# Stochastic Approach to Data Analysis in Fluorescence Correlation Spectroscopy

**Ramachandra Rao,*,† Rajesh Langoju,‡ Michael Gösch,† Per Rigler,§ Alexandre Serov,† and Theo Lasser†**

*Laboratoire d'optique biomédicale, Laboratoire d'imagerie biomédicale, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland, and Physikalische Chemie, Universität Basel, 4056 Basel, Switzerland*

*Received: October 10, 2005; In Final Form: July 19, 2006*

Fluorescence correlation spectroscopy (FCS) has emerged as a powerful technique for measuring low concentrations of fluorescent molecules and their diffusion constants. In FCS, the experimental data is conventionally fit using standard local search techniques, for example, the Marquardt−Levenberg (ML) algorithm. A prerequisite for these categories of algorithms is the sound knowledge of the behavior of fit parameters and in most cases good initial guesses for accurate fitting, otherwise leading to fitting artifacts. For known fit models and with user experience about the behavior of fit parameters, these local search algorithms work extremely well. However, for heterogeneous systems or where automated data analysis is a prerequisite, there is a need to apply a procedure, which treats FCS data fitting as a black box and generates reliable fit parameters with accuracy for the chosen model in hand. We present a computational approach to analyze FCS data by means of a stochastic algorithm for global search called PGSL, an acronym for Probabilistic Global Search Lausanne. This algorithm does not require any initial guesses and does the fitting in terms of searching for solutions by global sampling. It is flexible as well as computationally faster at the same time for multiparameter evaluations. We present the performance study of PGSL for two-component with triplet fits. The statistical study and the goodness of fit criterion for PGSL are also presented. The robustness of PGSL on noisy experimental data for parameter estimation is also verified. We further extend the scope of PGSL by a hybrid analysis wherein the output of PGSL is fed as initial guesses to ML. Reliability studies show that PGSL and the hybrid combination of both perform better than ML for various thresholds of the mean-squared error (MSE).

## I. Introduction

Fluorescence correlation spectroscopy (FCS) has become an important tool for investigating the dynamic properties of single molecules in solution.[1−3] It was introduced in the 1970s as a method for measuring molecular diffusion, reaction kinetics, and flow of fluorescent particles.[4−7]

FCS is based on the statistical analysis of fluorescence intensity fluctuations in solution. It has found widespread applications in the study of various processes such as diffusion in solutions and membranes,[8] rotational diffusion,[9] and singlet−triplet state kinetics.[10] Recent research has proved the power of FCS as a diagnostic tool in biochemical studies.[11]

The data in FCS is conventionally modeled with a finite number of diffusing components and fit with a nonlinear minimization algorithm like the Marquardt−Levenberg method.[12] However, it can generate wrong results for bad initial guesses or for large number of components. It identifies a region of good solutions and follows a downward path (gradient) by accepting only better solutions. In multidimensional solution spaces, it is likely to identify only local minima. Since FCS data are by themselves inherently sensitive to changes in experimental setups,[13] the fitting algorithms have to be robust and accurate in their behavior to avoid unnecessary interpretations in data analysis.[14] Even if the data can be adequately fit by a small number of diffusing components with minimum residual error, this may lead to an unphysical description of the real system under study.

To overcome these drawbacks, we present a data-fitting algorithm for FCS based on a global search method called Probabilistic Global Search Lausanne (PGSL). The algorithm uses random sampling with a probability density function to locate the global minimum of a user-defined objective function. Starting with a uniform probability density function over the entire search space (set of all possible solution points), probabilities are updated dynamically such that a more intensive search is performed in regions where good solutions are found. The PGSL algorithm has a distinct advantage over other local search algorithms such as Marquardt−Levenberg, conjugate gradient, Newton−Raphson technique, and so forth, in so far as these require a good initial guess to reach the global minimum. Tests carried out on complex nonlinear objective functions such as the Lennard-Jones cluster optimization problem, indicate that PGSL performs better in terms of obtaining the success rate and the mean values of the variables estimated as compared with other probabilistic methods such as genetic algorithm and simulated annealing.[15] Because of its proven robustness in identifying the global optima, PGSL has been successfully applied to various areas such as structural mechanics[16] and phase-shifting interferometry.[17] In our implementation in the context of FCS, PGSL finds optimal solutions for FCS data. Subsequently, we address the mean-squared error (MSE) of the fit using the PGSL algorithm.

* To whom correspondence should be addressed. E-mail: ramachandra.rao@epfl.ch.
† Laboratoire d'optique biomédicale.
‡ Laboratoire d'imagerie biomédicale.
§ Universität Basel.

10.1021/jp055763t CCC: $33.50 
Published on Web 08/25/2006

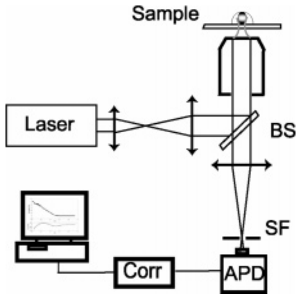


**Figure 1.** Schematic illustration of a typical high-NA objective-based epi-illumination FCS setup. (BS, dichroic beam splitter; SF, spatial filter (pinhole); APD, avalanche photodiode). Emitted fluorescence is detected by an APD and then processed by a multiple tau correlator (Corr).

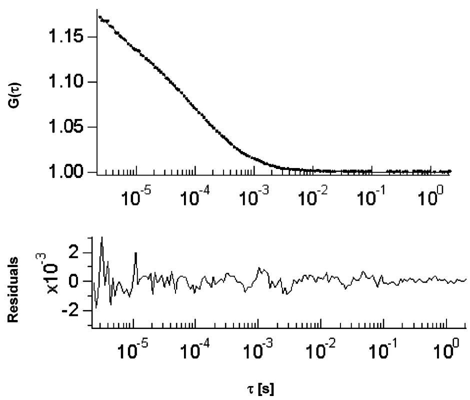


**Figure 2.** FCS curve of an M13 primer labeled with RhG at 10 nM concentration indicating the presence of two components. The dark lines show the fitted curve to our experimental data (dotted lines). The residuals shown indicate the fit quality. The two-component fit yielded the following parameters: $N = 7.2$, $\mathrm{pa} = 0.34$, $\tau_{\mathrm{Da}} = 35.20\ \mu$s, $\tau_{\mathrm{Db}} = 201.4\ \mu$s, $\omega = 5$, $p = 0.31$, and $\tau_{\mathrm{T}} = 2.02\ \mu$s.

In what follows, we first present in section II a general framework by presenting the Experimental Methods and the FCS parameters of a two-component model with triplet transitions along with the tests carried out on them. Section III presents the PGSL algorithm, explains the approach in determining the global minimum for the physical model, the salient features of the classical ML algorithm, and a graphical overview of the internal architectures of PGSL and ML. Section IV presents two case studies wherein the PGSL is applied first to a standard two-component system and second to a parameter estimation with an essential noise contribution. Section V presents a benchmark test of PGSL with ML and presents a reliability test performed on PGSL, ML, and on a hybrid concept merging the PGSL with the ML algorithm.

## II. Experimental Methods, Parameters in FCS Data, and Description of Tests

**i. Experimental Methods.** A FCS configuration is based on a standard confocal setup (Figure 1), the excitation laser light is directed by a dichroic mirror into a high-NA objective, which focuses the light into the sample. The fluorescence emission is collected through the same objective (epi-illumination), filtered by a dichroic beam splitter (BS), and focused onto a pinhole, so that the excited fluorescence light inside the sample is imaged onto the pinhole aperture acting as a spatial filter (SF), which efficiently confines the sampling volume to a diffraction limited size. After the pinhole, the fluorescence signal is collected directly by an avalanche photodiode (APD) and processed by a multiple tau correlator. Our experimental setup was based on the ConfoCor from Carl Zeiss which is basically a confocal microscope designed for FCS. The epi-illumination setup was done using a 40×/1.15 Olympus, Uapo/340 (cover slide corrected), water immersion objective. As a model two-component system, we take a mixture where there is a presence of two fractions, that is, free dye and a labeled primer sample, M13 primer, labeled with rhodamine green at 10 nM concentration for the two-component fits.

These dye molecules crossing the detection volume were excited with an $\mathrm{Ar}^+$ laser at 488 nm. Intensity variations of the fluorescent response were detected across a pinhole of diameter 50 $\mu$m with a single photon counting module and processed with a hardware correlator. Optical power on the sample was around 50 $\mu$W for all measurements during a measurement time of 30 s. A measurement obtained along with its corresponding fit using PGSL is seen in Figure 2 for a representation of the two-component case.

**ii. Parameters in FCS Data.** The autocorrelation $G(\tau)$ of the solute molecules in a small open volume of a dilute solution is defined as

$$G(\tau) = \frac{\langle I(t+\tau)I(t)\rangle}{\langle I(t)\rangle^2} \tag{1}$$

where $\langle\,\rangle$ denotes the time average and $I(t)$ is the instantaneous intensity of the fluorescence present in the detection volume element. A thorough analysis leading to an analytical expression for a two-component case with multiple numbers of differently weighted freely and independently moving molecules is given by[18]

$$G(\tau) = 1 + \frac{1}{N}\left(\frac{1+p}{1-p}\right)\exp\left(\frac{-\tau}{\tau_t}\right) \times \left[\left(\frac{\mathrm{pa}}{1+\dfrac{\tau}{\tau_{\mathrm{Da}}}}\right)\frac{1}{\sqrt{\left(1+\dfrac{\tau}{\omega^2\tau_{\mathrm{Da}}}\right)}} + \left(\frac{1-\mathrm{pa}}{1+\dfrac{\tau}{\tau_{\mathrm{Db}}}}\right)\frac{1}{\sqrt{\left(1+\dfrac{\tau}{\omega^2\tau_{\mathrm{Db}}}\right)}}\right] \tag{2}$$

For a consequent evaluation, we therefore have the following set of parameters, which need to be estimated for the two-component case: $N$, number of molecules in the excitation volume element; $pa$, percentage of species 1 in the confocal volume element; $\tau_{\mathrm{Da}}$, diffusion time of faster diffusing species in microseconds; $\tau_{\mathrm{Db}}$, diffusion time of slower diffusing species in microseconds; $\omega$, structure parameter for the three-dimensional (3-D) Gaussian volume element which is given by the experimental setup and is generally fixed; $p$, fraction of molecules in the triplet states; $\tau_{\mathrm{t}}$, triplet correlation time in microseconds.

Here, we assume that the fraction of molecules in the triplet state, $p$ and the triplet correlation time, $\tau_{\mathrm{t}}$ are the same for both species as there would be no additional information in terms of the photophysics while fitting by having additional parameters for multiple species. Also, in this study, we set the quantum yield to one. This expression assumes a 3-D Gaussian spatial distribution of the probe volume. $N$ is the average number of molecules present in the detection volume element $V = \pi^{3/2}\omega_{xy}{}^2\omega_z$, where $\omega_{xy}$ is the transversal extent and $\omega_z$ is axial extent at which the laser intensity has dropped by $1/e^2$. $\omega$ is consequently defined as $\omega = \omega_z/\omega_{xy}$, while $\tau_{\mathrm{D}} = \omega_{xy}{}^2/4D$ denotes the diffusion time across the sampling region, where $D$ is the diffusion coefficient. Thus, the average concentration of the molecules in the volume element is $C = N/V$.

**iii. Description of Tests.** A Pentium IV 2.4 GHz machine was used for all tests. The programs were written in C and

**TABLE 1: Large Bound and Small Bounds Results for Various NFC Parameters in PGSL**

| original values | [large bounds] | NSDC = 40 NFC = 20 | NSDC = 40 NFC = 40 | [small bounds] | NSDC = 40 NFC = 20 | NSDC = 40 NFC = 40 |
|---|---|---|---|---|---|---|
| $N = 4.5$ | [0.01−100] | 4.508 | 4.503 | [0.1−50] | 4.50 | 4.49 |
| | $((\Delta N)/(N)$ %) | (0.2%) | (0.08%) | $((\Delta N)/(N)$ %) | (0.1%) | (0.06%) |
| $pa$ =0.35 | [0.1−1] | 0.364 | 0.357 | [0.1−1] | 0.363 | 0.344 |
| | ((Δpa/pa) %) | (4.3%) | (2.1%) | ((Δpa/pa) %) | (3.8%) | (1.6%) |
| $\tau_{Da} = 60 \times 10^{-6}$ | [$1 \times 10^{-8}$ to $1 \times 10^{-4}$] | $64.13 \times 10^{-6}$ | $62.06 \times 10^{-6}$ | [$1 \times 10^{-7}$ to $1 \times 10^{-4}$] | $63.87 \times 10^{-6}$ | $58.49 \times 10^{-6}$ |
| | $((\Delta\tau_{Da}/\tau_{Da})$ %) | (6.8%) | (3.4%) | $((\Delta\tau_{Da}/\tau_{Da})$ %) | (6.4%) | (2.4%) |
| $\tau_{Db} = 5 \times 10^{-4}$ | [$1 \times 10^{-7}$ to $1 \times 10^{-3}$] | $5.13 \times 10^{-4}$ | $5.05 \times 10^{-4}$ | [$1 \times 10^{-5}$ to $1 \times 10^{-3}$] | $5.11 \times 10^{-4}$ | $4.95 \times 10^{-4}$ |
| | $((\Delta\tau_{Db}/\tau_{Db})$ %) | (2.6%) | (1.2%) | $((\Delta\tau_{Db}/\tau_{Db})$ %) | (2.2%) | (0.9%) |
| $p = 0.15$ | [0.01−1] | 0.1513 | 0.1513 | [0.1−1] | 0.1511 | 0.149 |
| | $((\Delta p/p)$ %) | (0.9%) | (0.4%) | $((\Delta p/p)$ %) | (0.7%) | (0.3%) |
| $\tau_t = 1.2 \times 10^{-6}$ | [$1 \times 10^{-9}$ to $1 \times 10^{-5}$] | $1.22 \times 10^{-6}$ | $1.21 \times 10^{-6}$ | [$1 \times 10^{-8}$ to $1 \times 10^{-5}$] | $1.22 \times 10^{-6}$ | $1.19 \times 10^{-6}$ |
| | $((\Delta\tau_t/\tau_t)$ %) | (2.0%) | (0.9%) | $((\Delta\tau_t/\tau_t)$ %) | (2.1%) | (0.5%) |
| | MSE → | $1.26 \times 10^{-6}$ | $2.87 \times 10^{-7}$ | | $1.06 \times 10^{-6}$ | $1.65 \times 10^{-7}$ |

interfaced with MATLAB (The Math Works, Inc.). We choose to use a simulated data set for the sake of effective comparison between the two algorithms. The values for the parameters described as the "original values" in Table 1 are $N = 4.5$, $pa = 0.35$, $\tau_{Da} = 60 \times 10^{-6}$ s, $\tau_{Db} = 5 \times 10^{-4}$ s, $\omega = 5$, $p = 0.15$, $\tau_T = 1.2 \times 10^{-6}$ s. Using eq 2, we generate a set of 288 points logarithmically spaced in the decades of time. For a data set of these 288 points as in Case Studies I and II, a typical fitting run with an NSDC of 40 and NFC of 40 took less that 3 s for output, which demonstrates the flexibility as well as the lesser computational cost as well. The number of iterations in PGSL for each of the evaluations is computed as NS (2) × NPUC (1) × NFC (40) × NSDC (40) × number of parameters (6) = 19200. Here, NS = 2 corresponds to an internal default value to the present version of the PGSL implementation used here. For a typical run of NFC = 20 and NSDC = 40, the number of iterations to reach convergence is around 9000 and for NFC = 40 and NSDC = 40 it goes up to 17 000. The only parameters to be determined for PGSL are the values of the Focusing cycles and Subdomain cycles. This fixes the number of iterations required and hence the computation time for a particular run. The sensitivity of the results obtained increases marginally as the number of iterations is increased. The table in the following subsection gives an overview of the quality of the fits obtained for a synthetic data set for the two-component case.

## III. Algorithms

**i. PGSL Algorithm.** We describe the algorithms in terms of the parameters of the two-component case here, that is, for the accurate determination of $N$, $pa$, $\tau_{Da}$, $\tau_{Db}$, $p$, and $\tau_t$. This is done by computing the global minimum of the least-squares error objective function, $\Pi$, defined by

$$\Pi = \sum_{i=1}^{K} [G_{fit}(\tau_i) - G_{data}(\tau_i)]^2 \qquad (3)$$

for $K$ number of sample points and time interval given by $\tau$.

We then utilize the PGSL algorithm, which is based on the direct global search technique.[19] This algorithm operates by organizing optimizations search through four nested cycles, namely, Sampling, Probability updating, Focusing, and Subdomain. Each cycle has a different role to perform while searching for the optimum solution. The user first defines the bounds for each variable that is used with the objective function $\Pi$. The method then searches for the optimum value of the objective function defined in $\Pi$, which is performed by matching the measured $G_{data}(\tau)$ with its predicted counterpart based on the estimated values of the variables $N$, $pa$, $\tau_{Da}$, $\tau_{Db}$, $p$, and $\tau_t$. The algorithm initially generates random values for each variable in the Sampling cycle. Assuming equal probability of finding good solution in the entire search space, the residual error $\Pi$ is evaluated by substituting all generated solutions in eq 3. This allows for selecting all points where the residual error is minimum. Probability-updating and Focusing cycles subsequently refine the search in the neighborhood of good solutions. Convergence to the optimum solution is achieved by means of the Subdomain cycle. The following are the main terms used for describing the PGSL algorithm.

**Solution Point.** A point consists of a value set for each of the variables $N$, $pa$, $\tau_{Da}$, $\tau_{Db}$, $p$, and $\tau_t$.

**Search Space.** Search space is the set of all potential solution points. It is an $M$-dimensional space with an axis corresponding to each variable. $M$ denotes the total number of variables. In the case presented here, $M = 6$. The user defines the minimum and maximum values, commonly known as bounds of variables along each axis. A subset of the search space is called a subdomain.

**Probability Density Function, PDF.** The PDF of a variable is defined in the form of a histogram. The axis represented by the variable is discretized into a fixed number of intervals. Uniform probability distribution is assumed within each interval. The PDF is used to search within a small neighborhood. Since PGSL works by global sampling, there is no point-to-point movement as compared with other random methods.

The function of each cycle is described below:

**Sampling Cycle.** In the sampling cycle, the number of samples evaluated in the sampling cycle (NS) are generated randomly by selecting a value for each variable according to its PDF. This sampling technique resembles the Monte Carlo technique. Each point is evaluated, and the point having the minimum cost, BS (Best Sample), is selected.

**Probability-updating Cycle.** The sampling cycle is repeated NPUC (number of iterations in the probability-updating cycles) times, and after each iteration, the PDF of each variable is modified using a probability-updating algorithm. This ensures that the sampling frequencies in regions containing good points are increased. In the probability-updating algorithm, the interval containing the value of the variable in BS is located. The evolution of the PDF for a variable after several sampling cycles is illustrated in Figure 3.

**Focusing Cycle.** The probability-updating cycle is repeated NFC (number of focusing cycles) times, and after each iteration, the current best point, CBEST, is selected. The PDF is updated by first locating the interval containing the value of each variable in CBEST. This probability is uniformly divided into its subintervals. The widths of these subintervals are calculated such that the PDF decays exponentially away from it. After subdivisions, intervals no longer have the same width and probabilities are heavily concentrated near the current best. The evolution

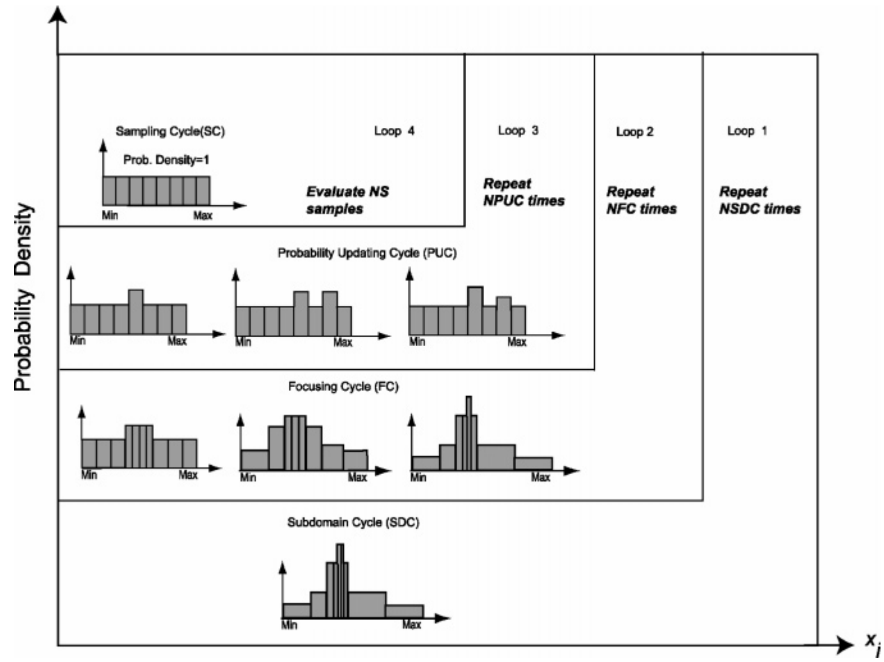


**Figure 3.** Illustration of the development of the probability density function of one optimization variable Xi during four nested loops of PGSL. This schematic summarizes the overall internal architecture of the PGSL.

of PDF after several probability-updating cycles is illustrated in Figure 3. Assume that the value of the variable in the best solution found in the first probability-updating cycle is Xb1. The interval containing this value is subdivided into four parts and is assigned 50% probability. The remaining probability is distributed to the other intervals according to an exponentially decaying function. Due to this probability distribution, 50% of variables in subsequent samples lie within their respective best intervals. This results in exploration of alternative values of some variables keeping the values of other variables in the best regions. Assume that the value of the variable in the best solution found in the second updating cycle is Xb2. The interval containing this value is further subdivided (for clarity, intervals are not shown to scale in the figure). Probability densities increase enormously due to fine division of intervals after many probability-updating cycles.

**Subdomain Cycle.** In the subdomain cycle, the focusing cycle is repeated NSDC (number of subdomain cycles) times, and at the end of each iteration, the current space search is modified. In the beginning, the entire space is searched, but in subsequent iterations, a subdomain of a smaller width is selected for search. The size of the subdomain decreases gradually, and the solution converges to a point. The scale factor is dynamically chosen such that there is no premature convergence. This is, however, not the case in ML (see section ii).

Each cycle therefore serves a different purpose in the search for a global optimum. The sampling cycle permits a more uniform and exhaustive search over the entire search space than the other cycles. The probability-updating and focusing cycles refine the search in the neighborhood of good solutions. Convergence is achieved by the subdomain cycle.

The parameter study revealed that only the values of the focusing cycles and subdomain cycles need to be adjusted to fix the total number of evaluations of the objective function in hand. This underlines the ease and simplicity of fixing PGSL parameters.

**a. Approaches for Determining Global Minimum.** The selection of an appropriate methodology is the key for successful data analysis. Before narrowing down to a particular method, it is imperative to study the topology of the parameter space for $\Pi$. Figure 4 shows a plot for $\Pi$ in eq 3 from the experimental measure of Figure 2 obtained with an M13 labeled with RhG at 10 nM concentration on a cover glass. This plot is generated with two sensitive variables, namely, $N$ and $\tau_{Da}$, where the values of $N$ are varied from 0 to 10 and $\tau_{Da}$ from $1 \times 10^{-9}$ to $1 \times 10^{3}$. This log plot shows the presence of many local minimas and only one global minimum. There would be many minima for the entire multiparameter space of $\Pi$ when one considers all the five variables simultaneously. The figure presented here considers only two of the sensitive variables for the sake of display. With traditional analysis, the FCS experimenter often encounters this situation while fitting experimental data when multiple solution sets are obtained for repetitive fitting on the same data set. The choice is then to obtain multiple data sets from many measurements and to do the cumulative analysis of the data therein or have multiple fitting sessions on the same data set with varying good initial guesses.

The proposed PGSL algorithm is ideal for the error function $\Pi$, since no initial guesses are required, and although the bounds need to be defined, the bounds do not require a difficult constraint selection.

For instance, for the variables in eq 3 is defined as follows

$$0 < N \leq 100,\ 0.1 \leq pa \leq 1,\ 1 \times 10^{-8} \leq \tau_{Da} \leq 1 \times 10^{-4},\ 1 \times 10^{-8} \leq \tau_{Db} \leq 1 \times 10^{-4},\ 0.01 \leq p \leq 1$$

and

$$1 \times 10^{-8} \leq \tau_t \leq 1 \times 10^{-5} \qquad (4)$$

where the maximum and minimum values above represent the theoretical ranges of values possible for a particular dye molecule under consideration.

**Boundary Sensitivity.** Boundary sensitivity is the dependence of the parameter estimates for the range of values given to the PGSL algorithm. They are highly flexible and need to be modified appropriately with the system in hand. We can also set the lower bounds of $pa$ and $p$ to zero and get similar results for the two-component fit with the triplet model here. In case these parameters are not desired for evaluation, then one could

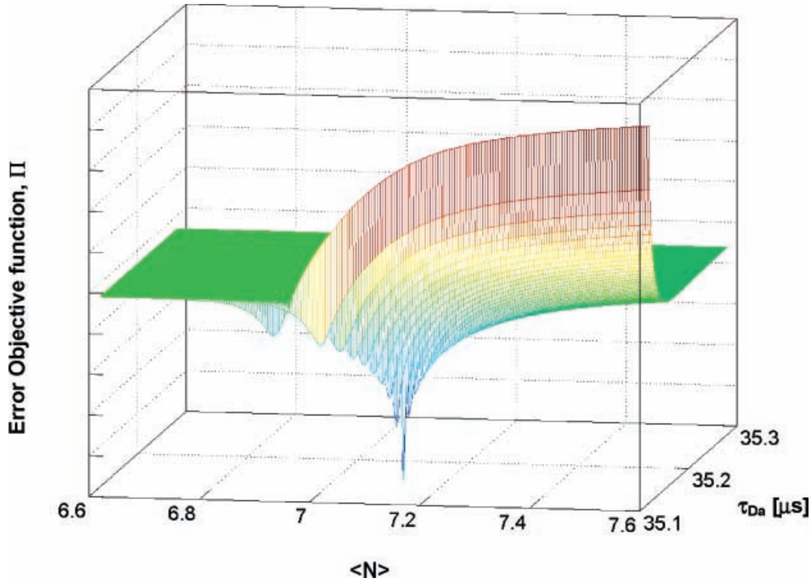


**Figure 4.** Magnitude of error for objective function Π defined in eq 3. The log plot shows the presence of several minima close to one global minimum for the choice of two parameters only for our two-component model.

choose the no triplet or the one-component model accordingly. The structure parameter $\omega$ is fixed in the analysis.

**b. Mean Squared Error (MSE) of Fit in PGSL.** It is desired for a fitting algorithm that it take into account the error of individual points.[20] We utilize the simple definition of MSE to evaluate the quality of fit in PGSL here.

$$\mathrm{MSE} = \frac{1}{K}\sum_{i=1}^{K}[G_{\mathrm{fit}}(\tau_i) - G_{\mathrm{data}}(\tau_i)]^2 \qquad (5)$$

This value measures the difference between the fitted function $G_{\mathrm{fit}}(\tau)$ and the experimental data $G_{\mathrm{data}}(\tau)$ at every time interval $\tau$, weighted by length of data points $K$. In Table 1, we see the performance of PGSL under various Focusing cycles (NFC) for the six parameters of the two-component case. The fitted values and also the relative percentage errors with respect to the initial original values are presented here. The values presented are a statistical value obtained after running the algorithm for 100 times. The choice of appropriate bounds also influences the MSE as seen in Table 1.

**ii. Marquardt−Levenberg Algorithm.** The ML algorithm[12,21] is briefly described in this section and is explained with regard to the two-component equation in eq 2 written in terms of the parameter **u**

$$G(\mathbf{u},\tau) = 1 + \frac{1}{u_1}\left(\frac{1+u_5}{1-u_5}\right)\exp\left(\frac{-\tau}{u_6}\right)\times \left[\left(\frac{u_2}{1+\frac{\tau}{u_3}}\right)\frac{1}{\sqrt{\left(1+\frac{\tau}{\omega^2 u_3}\right)}} + \left(\frac{1-u_2}{1+\frac{\tau}{u_4}}\right)\frac{1}{\sqrt{\left(1+\frac{\tau}{\omega^2 u_4}\right)}}\right] \qquad (6)$$

where, $u_1$, $u_2$, $u_3$, $u_4$, $u_5$, and $u_6$ the elements of **u**, represent $N$, $pa$, $\tau_{\mathrm{Da}}$, $\tau_{\mathrm{Db}}$, $p$, and $\tau_t$, respectively. To obtain any of the elements in **u**, we must calculate the residual error[21] $E(\mathbf{u})$

$$E(\mathbf{u}) = \sum_{i=1}^{K}(G(\mathbf{u},\tau_i) - d_i)^2 \qquad (7)$$

for $K$ number of sample points and the corresponding time interval given by $\tau_i$ and where $d_i$ is the measured value at $\tau_i$.

If $(\mathbf{u})^l$ is the initial guess for the parameter set **u**, the ML iteration step is given by

$$[\mathbf{J}((\mathbf{u})^l)^T\mathbf{J}((\mathbf{u})^l) + \lambda\mathbf{I}]((\mathbf{u})^{l+1} - (\mathbf{u})^l) = -\mathbf{J}((\mathbf{u})^l)^T\mathbf{E}((\mathbf{u})^l) \qquad (8)$$

where $\mathbf{J}(\mathbf{u})$ is the Jacobian matrix, which is defined as

$$\mathbf{J}(\mathbf{u}) = \begin{bmatrix} \frac{\partial E_1(\mathbf{u})}{\partial N} & \frac{\partial E_1(\mathbf{u})}{\partial pa} & \frac{\partial E_1(\mathbf{u})}{\partial \tau_{\mathrm{Da}}} & \frac{\partial E_1(\mathbf{u})}{\partial \tau_{\mathrm{Db}}} & \frac{\partial E_1(\mathbf{u})}{\partial p} & \frac{\partial E_1(\mathbf{u})}{\partial \tau_t} \\ \frac{\partial E_2(\mathbf{u})}{\partial N} & \frac{\partial E_2(\mathbf{u})}{\partial pa} & \frac{\partial E_2(\mathbf{u})}{\partial \tau_{\mathrm{Da}}} & \frac{\partial E_2(\mathbf{u})}{\partial \tau_{\mathrm{Db}}} & \frac{\partial E_2(\mathbf{u})}{\partial p} & \frac{\partial E_2(\mathbf{u})}{\partial \tau_t} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial E_K(\mathbf{u})}{\partial N} & \frac{\partial E_K(\mathbf{u})}{\partial pa} & \frac{\partial E_K(\mathbf{u})}{\partial \tau_{\mathrm{Da}}} & \frac{\partial E_K(\mathbf{u})}{\partial \tau_{\mathrm{Db}}} & \frac{\partial E_K(\mathbf{u})}{\partial p} & \frac{\partial E_K(\mathbf{u})}{\partial \tau_t} \end{bmatrix} \qquad (9)$$

The parameter $\lambda$ is dynamically adjusted during the course of minimization where $l$ is the current iteration and $l + 1$ is the next iteration. $T$ is the transpose operator and **I** is the identity matrix. The iteration process continues until some pre-specified termination criterion has been met, such as a given change in the value of the parameter **u** or a limit on the number of iterations.

Used in this way, the ML algorithm allows for computing of the optimal parameter **u**. We use the ML implementation available in the "Optimization Toolbox" in MATLAB with no changes in the default values for the various internal parameters described therein.

**iii. Illustration of the Internal Architectures of PGSL and ML.** PGSL has several interesting features not similar to other algorithms. First, it works by global sampling, thereby avoiding point-to-point improvement in a region around a current point. Second, it uses histograms for the PDF—a discontinuous function with multiple peaks. This allows fine control over probabilities in small regions by subdividing intervals. Third, the shape and form of the PDF can be changed by subdividing intervals as well as by directly increasing the probabilities of intervals. This is different from the normal practice of changing the standard deviation in other methods.

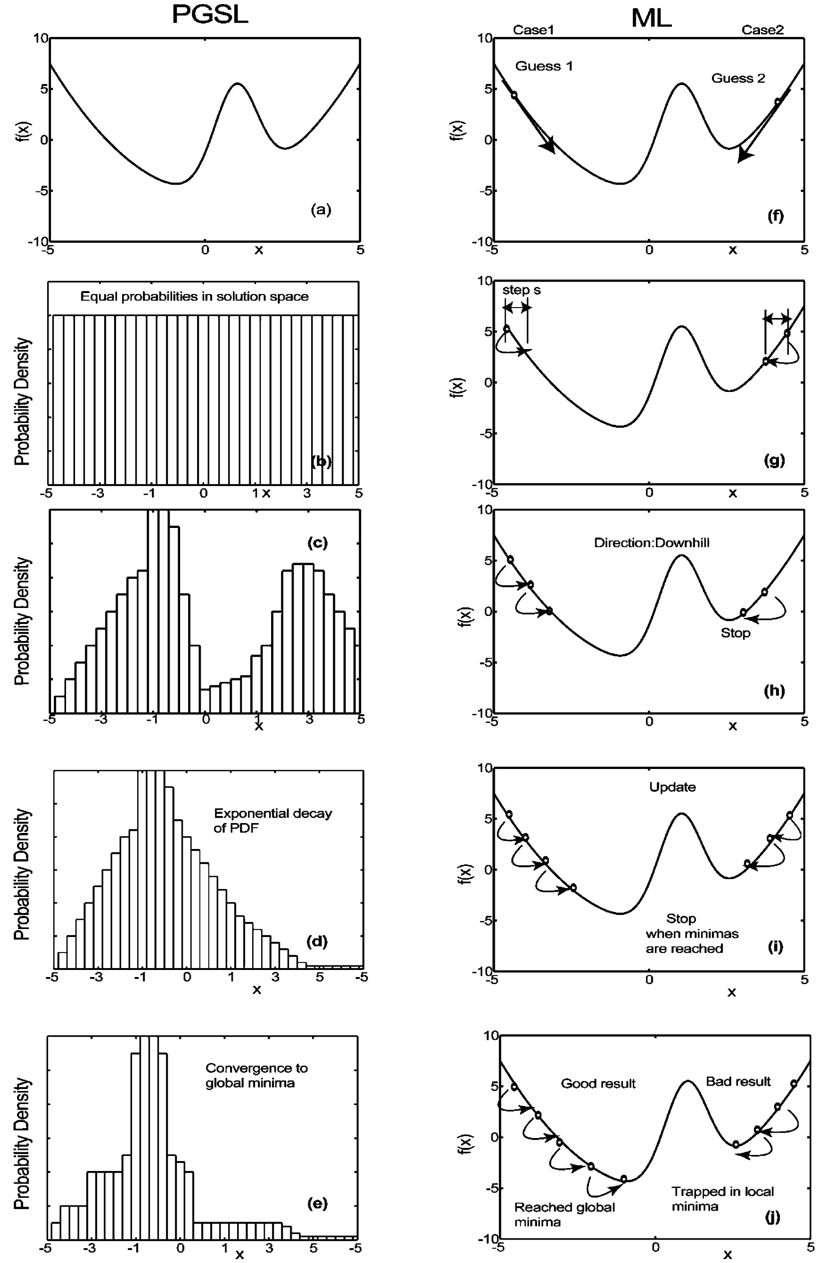


**Figure 5.** Illustration of basic differences between PGSL and ML. Points are randomly generated using a PDF in PGSL. Probabilities are increased in regions where good solutions are found: (a) a function $f(x)$ in a single variable, (b) a uniform PDF, (c) a PDF with higher probabilities in regions containing good solutions, (d) evolution of the PDF into regions with better solutions and the subsequent exponential decay of the PDF, and (e) search space progressively narrowed by converging to points in a subdomain of smaller size centered on the best point. The ML operates on the premise of intense searching in (f) initial guesses for two cases here considered by the seeking the minimum of the function, (g) the step size (s) is chosen in the direction of steepest descent, (h) chosen in the direction of steepest descent iteration continues towards the minimum, (i) point updated in the direction of the slope until minimum is reached, and (j) "stop" when the function hits the minimum value possible.

PGSL therefore works directly on the solution search space and does not seek any gradients for approaching the global minima. This key feature enables PGSL to generate an optimal solution set for any noise level by avoiding getting trapped into local minima irrespective of the complexity of the error objective function (see parts a−e of Figure 5).

Whereas a local search technique like the ML method finds good solutions using an exhaustive search over a constrained space provided, they are given good initial guesses. It is therefore a nontrivial task to compare two different categories of algorithms wherein their individual internal architectures differ as markedly as those seen here (see parts f−j of Figure 5).

## IV. Case Studies

**i. Case Study 1: PGSL Applied to a Two-Component System of Interacting Fluorophores.** On the left-hand side of Table 1, the bounds are varied to the maximum extent as permissible by the parameters of interest for the case of large bounds. This clearly shows the strength of PGSL in generating acceptable solutions with no initial guesses for arbitrarily large bounds.

On the right-hand side of Table 1, we see the performance of PGSL under various focusing cycles (NFC) with small bounds as compared with the previous case. The differences are mainly for the bounds in the various diffusion times for the

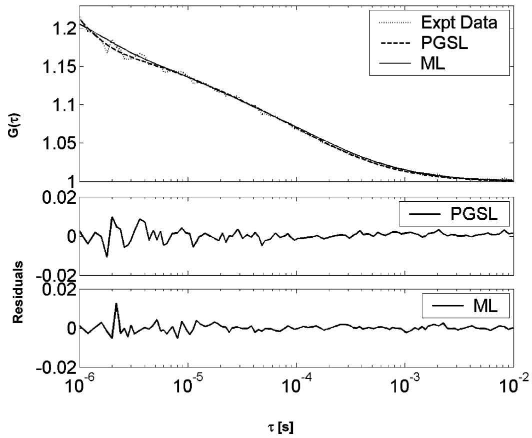


**Figure 6.** Fitting of experimental FCS data with PGSL and the Marquardt–Levenberg algorithm: (a) the experimental data with the fits in ML and PGSL with eq 2, (b) the residuals using PGSL, and (c) the residuals using the ML algorithm.

model under consideration here. We see an increase in the sensitivity of the obtained parameters.

The MSE is different for the various cases and gives the quality of fit for every particular choice of NFC and NSDC.

We have good solutions for both high and low required accuracies. The overall setting of the NSDC being 40 and an NFC of 20 or 40 provides optimal fits here. These results were also confirmed by testing them on several different models as well.

**ii. Case Study 2: PGSL Applied to Parameter Estimation with Noisy Data.** FCS is susceptible to various noise sources such as intrinsic photon shot noise dependent on the average count rate, excitation power instabilities, Raman scattering, and background fluorescence.[22–24] Also, depending on the type of individual setups or application (two-photon or intracellular systems) at hand, the contribution of various noise sources leading to improper fitting is another source of systematic error leading to erroneous interpretation of the obtained data.

In our context of addressing fitting artifacts especially in noisy data, we present the main influence of averaging noise here. The following measurement (Figure 6) was done on a 3 s measurement time interval on the M13–RhG sample as before at 10 nM concentration. The data mainly contains the overall effect of shot noise (governed by Poisson statistics) and

**TABLE 2: Comparison of PGSL and ML for a Noisy Experimental Data Set**

| | | $N$ | $pa$ | $\tau_{Da}$ ($\mu$s) | $\tau_{Db}$ ($\mu$s) | $p$ | $\tau_t$ ($\mu$s) | MSE |
|---|---|---|---|---|---|---|---|---|
| 30 s (ref) | PGSL | 7.2 | 0.34 | 35.20 | 201.4 | 0.31 | 2.02 | $4.32 \times 10^{-6}$ |
| 3 s | PGSL | 5.56 | 0.40 | 20.6 | 151.9 | 0.5 | 1.6 | $7.5 \times 10^{-4}$ |
| 3 s | ML | 5.52 | 0.54 | 192.3 | 11.2 | 0.28 | 1.33 | $4.4 \times 10^{-4}$ |

averaging noise (for the long lag times during the finite measurement interval $T$).

We clearly see in Table 2 that the values obtained for ML are not satisfactory for $\tau_{Da}$, $\tau_{Db}$, and $\tau_t$ for this particular fit here, although the residuals look satisfactory. It is evident that the ML has converged to a wrong minimum in the six-parameter space here. This is an illustration, where the experimenter has to refit the data with varying initial guesses and bounds until a good fit is obtained. For this simple case, it is possible to obtain better results for ML by fixing the faster diffusing parameter ($\tau_{Da}$) for the standard dye label attached by undertaking calibration measurements on the setup used. Therefore, with good initial guesses and prior knowledge of system behavior, ML does perform very well. However, in cases for measurements in, for example, living cells where additional noise terms are also an issue to be considered,[25] the very choice and reliability of good initial guesses would be a nontrivial task.

## V. Benchmark Tests between PGSL and ML

In this section, we benchmark and evaluate the performance of PGSL and ML for the objective function defined in eq 3. This is done by obtaining the best MSE for three different threshold values in both the situations.

**i. Description of Tests and Results in PGSL vs ML. PGSL Algorithm.** It is run for two different cases for large and small bounds (as in Table 1) for the given simulated data. The initial seed values for starting the solution search are randomly varied for each iteration for a fixed NSDC of 40 and an NFC of 40. This ensures that a new random number starts the global search process in the solution space for every iteration.

**ML Algorithm.** The initial guesses for the two sensitive parameters, $N$ and $\tau_{Da,}$ are automatically generated by the correlogram itself. The initial guess for $N$ follows from the simple relation $G(\tau \rightarrow 0) = 1/N$, while for $\tau_{Da}$ the time value corresponding to $G(\tau_{Da}) = {}^1/_2G(0)$ is chosen. For 100 runs, the

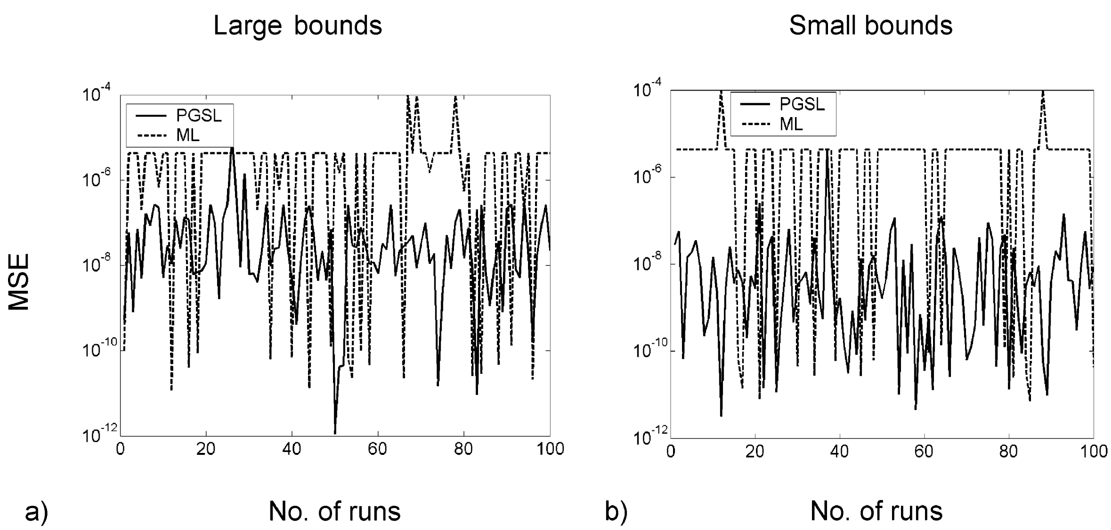


**Figure 7.** Mean-squared error plots for the data evaluated comparing the performances of PGSL (dark lines) and ML (dotted lines) for (a) large bounds and (b) small bounds.

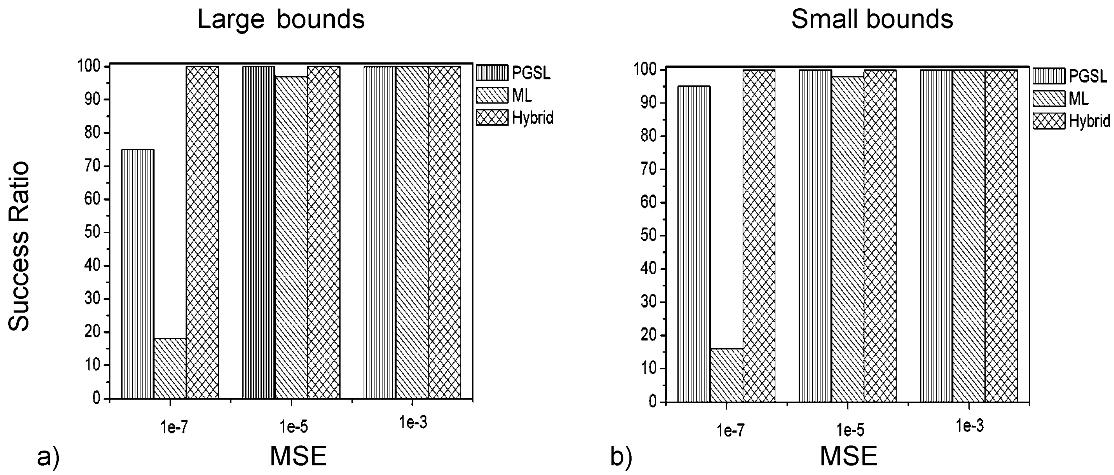


**Figure 8.** Reliability study of obtaining the best mean-squared error for the data is seen in the bar plots comparing the performances of PGSL, ML, and their hybrid: (a) large bounds and (b) small bounds. PGSL outperforms ML and results in a better MSE value, indicating better fit quality.

initial values are randomly varied around the above chosen mean value so that we have a different start value at each iteration for $\tau_{Da}$. For $\tau_{Db}$, the value chosen is twice that of the $\tau_{Da}$. The triplet fraction $p$ is chosen at 10% while for $\tau_t$ a typical value of 1 $\mu$s is given as the initial guess value.

**Hybrid Algorithm.** We further explore the possibility of combining the mutual abilities of PGSL and ML and call it a hybrid. Here, the output of PGSL is fed as an initial guess to ML, and the resultant hybrid performance is evaluated here as well.

After performing 100 runs for varying guesses for ML and by varying the initial random value for PGSL, we obtain the varying MSE for large and small bounds can be seen again in Figure 7.

We see that PGSL outperforms ML consistently in terms of the MSE throughout. This is further summarized in terms of the success ratio in Figure 8.

The success ratio is defined as the percentage of iteration steps which gives the MSE below the predefined threshold.

In the bar graphs, we have fixed three threshold values for the MSE. For each iteration, in all the three cases, the fit generates a particular value of the MSE. The sensitivity of our iterations was evaluated by checking on the various MSE thresholds. For a MSE threshold value such as $1 \times 10^{-7}$, which corresponds to a better quality of the fit obtained, we see that the success ratio is lower as compared with the other two cases. Correspondingly, we obtain a higher success ratio when we choose to lower our fit quality (as for an MSE threshold of $1 \times 10^{-3}$).

For both large and small bounds, we see that PGSL outperforms ML throughout and their corresponding hybrid approach moves toward the limit of performance as that of PGSL. These tests conclusively prove that PGSL is a robust search technique that performs well in spaces with multiple local minima.

Because of the generality of the procedures discussed here, we believe that PGSL is useful for various FCS fitting models. Simple guidelines leading to parameter adaptation as seen above depending on the models and also on the physical systems under study would lead to even better convergence in PGSL. This would then result in faster parameter retrieval for large data sets as in assays.[26] Also, for highly heterogeneous systems with multiple diffusing species, PGSL would provide a bias-free fitting of the data for the model chosen on hand since it would always generate acceptable solutions for the chosen model without the need for a priori user knowledge of the physical system.

## VI. Conclusions

To conclude, we have described a novel stochastic data analysis concept applicable to the determination of fitting parameters in FCS experiments. For the two-component case, the FCS parameters are determined with high precision by the application of this algorithm. The fitting values are given the lower and upper bounds with no initial guesses thereby giving the experimentalist the confidence in data validation when experiments are to be planned requiring the application of various models. It is therefore useful in experimental situations wherein accurate determination of parameters with no fitting artifacts from various physical models is also a prime criterion. We also determine the quality of the fit from the MSE analysis. Statistical studies further demonstrate the capability of the method to estimate the fit parameters with greater confidence along with existing standard gradient-based methods. The robustness of PGSL on noisy experimental data for parameter estimation is also shown. Therefore, PGSL appears to be a viable tool for unbiased parameter retrieval of FCS data.

**Acknowledgment.** We would like to thank Benny Raphael and Per Thyberg for valuable discussions and suggestions. We also acknowledge the positive feedback of one of the reviewers toward the effective presentation of this research. This research is partly funded by the Swiss National Science Foundation. Program Availability: The PGSL programs are available free of charge either directly from the authors (e-mail or conventional mail) or can be downloaded from the laboratory website: http://lob.epfl.ch/page58512.html.

## References and Notes

(1) Magde, D.; Webb, W. W.; Elson, E. *Phys. Rev. Lett.* **1972**, *29*, 705.
(2) Magde, D.; Elson, E. L.; Webb, W. W. *Biopolymers* **1974**, *13*, 29.
(3) Magde, D. *Q. Rev. Biophys.* **1976**, *9*, 35.
(4) Aragon, S. R.; Pecora, R. *Biopolymers* **1975**, *14*, 119.
(5) Ehrenberg, M.; Rigler, R. *Q. Rev. Biophys.* **1976**, *9*, 69.
(6) Ehrenberg, M.; Rigler, R. *Chem. Phys.* **1974**, *4*, 390.
(7) Magde, D.; Elson, E. L. *Biopolymers* **1978**, *17*, 361.
(8) Pick, H.; Preuss, A. K.; Mayer, M.; Wohland, T.; Hovius, R.; Vogel, H. *Biochemistry* **2003**, *42*, 877.
(9) Ehrenberg, M.; Rigler, R. *Chem. Phys.* **1974**, *4*, 390.

(10) Widengren, J.; Mets, U.; Rigler, R. *J. Phys. Chem.* **1995**, *99*, 13368.
(11) Cluzel, P.; Surette, M.; Leibler, S. *Science* **2000**, *287*, 1652.
(12) Marquardt, D. W. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431.
(13) Enderlein, J.; Gregor, I.; Patra, D.; Fitter, J. *Curr. Pharm. Biotechnol.* **2004**, *5*, 155.
(14) Hess, S. T.; Webb, W. W. *Biophys. J.* **2002**, *83*, 2300.
(15) Raphael, B.; Smith, I. F. C. *Appl. Math. Comput.* **2003**, *146*, 729.
(16) Robert-Nicoud, Y. R. I.; Raphael, B.; Smith, I. F. C. *J. Comput. Civ. Eng.* **2005**, *19*, 239.
(17) Patil, A.; Raphael, B.; Rastogi, P. *Opt. Lett.* **2004**, *29*, 1381.
(18) Krichevsky, O.; Bonnet, G. *Rep. Prog. Phys.* **2002**, *65*, 251.
(19) Masri, S. F.; Bekey, G. A.; Safford, F. B. *Appl. Math. Comput.* **1980**, *7*, 353.
(20) Wohland, T.; Rigler, R.; Vogel, H. *Biophys. J.* **2001**, *80*, 2987.
(21) Aster, R.; Borchers, B.; Thurber, C. *Parameter Estimation and Inverse Problems*; Academic Press: Amsterdam, The Netherlands, 2004; p 184.
(22) Koppel, D. E. *Phys. Rev. A* **1974**, *10*, 1938.
(23) Qian, H. *Biophys. Chem.* **1990**, *38*, 49.
(24) Kask, P.; Gunther, R.; Axhausen, P. *Eur. Biophys. J. Biophys. Lett.* **1997**, *25*, 163.
(25) Schwille, P.; Haupts, U.; Maiti, S.; Webb, W. W. *Biophys. J.* **1999**, *77*, 2251.
(26) Eggeling, C.; Berger, S.; Brand, L.; Fries, J. R.; Schaffer, J.; Volkmer, A.; Seidel, C. A. M. *J. Biotechnol.* **2001**, *86*, 163.