Ritsch, A. Godt, G. Jeschke, A. Savitsky, M. Yulikov, *Phys. Chem. Chem. Phys.* **2019**, *21*, 8228.

[16] D. Abdullin, M. Suchatzki, O. Schiemann, *Appl. Magn. Reson.* **2022**, *53*, 539.

[17] D. Abdullin, H. Matsuoka, M. Yulikov, N. Fleck, C. Klein, S. Spicher, G. Hagelueken, S. Grimme, A. Lützen, O. Schiemann, *Chem. Eur. J.* **2019**, *25*, 8820.

[18] D. Abdullin, P. Brehm, N. Fleck, S. Spicher, S. Grimme, O. Schiemann, *Chem. Eur. J.* **2019**, *25*, 14388.

[19] D. Abdullin, F. Duthie, A. Meyer, E. S. Müller, G. Hagelueken, O. Schiemann, *J. Phys. Chem. B* **2015**, *119*, 13534.

[20] D. Abdullin, O. Schiemann, *ChemPlusChem* **2020**, *85*, 353.

[21] A. Marko, D. Margraf, H. Yu, Y. Mu, G. Stock, T. Prisner, *J. Chem. Phys.* **2009**, *130*, 64102.

[22] A. V. Astashkin, *Methods Enzymol.* **2015**, *563*, 251.

[23] R. A. Stein, A. H. Beth, E. J. Hustedt, *Methods Enzymol.* **2015**, *563*, 531.

[24] G. Jeschke, V. Chechik, P. Ionita, A. Godt, H. Zimmermann, J. Banham, C. R. Timmel, D. Hilger, H. Jung, *Appl. Magn. Reson.* **2006**, *30*, 473.

[25] E. J. Hustedt, F. Marinelli, R. A. Stein, J. D. Faraldo-Gómez, H. S. Mchaourab, *Biophys. J.* **2018**, *115*, 1200.

[26] F. Gao, L. Han, *Comput. Optim. Appl.* **2012**, *51*, 259.

[27] D. Abdullin, G. Hagelueken, R. I. Hunter, G. M. Smith, O. Schiemann, *Mol. Phys.* **2015**, *113*, 544.

[28] S. Katoch, S. S. Chauhan, V. Kumar, *Multimed. Tools Appl.* **2021**, *80*, 8091.

[29] A. Vie, A. M. Kleinnijenhuis, D. J. Farmer, "Qualities, challenges and future of genetic algorithms: a literature review", can be found under https://arxiv.org/pdf/2011.05277, **2020**.

[30] A. Meyer, D. Abdullin, G. Schnakenburg, O. Schiemann, *Phys. Chem. Chem. Phys.* **2016**, *18*, 9262.

[31] M. Kerzhner, D. Abdullin, J. Więcek, H. Matsuoka, G. Hagelueken, O. Schiemann, M. Famulok, *Chem. Eur. J.* **2016**, *22*, 12113.

[32] C. Wuebben, S. Blume, D. Abdullin, D. Brajtenbach, F. Haege, S. Kath-Schorr, O. Schiemann, *Molecules* **2019**, *24*, 4482.

[33] L. M. Stratmann, Y. Kutin, M. Kasanmascheff, G. H. Clever, *Angew. Chem. Int. Ed.* **2021**, *60*, 4939.

[34] D. Abdullin, P. Brehm, N. Fleck, S. Spicher, S. Grimme, O. Schiemann, *Chem. Eur. J.* **2019**, *25*, 14388.

[35] D. Abdullin, *Appl. Magn. Reson.* **2020**, *51*, 725.

[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes in C: The art of scientific computing*, Cambridge University Press, Cambridge, **1992**.

[37] T. H. Edwards, S. Stoll, *J. Magn. Reson.* **2016**, *270*, 87.