Chapter 5

Algorithmic challenges

Introductory comments. For model discrimination of dynamical systems as introduced in (2.1), the implementation of the model–data–overlap consists out of two major parts: First the calculation of the model variability $\mathcal{M}$, which is mainly a density propagation problem, and second, the optimization of the overlap functional $\mathcal{F}_{\mathcal{O}}$ itself.

Optimization. The biggest challenge within the optimization lies in the number of entities that are considered. For the optimization within classical parameter estimation, the parameters themselves have to be found. After turning to distributed parameters, one now has to find the optimal distribution's hyperparameters. For normally distributed parameters, one now has to optimizes the mean and the variance, which amounts to twice as many entities to estimate compared to the classical case.

Propagation. The second problem concerns the model variability propagation scheme. Three different propagation schemes are suggested in this chapter. The first approach (section 5.1) linearly approximates the propagation of model variability, the second one (section 5.2) suggests sampling the initial parameter distribution and propagate the bundle of trajectories, and the third one (section 5.3) is a sophisticated non–linear propagation scheme that combines a stepwise linear propagation with error-control that compensates the loss of nonlinearity. For the second and third propagation scheme, the choice of the optimization depends on the considered application. The first approach can be combined with a Gauss–Newton algorithm.

General comments. The time propagation of densities, as needed to calculate the model variability distribution, surfaces in many applications and translates

into numerous settings. For ordinary differential equations like (2.1), the problem is translated into to differential equations with stationary parameter distribution functions and is referred to as *random differential equations*.

Theoretical and numerical investigations on this topic in past decades can be broadly divided in three groups of approaches. The first group is represented by Monte Carlo methods based on the sampling of the parameter space and subsequent solution of the differential equations for each of the sampled parameters (c.f. [225], section 5.2). While it is a method of choice for problems with many parameters and degrees of freedom in order to avoid the "curse of dimension", the questions of numerical accuracy, applicability and adaptivity in higher dimensional cases still remain unclear. The second group of methods is known as the stochastic finite elements (SFEMs) approach, for example, Galerkin methods (c.f. [133, 140, 168]). These methods are based on the assumption that the overall statistical response of the system under consideration can be represented as a linear combination of orthogonal basis functions. However all these methods were designed for predominantly problems with small parametric variations and small number of parameters under consideration. These approaches cannot be applied to higher dimensional problems with different time and length scales as it is typical for reaction kinetics systems. The third group is represented by moment equation based methods, where the statistical moments of the system response distributions are derived from the solutions of deterministic ordinary differential equations. However, only simple systems with few degrees of freedom have been studied as yet.

5.1 Linear propagation

Linear Sensitivity. For complex and high dimensional systems, the calculation of the numerous trajectories takes much computational effort. The problems intensifies as the overlap has to be calculated several times within the optimization procedure. As an alternative, one could consider only the linear approximation of the model variability propagation. This means one only considers the linear effects of a parameter perturbation. Therefore, only linear effects are incorporated in the linear overlap $\mathcal{F}_{\mathcal{L}}$.

In other words, one takes a look at the linear parameter sensitivity

$$\boldsymbol{\theta}_0 \mapsto \boldsymbol{\theta}_0 + \delta\boldsymbol{\theta}$$

on the model trajectory

$$\mathbf{y}(t) \mapsto \mathbf{y}(t) + \delta\mathbf{y}(t).$$

For the remainder of this section, let the parameters be normally distributed with $\pi \sim \mathcal{N}(\theta_0, \boldsymbol{\Delta\theta}^2)$. The linear propagation of the parameter perturbations $\delta\boldsymbol{\theta}$ that later assemble the model variability $\mathcal{M}_t$ can be described by the sensitivity matrix $\mathbf{S}$

$$\delta\mathbf{y}(t) = \mathbf{S}(t;\boldsymbol{\theta}_0)\,\delta\boldsymbol{\theta}, \tag{5.1}$$

which is the Jacobian of the flow with respect to the parameter $\boldsymbol{\theta}$ (taken componentwise)

$$\mathbf{S}(t;\boldsymbol{\theta}_0) = D\ \ \Phi^t\,\mathbf{y}_0|_{\ =\ 0} = \mathbf{J}(\boldsymbol{\theta}, t) \tag{5.2}$$

and fulfills the sensitivity equation for initial value problems (2.1)

$$\mathbf{S}'(t;\boldsymbol{\theta}_0) = \frac{\partial}{\partial y} f(\mathbf{y}(t), \boldsymbol{\theta}_0)\mathbf{S}(t;\boldsymbol{\theta}_0) + \frac{\partial}{\partial \boldsymbol{\theta}} f(\mathbf{y}(t), \boldsymbol{\theta}_0)|_{\ =\ 0}$$

with $\mathbf{S}(t_0, \boldsymbol{\theta}_0) = 0$ (c.f. [75]).

Since the initial parameter distribution is supposed to be normal and is propagated linearly, the gained model variability distribution is therefore normal also (c.f. [14]). Consequently, it suffices to propagate its mean and its standard deviation. In the implementation to be presented, the mean is exactly propagated by the trajectory $\Phi^t\,\mathbf{y}_0$, while the variance-covariance matrix of the model variability distribution is given by

$$\boldsymbol{\Sigma}_{\mathcal{M}}(\theta, \boldsymbol{\Delta\theta}, t) \ \ = \ \ \mathbf{J}(\boldsymbol{\theta}, t)\,\boldsymbol{\Delta\theta}^2\,\mathbf{J}(\boldsymbol{\theta}, t)^T. \tag{5.3}$$

The variance of the i^{th} dimension of the model variability at time t is the i^{th} diagonal entry of the variance-covariance matrix in (5.3) and is denoted by $\Sigma_i(\theta, \boldsymbol{\Delta\theta}^2, t)^2$. They are calculated by using (5.3) and (4.4)

$$\Sigma_i(\theta, \boldsymbol{\Delta\theta}, t)^2 = \sum_{j=1}^{p} \left(\frac{\partial}{\partial \theta_j} \left(\Phi_\theta^t y_0\right)_i \right)^2 \Delta\theta_j^2 \Bigg|_{\theta_j = \theta_{j0}}. \tag{5.4}$$

Consequently, the linear overlap functional at time t is given by

$$\begin{aligned} &\mathcal{F}_{\mathcal{L}}(\Phi_\theta^t y_0, \boldsymbol{\Sigma}_{\mathcal{M}}(\theta, \boldsymbol{\Delta\theta}, t), d(t), \sigma(t)) \ \ = \\ &\sum_{i=1}^{D} \sqrt{\frac{2\,\sigma_i(t)\,\Sigma_i(\theta, \boldsymbol{\Delta\theta}, t)}{\sigma_i(t)^2 + \Sigma_i(\theta, \boldsymbol{\Delta\theta}, t)^2}}\ \exp\left\{ -\frac{1}{2}\,\frac{\left(\Phi_\theta^t y_0 - d_i(t)\right)^2}{\sigma_i(t)^2 + \Sigma_i(\theta, \boldsymbol{\Delta\theta}, t)^2} \right\}. \end{aligned} \tag{5.5}$$

The linear overlap $\mathcal{F}_{\mathcal{L}}$ in (5.5) is calculated in each direction of the state space, since only there information about the data is available. Therefore, the off-diagonal entries of $\boldsymbol{\Sigma}_{\mathcal{M}}$ are not considered.

Algorithmic Realization. In order to apply commonly used optimization methods, the original problem

$$(\boldsymbol{\theta}_{\mathcal{L}}, \Delta\boldsymbol{\theta}_{\mathcal{L}}) = \underset{(\ ^*, \Delta\ ^*)}{\arg\max} \left(\mathcal{F}_{\mathcal{L}}(\boldsymbol{\theta}^*, \boldsymbol{\Sigma}_{\mathcal{M}}(\boldsymbol{\theta}^*, \Delta\boldsymbol{\theta}^*, t), \mathbf{d}(t), \boldsymbol{\Sigma}(t))\right). \tag{5.6}$$

is reformulated into a minimization problem

$$(\boldsymbol{\theta}_{\mathcal{L}}, \Delta\boldsymbol{\theta}_{\mathcal{L}}) = \underset{(\ ^*, \Delta\ ^*)}{\arg\min} \left(1 - \mathcal{F}_{\mathcal{L}}(\boldsymbol{\theta}^*, \boldsymbol{\Sigma}_{\mathcal{M}}(\boldsymbol{\theta}^*, \Delta\boldsymbol{\theta}^*, t), \mathbf{d}(t), \boldsymbol{\Sigma}(t))\right), \tag{5.7}$$

which is possible as the overlap is normalized. From now on, the notation is slightly misused: $\boldsymbol{\Delta\theta}$ denotes the diagonal entries of $\boldsymbol{\Delta\theta}$, with $\boldsymbol{\Delta\theta} = \operatorname{diag}(\Delta\boldsymbol{\theta})$. In principle, one could adopt very different strategies to solve this minimization problem, f.e., stochastic techniques like simulated annealing or other approaches to global optimization. However, the costly evaluations of $\mathcal{F}_{\mathcal{L}}$ suggest the application of a Gauss-Newton-type minimization, more specifically a (damped) Gauss–Newton algorithm or Levenberg–Marquardt of section 3.1 with statistical dimension reduction (c.f. [90]). The numbers of model variability calculations are significantly lower than for the stochastic methods.

The general Gauss-Newton-algorithm solves a series of linearized problems

$$z_{\text{optinc}} = \underset{\text{inc}}{\arg\min} \|\mathbf{J}_{\mathcal{F}}(z) z_{\text{inc}} - \mathcal{F}(z)\|_2 \tag{5.8}$$

with z_{inc} being an update for $z_{\text{new}} = z_{\text{old}} + z_{\text{optinc}}$, $\mathbf{J}_{\mathcal{F}}$ the Jacobian an of target function $\mathcal{F}$ with respect to z (c.f. [75]).

In comparison to the standard application of (damped) Gauss-Newton strategies to residuum minimization like $\mathcal{F}_{\text{LS}}$ in (3.2) or $\mathcal{F}_{\text{NLS}}$ in (3.13), one faces a new challenge: Due to the dependency of the overlap on the variances $\boldsymbol{\Sigma}_{\mathcal{M}}(\boldsymbol{\theta}, \boldsymbol{\Delta\theta}, t)$ of the model, the parameter variabilities $\Delta\boldsymbol{\theta}$ are to be optimized simultaneously. Consequently, (a) the dimension of the optimization problem is doubled in contrast to residuum optimization for the same model, (b) statistical correlations between parameters and the associated numerical problems will be more pronounced (since one has to expect correlations between a parameter θ_i and its variance $\Delta\theta_i$), and (c) one will see that the numerical effort for the evaluation of the Jacobian for each Gauss-Newton step increases quadratically.

For the linear problem in (5.8), one can analytically calculated the Jacobian $\mathbf{J}_{\mathcal{L}}$ with the notation of (2.1) and write it in a compact way

$$\begin{aligned} \mathbf{J}_{\mathcal{F}_{\mathcal{L}}} &= (\mathbf{J}_1, \mathbf{J}_2) \\ \mathbf{J}_1 &= \left[\frac{\partial \mathcal{F}_{\mathcal{L}}}{\partial \Phi^t \mathbf{y}_0} \cdot \frac{\partial \Phi^t \mathbf{y}_0}{\partial \boldsymbol{\theta}} + \frac{\partial \mathcal{F}_{\mathcal{L}}}{\partial \Sigma(\boldsymbol{\theta}, \Delta\boldsymbol{\theta}, t)} \cdot \frac{\partial \Sigma(\boldsymbol{\theta}, \Delta\boldsymbol{\theta}, t)}{\partial \theta} \right]_{(\ ,\Delta\ ,t)} \\ \mathbf{J}_2 &= \left[\frac{\partial \mathcal{F}_{\mathcal{L}}}{\partial \Sigma(\theta, \Delta\theta, t)} \cdot \frac{\partial \Sigma(\theta, \Delta\theta, t)}{\partial \Delta\boldsymbol{\theta}} \right]_{(\ ,\Delta\ ,t)} . \end{aligned} \tag{5.9}$$

One observes that this requires the evaluation of *second* derivatives of the flow, since from (5.4) one gets that,

$$\frac{\partial \boldsymbol{\Sigma}(\boldsymbol{\theta}, \Delta\boldsymbol{\theta}, t)}{\partial \theta} = \frac{1}{\Sigma(\boldsymbol{\theta}, \Delta\boldsymbol{\theta}, t)} \sum_{j=1}^{p} \left(\frac{\partial}{\partial \theta_j} (\Phi^t \mathbf{y}_0)_i \right) \left(\frac{\partial^2}{\partial \theta_j^2} (\Phi_\theta^t y_0)_i \right) \Delta\theta_j^2.$$

This obviously will not happen if one has to deal with the residuum functional $\mathcal{F}_{\mathcal{R}}$ instead of $\mathcal{F}_{\mathcal{L}}$. The computational effort for evaluation of the Jacobian will increase like p^2 (with $\theta \in \mathbb{R}^p$) for $\mathcal{F}_{\mathcal{L}}$ instead like p for the residuum $\mathcal{F}_{\mathcal{R}}$. The numerical evaluation of the derivatives involved is realized by numerical differentiation as it has been implemented within PRESTO[TM1].

Software package. PRESTO is a professional software tool used within research and development by many of the leading chemical and pharmaceutical companies internationally. This software package focusses on the modelling and dynamic simulation of arbitrary kinetic reactions. It provides general reaction step patterns for reaction kinetics and biokinetics as well as possibilities for the input of arbitrary ODE-systems. Therefore, PRESTO is the software of choice for investigations of the kind presented herein. It contains a quite general Gauss-Newton framework for parameter estimation for dynamical systems with damping strategy, convergence monitor, and update strategy as given in [75, 118]. This framework has been extended to implement and test the stochastically damped Gauss-Newton approach to overlap optimization as presented in the following.

In order to determine the initial values $\boldsymbol{\theta}_0$ and $\Delta\boldsymbol{\theta}_0$ of the parameters and their variances for the Gauss-Newton iteration, one may, for example, use box search. In the following it is assumed that there is a unique (local) maximum of $\mathcal{F}_{\mathcal{L}}$ in the vicinity of these initial values. With this preparations, the stochastically damped Gauss-Newton-algorithm consists of the following steps:

i) Initially set $k = 0$.

[1] PRESTO is a registered trademark by CiT GmbH Rastede

ii) Compute $\mathcal{F}_{\mathcal{L}}(\boldsymbol{\theta}^{(k)}, \Sigma(\boldsymbol{\theta}^{(k)}, \Delta\boldsymbol{\theta}^{(k)}, t), d(t), \boldsymbol{\sigma}(t))$. Compute a set of realizations of the Jacobian $\mathbf{J} = \mathbf{J}_{\mathcal{F}_{\mathcal{L}}}$ at $(\boldsymbol{\theta}^{(k)}, \Delta\boldsymbol{\theta}^{(k)})$ by numerical differentiation.

iii) Conduct dimension reduction by means of a truncated singular value decomposition (c.f. [90]).

iv) Compute the increment $(z, \Delta z)$ to $(\boldsymbol{\theta}, \Delta\boldsymbol{\theta})$ by solving

$$\mathbf{J}(\boldsymbol{\theta}^{(k)}, \Sigma(\boldsymbol{\theta}^{(k)}, \Delta\boldsymbol{\theta}^{(k)}, t))(z, \Delta z)^T = \mathcal{F}_{\mathcal{L}}(\boldsymbol{\theta}^{(k)}, \Sigma(\boldsymbol{\theta}^{(k)}, \Delta\boldsymbol{\theta}^{(k)}, t), \mathbf{d}(t), \boldsymbol{\sigma}(t))$$

in the sense of (5.8) while possibly incorporating the dimension reduction.

v) Set

$$(\boldsymbol{\theta}^{(k+1)}, \Delta\boldsymbol{\theta}^{(k+1)}) = (\theta^{(k)}, \Delta\boldsymbol{\theta}^{(k)}) + \kappa\,(z, \Delta z)$$

with damping parameter κ. Verify monotony as reported in [118].

vi) Test convergence by means of the stopping criteria given in [74]. If not converged, set $k = k + 1$ and iterate from ii) onwards.

Overlap optimization. As mentioned earlier on page 61, one is interested in small model mean deviations. Therefore, to improve the convergence of the optimization, one can extend the functional to

$$\mathcal{F}_{\text{ext}} = 1 - \mathcal{F}_{\mathcal{L}} + \alpha\mathcal{F}_{\text{NLS}}. \tag{5.10}$$

The choice of α depends on the user himself. It is sensible to gradually decrease α within the optimization scheme, so that the influence of $\mathcal{F}_{\mathcal{L}}$ dominates and $\mathcal{F}_{\text{NLS}}$ vanishes. This algorithmic scheme is also implemented.

Condition Jacobian. Before turning to the next section, a last remark on the condition of the Jacobian with the adapted Gauss–Newton in (5.9) is necessary. As the hyperparamters to optimize, namely $\boldsymbol{\theta}$ and $\boldsymbol{\Delta\theta}^2$ do belong to certain parameter distributions, it is likely that they are also correlated to some extend. Therefore as mentioned before, the Gauss–Newton step within the optimization problem becomes ill–conditioned. For some examples, it is possible to use some truncated Gauss–Newton–method (c.f. [90, 213, 242]) to avoid an ill-conditioned problem.

By means of a singular value decomposition (c.f. [103]), one calculates the spectrum of the Jacobian of (5.9). Depending on a user given threshold, one can

eliminate some not essential direction for the optimization. If such an dimension reduction is possible, concentrating on the essential improves the convergence significantly (c.f. [227]).

However, not for all applications, it is possible to find a spectral gap and therefore separate the hyperparameters into essential and non–essential ones. The following two figure[2] document the mentioned situation. In the left one, a spectrum taken from an optimization for a polymer reaction, one can see such a gap and find a sensible threshold. The right figure shows a biokinetic reaction, where detecting a gap is not possible.

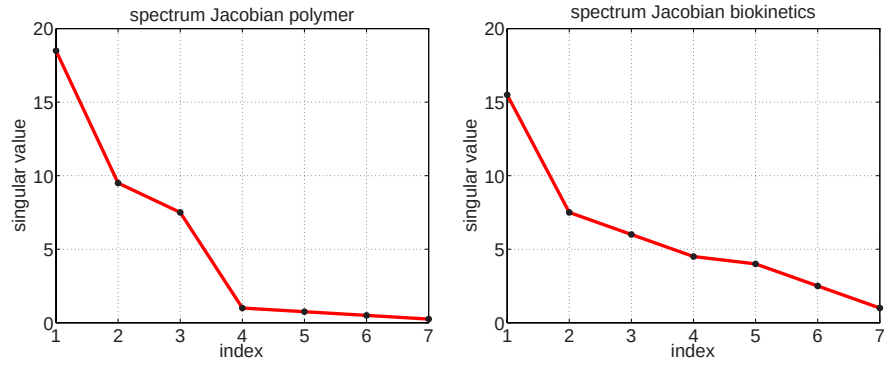

Figure 5.1: Spectrum of Jacobian

The separation between essential and non–essential hyperparameters is not static. This means, after each iteration step, different hyperparameters can be identified as essential or non–essential.

5.2 Sampling and propagation

Sampling. As an analytical solution for the model variability $\mathcal{M}_t$ in (4.2) is virtually impossible to find, one has to numerically calculate it. A very canonical approach is to sample the parameter distribution, solve the ordinary differential equation (2.1) for each draw of the parameter ensemble individually and reassemble the density for $\mathcal{M}_t$.

There are several ways to generate the ensemble of parameters. Very commonly, the sample is drawn randomly from the distribution. Several sophisticated random sampling methods have been developed over the last years (c.f. [156]) and

[2] A special thank to Elmar Diederichs for providing the figures in [211].

are available in numerous standard software packages.

For calculating the model variability one is only interested in the approximation quality of the distribution to perform the integration (4.1) and not in statistical properties of the sample itself. Therefore, one could use more suitable sampling methods than the Monte Carlo methods that only converges with $N^{1/2}$. Two alternatives are pseudo– and quasi–random numbers. The latter is of interest for the overlap setting.

Quasi–random numbers. Quasi–random numbers are a sequence of constructed numbers that uniformly cover a volume. The three most commonly used algorithms to construct this sequence were introduced by FAURE in [86], by NIEDERREITER in [185] and by SOBOL in [218]. The "uniformly distributed" sequence constructed there is then transformed into the desired parameter distribution. As a thumb rule literature suggest to use the quasi–random number for problems with less than 15 to 20 dimensions. Beyond that, the "normal" random numbers shall be used.

Unlike in the previous section, the propagation method does not favor a specific optimization algorithm. In many applications, some simplex optimization method can be employ, despite knowing about the convergence problems (c.f. [148]). Therefore, it is suggested that before starting an optimization by some simplex method, a linear optimization of section 5.1 should be conducted prior to it in order to gain reasonable starting values.

Before turning to the model variability propagation by the TRAIL algorithm, the advantages and disadvantages of using the sampling method as an "exact" method shall be discussed.

One advantage of using parameter sampling is certainly the flexibility of the initial distribution's type. In comparison to the previously employed methods, one can abandon the normal parameter distribution and choose arbitrary ones. In many cases, also for biokinetics, a parameter is associated with a certain interpretation that only allows positive values. This can happen if a parameter is for example interpreted as a reaction rate. Therefore, it sensible for use a distribution with strict positive definition values like a log–normal distribution.

Another natural advantage is that during the propagation all the effects of all orders are considered and not only the ones of first and second order when using

$\mathcal{F}_{\mathcal{L}}$. Nevertheless, in order to reduce the computations effort when sampling the parameter distribution, one has to use an efficient way to do so.

Optimization. At this point, the optimization is revisited. All sampling simulation are programmed in MATLAB^{TM} are performed with quasi–random numbers kindly contributed by ELMAR DIEDERICHS. Further, the following illustrations and calculations are done for a log–normal parameter distribution (c.f. [83]). The density is given by

$$f(x) = \frac{1}{xs\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{\log x - m}{s}\right)^2\right\}, \tag{5.11}$$

where s and m are referred to as the hyperparameters of the distribution. Besides the positive state space of the distribution, it also possesses higher momentums, namely the expectation μ, standard deviation σ, skewness η_1 and kurtosis η_2:

$$\begin{aligned}
\mu &= \exp\left\{\frac{2m + s^2}{2}\right\} && (5.12)\\
\sigma &= \sqrt{\exp\{2m + 2s^2\} - \exp\{2m + s^2\}} && (5.13)\\
\eta_1 &= \left(\exp s^2 + 2\right) \cdot \sqrt{\exp s^2 - 1} && (5.14)\\
\eta_2 &= \exp 4s^2 + 2\exp 3s^2 + 3\exp 2s^2 - 3. && (5.15)
\end{aligned}$$

Therefore through a choice of the hyperparameters s and m,

$$\begin{aligned}
m &= \log\left\{\frac{\mu^2}{\sqrt{\sigma^2 + \mu^2}}\right\} && (5.16)\\
s &= \sqrt{\log\left\{\left(\frac{\sigma}{\mu}\right)^2 + 1\right\}} && (5.17)
\end{aligned}$$

the parameter distribution can take an almost symmetric or asymmetric form as seen in figure 5.2. Unlike in (5.7), the formulation of the optimization problem into a linear overlap optimization problem is not possible. Among others, one cannot rely on the invariance of the model variability distribution family. Only for the special case of linear transformation, normally distributed variables remain normal. Therefore, despite the convergence challenges, a simplex optimizer was used (c.f. [148]).

The following simulations were done for the Monod kinetics (6.9), but for log–normally distributed parameters. In the figure 5.3, the vertices of the simplex

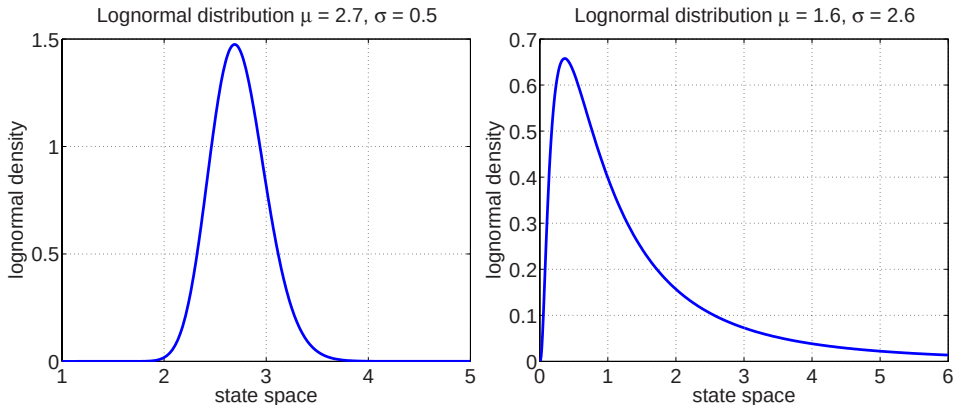

Figure 5.2: Quasi-symmetric and asymmetric shapes of the log–normal distribution

during the last 200 iterations steps of a simplex optimization are plotted. Using a sloppy language, one could say that they show the ”path of the simplex”. Through the projection one sees that the surface of the overlap functional is some sort of cliffy (see figure 5.3).

This is supported by the plot in figure 5.4. There a very time consuming boxplot search for the overlap functional over the entire parameter space was performed. In the upper two plots, one can recognize some sort of frontier or a clear funnel, respectively, towards the optimal hyperparameters. However, the two lower plots indicate that the hyperparameter take on equally optimal values for several secluded values. This indicates the possibility of having several local suboptimal for the hyperparameters.

To avoid such convergence problems due to complex functional surface, the optimization of the functional $\mathcal{F}_{\mathcal{O}}$ can be preceded by a linear overlap optimization which results serve as a starting points for the $\mathcal{F}_{\mathcal{O}}$ optimization. Further, optimization results should be clearly seen as local and not global results.

Discretization. Within the simulation one choose the appropriate sample size for the parameter distribution. The larger, the better the approximation quality is going to be. The smaller, the faster simulation is going to become, since unnecessary calculations are avoided. The plot in figure 5.5 shows the connection between the approximation quality and the sample size. The number that is actually employed has to be chosen individually.

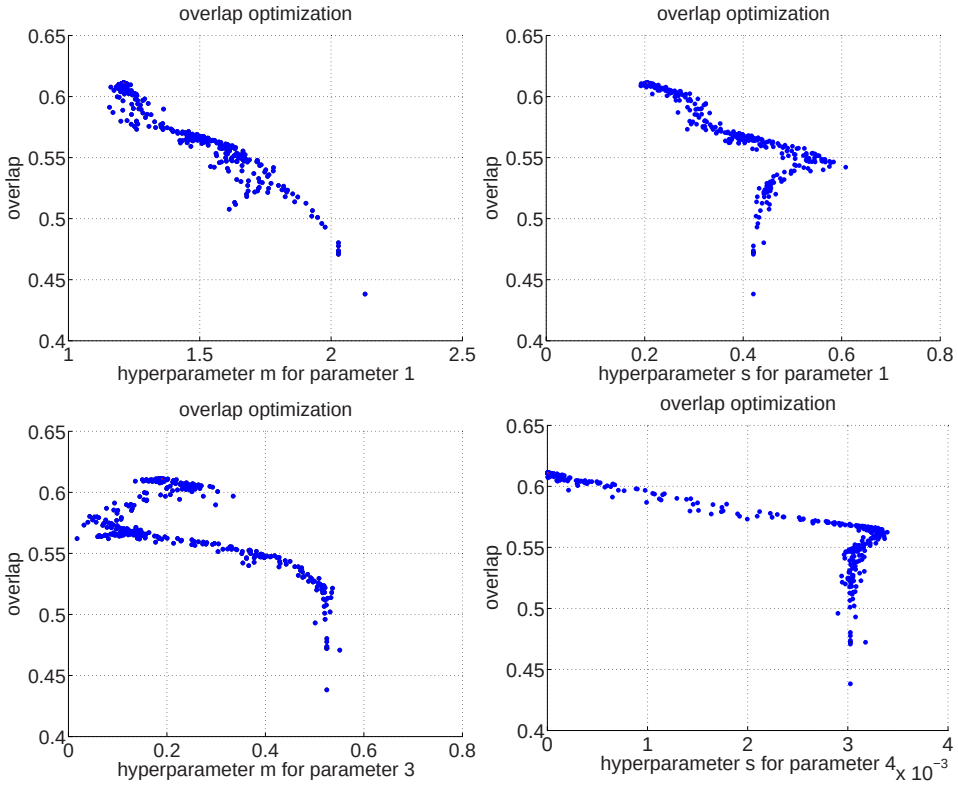

Figure 5.3: Simplex vertices within the optimization

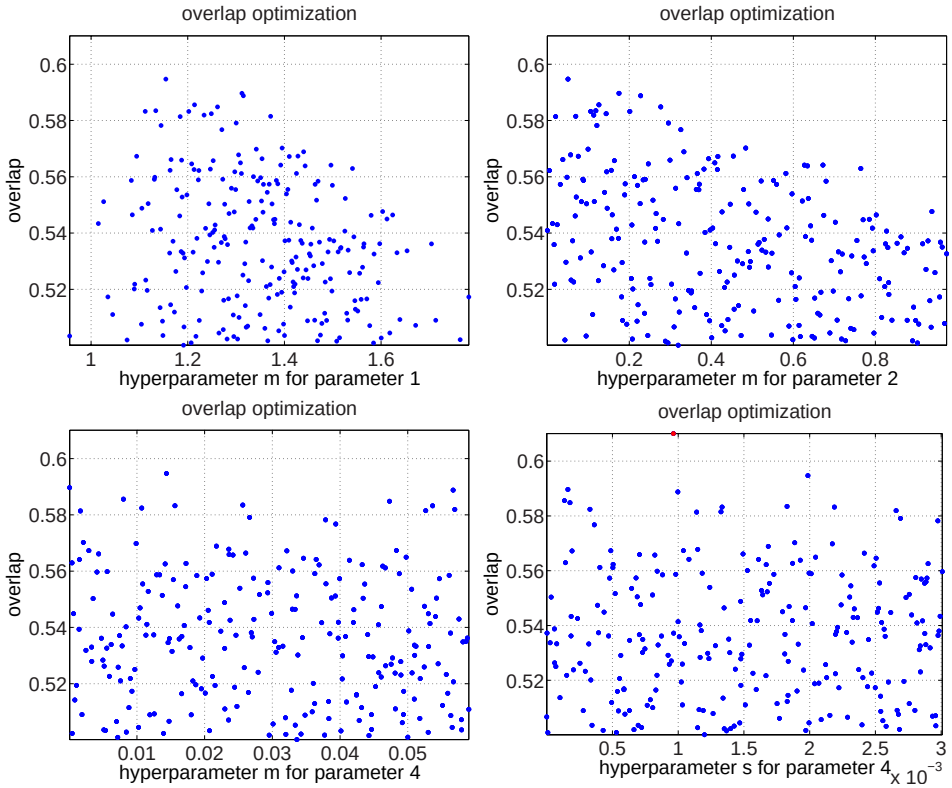

Figure 5.4: Sample points boxplot search

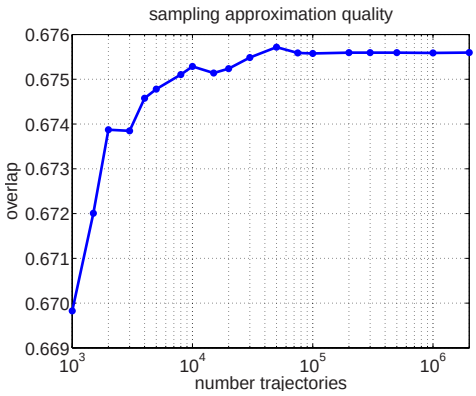

Figure 5.5: Approximation quality of the overlap integral

5.3 Nonlinear propagation

This sections presents a theoretical framework and adaptive numerical realization of deterministic ordinary differential equations with stationary parameter distributions consisting out of the linear propagation with a non–linear correction scheme. The approach is based on the reformulation of the problem (2.1) in the context of the FOKKER–PLANCK theory for the evolution of multidimensional distribution functions as needed to describe the model variability. In order to solve the resulting partial differential equation (PDE) numerically, the adaptive Gaussian-based particle method designed in the context of molecular dynamics (c.f. [119, 121, 122, 123]) is modifies and applied. The nonlinear sensitivity problem follows two consecutive steps: A linear prediction of the evolving density function being represented by an ensemble of Gaussians is consequently corrected by the controlled approximation of nonlinear effects within the adaptive solution of the linear regression problem. The performance of the method is illustrated with the parameter sensitivity analysis of enzyme–substrate reaction kinetics and compared with alternative propagation results.

This adapted algorithm can be regarded as the further development of the frozen Gaussian technique that was introduced by HELLER and others in numerous applications (c.f. [11, 50, 80, 111, 142, 150, 166, 167, 172, 209, 241]).

Theoretical setting

Fokker-Planck equation. Abandoning the particle in favor of the ensemble presentation results in considering a distribution $u(x,t)$, symbolizing the bundle of trajectories over time t, instead of a single trajectory $\Phi^t x_0$. The density u symbolizes the time dependant model variability distribution with $\mathcal{M}_t \sim u(.,t)$. This translation is a very common approach in statistical physics, especially in molecular dynamics (c.f. [29, 69, 210, 212]).

The time propagation of an initial density $u(x, t = 0)$ can be described by the general FOKKER-PLANCK or LIOUVILLE-equation (c.f. [97, 132, 200])

$$\begin{aligned} \frac{\partial}{\partial t} u(x,t) &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left(f_i(x,t)\ u(x,t) \right) \\ &= -\sum_{i=1}^{d} \left(\frac{\partial}{\partial x_i} f_i(x,t)\, u(x,t) + f_i(x,t) \frac{\partial}{\partial x_i} u(x,t) \right). \end{aligned} \tag{5.18}$$

Challenge. Whereas solving ordinary differential equations like (2.1) numerically for real world application is well established (c.f. [75]), solving the partial differential equations like the FOKKER-PLANCK-equation (5.18), especially in the context of molecular dynamics, still is a challenge. The curse of dimension, f.e., results in exponentially growing computational costs for many traditional grid discretization techniques.

As a consequence, for each class of applications, an individual procedure, employing the structure of the problem, has to be employed. For the density propagation of Liouville type problems, HORENKO and WEISER have developed the TRAIL algorithm (**T**rapezoid **R**ule for **A**daptive **I**ntegration of **L**iouville dynamics) as it is shown in f.e. in [119, 121, 122, 123]. The next sections describes the concept of the TRAIL algorithm within the framework of FOKKER-PLANCK-type transport problems.

The TRAIL-algorithm

The TRAIL-algorithm is a multidimensional, fully adaptive particle method, using a superposition of Gaussians, to describe the propagation of sensitivities in nonlinear dynamical systems. The adaptive discretization scheme employed is based on the adaptive Rothe method (c.f. [36, 76, 204]) followed by an adaptive method of line approach for solving the deduced but locally linearized spatial problem.

The (in most cases nonlinear) structure of the distribution in question $u(x,t)$ is dissolved by a superposition of Gaussians. Locally at each time propagation step $t \to t + \Delta\tau$, a linearized FOKKER-PLANCK-transport problem is solved. To maintain a suifficient approximation quality of the solution, however, the locally linearly propagated Gaussians ought to be corrected or their spatial discretization adapted, respectively. As a result, the algorithm's strategy can be summarized in two steps: (1) linear prediction and (2) correction and adaptivity control.

Linear prediction

Since in all linearized transport problems, as considered in the linear prediction step, Gaussians remain Gaussians, one merely needs to describe the propagation of a single Gaussian (as an ansatz function)

$$\begin{aligned} u(x,t) &= A(t) \cdot \exp\left\{ (x - x_0(t))^T \, \mathbf{G}(t) \, (x - x_0(t)) \right\} \\ &= A(t) \cdot \exp T(x,t), \end{aligned} \tag{5.19}$$

with $A(t)$ being the amplitude, $x_0(t)$ the center of the Gaussian, $\mathbf{G}(t)$ the shape matrix, and $T(x,t)$ the abbreviation for

$$T(x,t) = (x - x_0(t))^T \, \mathbf{G}(t) \, (x - x_0(t)) \,. \tag{5.20}$$

Shape matrix. For biokinetic and pharmacokinetic applications, correlations between the state space coordinates can be observed. In order to apply the TRAIL algorithm efficiently there, one has to extend the existing concept by propagating the shape matrix $\mathbf{G}(t)$ of the Gaussian in (5.19), too. The partial spatial and temporal derivatives

$$\begin{aligned} \frac{\partial}{\partial t} u(x,t) &= \left(\frac{\partial}{\partial t} A(t) \right) \cdot \exp T(x,t) + A(t) \cdot \exp T(x,t) \cdot \left(\frac{\partial}{\partial t} T(x,t) \right) \\ \frac{\partial}{\partial x_i} u(x,t) &= A(t) \cdot \exp T(x,t) \cdot \left(\frac{\partial}{\partial x_i} T(x,t) \right) \end{aligned}$$

of the Gaussian are assembled in (5.18) and devided by $\exp T(x,t)$

$$\begin{aligned} \frac{\partial}{\partial t} A(t) + A(t) \cdot \frac{\partial}{\partial t} T(x,t) &= \\ -A(t) \sum_{i=1}^{d} \left(\frac{\partial}{\partial x_i} f_i(x,t) + f_i(x,t) \frac{\partial}{\partial x_i} T(x,t) \right) . \end{aligned} \tag{5.21}$$

Separating the amplitude and shape matrix. According to (5.20), $T(x,t)$ does depend on x quadratically. The same applies for its partial spatial and temporal derivatives. Therefore, one can separate the spatial and non-spatial terms of (5.21) and gets an differential equation for the amplitude $A(t)$ and the spatial coordinates

$$\frac{\partial}{\partial t} A(t) = -A(t) \sum_{i=1}^{d} \frac{\partial}{\partial x_i} f_i(x,t) \tag{5.22}$$

$$\frac{\partial}{\partial t} T(x,t) = -\sum_{i=1}^{d} \left(f_i(x,t) \frac{\partial}{\partial x_i} T(x,t) \right) . \tag{5.23}$$

Linearization. A linearization of the original model system's right side in (2.1) around the Gaussian's center $x_0(t)$

$$\begin{aligned} f_i(x,t) &\approx f_i(x_0,t) + \nabla_x f_i(x,t)|_{x=x_0} \, (x - x_0(t)) \\ f(x,t) &\approx f(x_0,t) + \mathbf{J}_x (x - x_0(t)), \end{aligned}$$

with $\mathbf{J}_x$ being the Jacobian of the RHS, approximates (5.23) by

$$\frac{\partial}{\partial t} T(x,t) = \\ -\sum_{i=1}^{d} \left(f_i(x_0,t) + \nabla_x f_i(x,t)|_{x=x_0} (x - x_0(t)) \right) \frac{\partial}{\partial x_i} T(x,t),$$

which results in

$$(x - x_0(t))^T \dot{\mathbf{G}}(t) (x - x_0(t)) = \\ -(x_0 - x)^T \mathbf{J}_x \mathbf{G}(t) (x - x_0(t)) - (x - x_0(t))^T \mathbf{G}(t) \mathbf{J}_x^T (x - x_0(t))$$

and finally in the differential equation for the shape matrix $\mathbf{G}$

$$\dot{\mathbf{G}}(t) = -\mathbf{J}_x \mathbf{G}(t) - \mathbf{G}(t) \mathbf{J}_x^T. \tag{5.24}$$

Shape matrix symmetry. All shape matrices of Gaussians are symmetric, so does the initial one $\mathbf{G}(0)$. Since the locally linearily propagated distribution stay Gaussian, its shape matrix $\mathbf{G}(t)$ ought to remain symmetric: $\mathbf{G}(t) = \mathbf{G}(t)^T$, too. Demanding this property, the ordinary differential equation (5.24) for $\mathbf{G}(t)$ also shows that symmetry

$$\begin{aligned} \dot{\mathbf{G}}(t) &= -\mathbf{J}_x \mathbf{G}(t) - \left(\mathbf{J}_x \mathbf{G}(t)^T \right)^T \\ &= -\mathbf{J}_x \mathbf{G}(t) - \left(\mathbf{J}_x \mathbf{G}(t) \right)^T. \end{aligned} \tag{5.25}$$

That final equation (5.25) has the analytical solution (c.f. [10], [16])

$$\mathbf{G}(t) = \exp(-\mathbf{J}_x t)^T \cdot \mathbf{G}(0) \cdot \exp(-\mathbf{J}_x t). \tag{5.26}$$

Final system. To implement the TRAIL algorithm for Gaussians with linearily propagated shape matrix, one needs to implement the three equations (2.1), (5.22) as well as (5.26) for as propagation equations for the Gaussians

$$\dot{x}(t) = f(x,t), \tag{5.27}$$

$$\dot{A}(t) = -A(t) \sum_{i=1}^{d} \frac{\partial}{\partial x_i} f_i(x,t) \quad \text{and} \tag{5.28}$$

$$\mathbf{G}(t) = \exp(-\mathbf{J}_x t)^T \cdot \mathbf{G}(0) \cdot \exp(-\mathbf{J}_x t) \tag{5.29}$$

considered in the predictor step.

Nonlinear correction and adaptivity control

In order to meet a global approximation tolerance, the ensemble of Gaussians might have to be constantly enhanced during the propagation process. Even though, the linear transport problem itself can be solve exactly, the finite ensemble of Gaussians, however, is an approximation for the true density only. This especially applies for the initial one at $t = 0$, whose approximation errors are propagated as well. In either way, by enhancing the approximation quality one is embedding the linear problems into a nonlinear propagation scheme.

Two strategies for enhancing the approximation quality of the Gaussians are employed: correcting the Gaussians themselves or adapting their spatial discretization. For both cases, the local approximation error of the integrator used (implicit trapazoid rule) is used as an quality indicator.

Correction. The Gaussians' amplitudes are fitted to minimize the local approximation error of the integrator used (implicit trapezoid rule). Since the amplitudes are fitted only, the optimization problem is a linear least square one. The result of the predictor step is used as a starting value for the optimizer.

Spatial adaptivity. Since the shape structure of the approximated density might change over time as well, the spatial discretization has to be controlled and adapted, too.

Adapting the spatial discretization means either to generate (*spawn*) new Gaussians, when the approximation quality is not met or the distribution starts show a more complicated structure, or to drop (*prune*) one when two Gaussians are to close together or the distribution starts to show a more simpler structure than originally.

Spawning case: The local approximation error estimator of the employed implicit trapezoid rule spots the spatial coordinates in whose vicinity new Gaussians are to be created by some Monte Carlo sampling.

Pruning case: Doing an integration step by means of the implicit trapezoid rule numerically means to solve a least square problem in order to adapt the amplitude and the center of each Gaussian. The observation of two Gaussians being to close together translates into the linear least square problem become ill-conditioned. An analysis of the problem's subcondition spots the particles that ought to be removed.

Time step size. The estimation error from the trapazoid rule is also used to determine the time step for the Rothe method, dictating the time discretization. The local approximation error for the time integration of the integrator used (implicit trapezoid rule) along with the predefined user accuracy translates into a temporal step size $\Delta\tau$. That semi-discretization produces stationary PDEs, which is solved as described above: by a method of lines approach with adaptive spatial discretization using Gaussians as ansatz functions. By that, the spatial discretization error can be matched with the temporal one.

In the further course of the article, the extended TRAIL algorithm shall be applied and compared with other numerical methods for the class of Michaelis-Menten-Kinetics. Before that, it is necessary to describe the mathematical system behind the kinetic to apply.

Parameter sensitivity analysis

As mentioned before, in many applications, one is interested in the impact of parameter variability. This curiosity can easily be integrated in the existing framework. By considering parameters as "normal" constant variables, the model in equation (2.1) can be extended to an $(d+p)-$dimensional problem

$$\dot{x} = f(x,\theta,t), \quad x(0) = x_0 \tag{5.30}$$

$$\dot{\theta} = 0, \quad \theta(0) = \theta \tag{5.31}$$

The density to propagate u is also extended to an $(d+p)-$dimensional one: $u(x,\theta,t)$. For reasons of convenience, however, the notation itself is not changed and the temporal dependencies are omitted.

The derivation of the equations (5.27), (5.28) and (5.29) is not effected by including the parameters. Due to the structure of the extended problem, the shape matrix shows the structure. For reasons of better readability, the time dependencies are omitted. Let $\mathbf{J}_x$ and $\mathbf{J}_\theta$ be the Jacobians of f in (5.30) with respect to the spatial coordinates x and the parameters θ, then the Jacobian of the extended problem has the structure

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_x & \mathbf{J}_\theta \\ 0 & 0 \end{pmatrix}. \tag{5.32}$$

The joint density's general shape matrix $u(x,\theta,t)$ of state and parameter space has the structure

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_x & \mathbf{G}_m \\ \mathbf{G}_m^T & \mathbf{G}_\theta \end{pmatrix}. \tag{5.33}$$

$\mathbf{G}_x$ or $\mathbf{G}_\theta$ describe the statistic interdependencies within the space or parameter coordinates, respectively; the sub-matrix $\mathbf{G}_m$ the ones between the parameter and state space. Then calculating $\dot{\mathbf{G}}$ by (5.25) results in

$$\dot{\mathbf{G}} = -\begin{pmatrix} \mathbf{G}_1 & \mathbf{G}_2 \\ \mathbf{G}_2^T & 0 \end{pmatrix}, \tag{5.34}$$

while using the notation

$$\mathbf{G}_1 = \mathbf{J}_x\mathbf{G}_x + \mathbf{J}_\theta\mathbf{G}_m^T + \left(\mathbf{J}_x\mathbf{G}_x + \mathbf{J}_\theta\mathbf{G}_m^T\right)^T \tag{5.35}$$

$$\mathbf{G}_2 = \mathbf{J}_x\mathbf{G}_m + \mathbf{J}_\theta\mathbf{G}_\theta \tag{5.36}$$

or similar to the analytical solution of equation (5.26) in

$$\mathbf{G}(t) = \exp\left\{-\begin{pmatrix} \mathbf{J}_x & \mathbf{J}_\theta \\ 0 & 0 \end{pmatrix} t\right\} \cdot \begin{pmatrix} \mathbf{G}_x(0) & 0 \\ 0 & \mathbf{G}_\theta \end{pmatrix} \cdot \exp\left\{-\begin{pmatrix} \mathbf{J}_x & \mathbf{J}_\theta \\ 0 & 0 \end{pmatrix} t\right\}^T.$$

In order to reduce the computational effort in higher dimensional cases, one should make benefit out of the structure of the shape matrix as well as of the rest of the problem.