Automated parameterization of intermolecular pair potentials using global optimization techniques

Andreas Krämer[a,b], Marco Hülsmann[b], Thorsten Köddermann[b], Dirk Reith[a,b]

[a] *Department of Electrical Engineering, Mechanical Engineering and Technical Journalism, Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany*

[b] *Department of Simulation Engineering, Fraunhofer Institute for Algorithms and Scientific Computing, Schloß Birlinghoven, 53757 Sankt Augustin, Germany*

Abstract

In this work, different global optimization techniques are assessed for the automated development of molecular force fields, as used in molecular dynamics and Monte Carlo simulations. The quest of finding suitable force field parameters is treated as a mathematical minimization problem. Intricate problem characteristics such as extremely costly and even abortive simulations, noisy simulation results, and especially multiple local minima naturally lead to the use of sophisticated global optimization algorithms. Five diverse algorithms (pure random search, recursive random search, CMA-ES, differential evolution, and taboo search) are compared to our own tailor-made solution named CoSMoS. CoSMoS is an automated workflow. It models the parameters' influence on the simulation observables to detect a globally optimal set of parameters. It is shown how and why this approach is superior to other algorithms. Applied to suitable test functions and simulations for phosgene, CoSMoS effectively reduces the number of required simulations and real time for the optimization task.

Keywords: optimization, molecular simulation, molecular modeling, automation, force field parameters

1. Introduction

Nowadays, molecular dynamics and Monte-Carlo simulations are indispensable in various areas, including thermodynamic properties of fluids [1], transport processes in liquids [2], protein folding [3], polymer properties [4] or pharmaceutics [5]. They are likely to become even more important due to the rapid development and affordability of powerful computers.

Email addresses: andreas.kraemer@h-brs.de (Andreas Krämer), marco.huelsmann@scai.fraunhofer.de (Marco Hülsmann), thorsten.koeddermann@scai.fraunhofer.de (Thorsten Köddermann), dirk.reith@h-brs.de (Dirk Reith)

As possibilities open up, there is a growing need for accurate molecular models that are tailored for specific applications with quantitatively matching capabilities. The parameterization of the models, which are given as force field equations, is the most critical part of the modeling process. While *intramolecular* parameters can be derived from quantum mechanics, *intermolecular* parameters, e.g. the Lennard-Jones parameters, have to be adjusted empirically. In practice, this means that the latter have to be tuned in order to reproduce macroscopic physical observables, such as density, diffusion coefficients, viscosity, vapor pressure and heat of vaporization. This so-called calibration is the principal bottleneck in the modeling process.
The first reason for the high cost of the calibration is the lengthy computation time required for a single simulation. Simulations have to be iteratively repeated with changing parameter settings in order to minimize the loss between simulated and experimental observables. Second, the simulation observables are calculated as statistical averages and hence they are noisy. Third, simulations can terminate without any useful result, for instance when the simulation system becomes unstable for a certain parameter combination. Fourth, the objective loss functional may have plenty of local optima which is unfavorable for the discovery of a global optimum as well.
Consequently, the calibration requires sophisticated optimization algorithms that are capable of detecting a globally optimal set of intermolecular force field parameters automatically and within an acceptable amount of time. Hence, they have to scale well on multi-core computers, be robust with respect to noise as well as abortive function evaluations and prevent preliminary convergence to local optima. Each of the above criteria is indispensable for a generic force field optimization workflow.
Although several groups have worked on developing accurate force fields, the authors do not know any approach that meets all of the criteria. The most significant contributions are briefly highlighted in the next section.

1.1. Related work

The Nelder-Mead simplex method [6] was one of the first optimization algorithms used for automated force field design [7, 8]. It is robust with respect to noise and derivative-free but does not guarantee convergence, not even to a local optimum. Up to a hundred sequential function evaluations were needed to solve calibration problems for hydrocarbons at a single temperature. The obtained parameters failed to produce similarly good results for other temperatures [9].
Later, Ungerer et. al. [10, 11] used the gradient-based Gauß-Newton method, obtaining accurate force fields for small molecules. In the meantime, a similar method by Stoll [12] was successfully applied to other molecules such as cyclohexanol [13] and acetonitrile [14]. These methods are efficient in the immediate neighborhood of an optimum but they require a suitable initial guess for the objective parameters. Otherwise, they converge to a non-optimal solution.
The recently proposed gradient-based optimization workflow (GROW) by Hülsmann et. al. [15] has extended this approach by various descent methods like *steepest descent, conjugate gradients,* and *trust region*, allowing the initial parameters to

be situated farther away from the optimum. GROW has proven to be a useful and reliable tool in diverse applications: test functions [16], small molecules [17], and ionic liquids [18]. Nevertheless, suitable starting parameters are necessary for convergence. Parallel iterations are not possible for these gradient-based methods since iterations depend on each other.
Alternatively, metamodels were used for the optimization task. Metamodels are also called *response surface models* in the literature [19]. The terms are used interchangeably throughout the present paper, denoting multimodal interpolations or regressions with cheap evaluations. Maaß et. al. [20] have studied the global influence of force field parameters for ethylene oxide. Metamodels, based on 80 random parameter sets, were created and studied with the interactive tool DesParO [21]. A set of parameters was selected manually. Most recently, Hülsmann et. al. [22, 23] have shown that global optimization in connection with GROW is likely to be a generalizable workflow for optimizing intermolecular parameters entirely automatically from scratch (cf. Figure 1). The parameters provided by DesParO were not optimal, but the subsequent GROW optimization turned them into an excellent force field within 14 steepest descent iterations. However, the interactivity of the global metamodeling as well as the great number of overall function evaluations are unfavorable for a generic calibration tool, as the overall optimization process took weeks [22]. At that point, it became clear that the global part of the optimization process could be enhanced in terms of efficiency and automation. Consequently, in this work, research was focused on appropriate global optimization techniques.

1.2. Developing an appropriate global optimization strategy

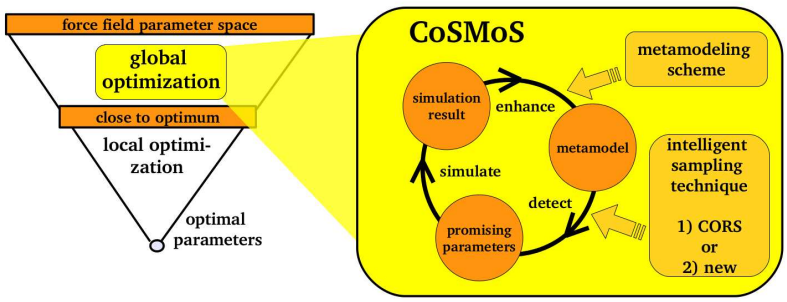

Figure 1: Desired optimization workflow to detect optimal force field parameters. The global CoSMoS optimization is based on metamodeling. The metamodel is utilized to detect a promising parameter vector. The simulation observables are calculated and integrated into the metamodel in order to enhance its accuracy.

Global optimization is concerned with two goals: *exploration* of the search space and *exploitation* of previous function evaluations. Various techniques have been developed, for instance simulated annealing [24, 25], evolutionary methods (including swarm optimization and genetic algorithms) [26–29], taboo

search [30, 31], multistart [32], direct search [33, 34], and finally metamodel-based optimization [19, 35–37]. The last are particularly suited for functions with computationally expensive evaluations as they involve the results of all previous evaluations [35].
Different types of global optimizers were tested by the authors with respect to the given calibration problem. In particular, six different algorithms were selected for comparison: pure random search (PRS), recursive random search (RRS) [38], the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [39, 40], differential evolution (DE) [27], a special taboo search (TS) algorithm [30], and Constrained Optimization using Response Surfaces (CORS) [19, 37]. Each of them represents a particular class of global optimizers.
In contrast to the random sampling favored by DesParO, modern metamodel-based optimizers, like CORS, rely on intelligent sampling techniques. In fact, the metamodel is exploited to concentrate sampling onto more interesting domains of the search space. Hence, the accuracy of the model grows especially in these domains and, furthermore, the sampling points are likely to approach an optimal set of parameters. Taken together, metamodeling and optimization complement and stimulate each other.
CoSMoS follows this approach as a global optimization tool for the **C**alibration **o**f molecular force fields by **S**imultaneous **M**odeling **o**f **S**imulated data (cf. Figure 1). Currently, three metamodeling schemes and two different intelligent sampling techniques are implemented in CoSMoS: The CORS sampling technique and a new method developed by the authors. Additional components of CoSMoS are a suitable normalization of the metamodels, different loss functionals for the optimization, a parallelization framework and a way to make use of abortive simulations. New features like other sampling techniques or metamodeling schemes can easily be integrated through the modular program structure. CoSMoS uses the GROW interface to molecular simulations. Taken together, it is tailored to meet all of the above-mentioned criteria and satisfy the need for an appropriate globally convergent force field calibration tool. It can be used as a pre-optimizer for gradient based optimization (cf. Figure 1) or as a stand-alone solution.
The numerical results on test functions and a case study on phosgene illustrate that metamodel-driven optimization is a viable concept to calibrate force field parameters. It is shown that the presented tool suits a wide range of calibration problems and that it saves simulations and real time compared with the other tested global optimization methods.

2. Optimization of force field parameters: General methodology

2.1. Notation

The following notation is used throughout the paper:

- n denotes the number of force field parameters,
- m denotes the number of observables, multiplied by the number of temperatures,

- $x \in \mathbb{R}^n$ denotes a vector of objective force field parameters,
- $f^{\text{sim}}(x) \in \mathbb{R}^m$ denotes a vector of simulated observables,
- $f^{\text{exp}}(x) \in \mathbb{R}^m$ denotes the vector of corresponding experimental observables, and
- $\mathcal{M}^{(\nu)}(x) \in \mathbb{R}^m$ denotes a metamodel for the ν-th observable, $\nu = 1, \ldots, m$.

2.2. The optimization task

The quest of finding suitable force field parameters is stated as a minimization problem:

$$\text{min!} \quad F(x),\ x \in D. \tag{1}$$

The search space $D \subset \mathbb{R}^n$ is defined by box constraints: $D = \{x \in \mathbb{R}^n : x^{\min} \leq x \leq x^{\max}\}$, where $x^{\min}$ and $x^{\max} \in \mathbb{R}^n$ are vectors and the inequalities are considered componentwise. The loss functional $F : D \to \mathbb{R}$ mirrors the relative deviation between simulated observables and the experimentally measured counterparts, in an arbitrary p-norm:

$$F_p(x) = \|W(f^{\text{sim}}(x) - f^{\text{exp}})\|_p^2,\ p \in [1, \infty].$$

The weighting matrix

$$W = \begin{pmatrix} \frac{w_1}{f_1^{\text{exp}}} & 0 & \ldots & 0 \\ 0 & \frac{w_2}{f_2^{\text{exp}}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \frac{w_m}{f_m^{\text{exp}}} \end{pmatrix}$$

with observable-specific weighting factors $w_1, \ldots, w_m > 0$ enables the user to priorize certain observables over others. For $p = 2$, the loss functional $F_2(x)$ is equivalent to the one defined in Hülsmann et. al. [15]. As formula (1) gives a mathematical description of the parameterization task, general numerical optimization algorithms can be applied.

2.3. The global optimization algorithms

In this subsection, five global optimization algorithms are briefly introduced. They were selected based on a detailed problem analysis and required to cover a range of different global optimization approaches. All of them can be scheduled in parallel, applied to noisy objective functions and used for functions with abortive evaluations.

The first selected global optimization method is **pure random search (PRS)**. It samples iteration points according to a uniform distribution on the search space. The method was not chosen to compete with the other methods but to assess their performance. The second is a simple **recursive random search (RRS)** method, which was successfully used for the optimization of force field

parameters but in connection with a less costly simulation technique [38]. It is also based on uniform distributions and box constrained search domains. In contrast to PRS, RRS iteratively reduces the size of the search domain by a problem-specific scaling factor to zoom in on the global optimum. The number of function evaluations in each iteration is called *population size*.
The third selected algorithm is the **Covariance Matrix Adpatation Evolution Strategy (CMA-ES)** [39, 40], a state-of-the art evolutionary algorithm. It has shown desireable behavior in hundreds of practical applications and test functions [41, 42], including noisy and multimodal objective functions. CMA-ES samples its populations based on multivariate normal distributions. After each iteration, the distributions are sophisticatedly scaled and moved along descent directions. The population size is the most important problem-specific parameter.
The fourth algorithm is **differential evolution (DE)** [27], which has been shown to be superior to a variety of genetic optimization methods [43]. Like genetic methods, it mimes mechanisms like *mutation* and *crossover*. Viewing the objective function as a fitness landscape, the unfitter individuals are replaced by fitter ones. New sampling points emerge as combinations of old sampling points, through mutation and crossover. Two mutations factors and a so-called crossover probability are problem-specific parameters.
The fifth algorithm is a **taboo search (TS)** algorithm which is based on the derivate-free local optimization method by Hooke and Jeeves [30, 44]. It provides additional functionality to prevent preliminary convergence, i.e. intensifying search after a certain number of function evaluations without improvement, diversifying search after another number of function evaluations without improvement and reducing the step size by half after another number of function evaluations without improvement. The initial step sizes and the three mentioned numbers are problem-specific parameters.
The sixth algorithm, CORS, was integrated into CoSMoS and is explained in a later section. Since it is based on metamodels, the metamodeling scheme is explained first.

3. Tailor-made solution: The CoSMoS methodology

3.1. Metamodeling scheme

Metamodels are constructed by appropriate interpolation or regression schemes. They estimate the result of a simulation based on simulation results that were obtained for other force field parameters. Consequently, they become more accurate as the number of former simulations increases.
CoSMoS has three different metamodeling schemes implemented: simple radial basis function (RBF) networks, RBF networks with error bars and *inverse distance weighting*. The metamodels used in this work are built upon simple RBF networks. Those are well-established for interpolation in lower-dimensional spaces. A variety of nonlinear functions can be formed by RBF networks. This enables them to reliably reproduce the multimodal dependency between force field parameters and simulation observables.

3.1.1. Construction of RBF metamodels

RBF networks can form very complex shapes (e.g. cf. Figure 2) but they are composed of simple building blocks, the RBFs. RBFs are rotational-symmetric functions $\tilde{\Phi} : \mathbb{R}^n \to \mathbb{R}$. Convenient examples are linear functions

$$\tilde{\Phi}(x) = \Phi(\|x\|) = \|x\|, \tag{2}$$

cubic ($\|x\|^3$) functions; thin plate splines ($\|x\|^2 \log \|x\|$); Gaussian functions ($\exp(-(\hat{\gamma}\|x\|)^2)$); multiquadrics ($\sqrt{\|x\|^2 + \hat{\gamma}^2}$); and inverse multiquadrics ($(\|x\|^2 + \hat{\gamma}^2)^{-\frac{1}{2}}$) [45]. The *RBF parameter* $\hat{\gamma} > 0$ impacts the shape of the last three types of RBFs. RBF networks are sums of RBFs, which are located at distinct sampling points $x_1, \ldots, x_i \in \mathbb{R}^n$.

It is common to combine RBFs with low-order polynomial basis functions: $p_1, \ldots, p_r,\ r \in \mathbb{N}$. Consequently, the metamodels take the form

$$\mathcal{M}^{(\nu)}(x) = \sum_{i=l}^{i} \alpha_l^{(\nu)} \Phi(\|x - x_l\|) + \sum_{k=1}^{r} \beta_k^{(\nu)} p_k(x),$$

where $\nu \in \{1, \ldots, m\}$ is the index of the target observable.

Given distinct data points $(x^{(l)}, f_\nu^{\mathrm{sim}}(x^{(l)}))$, $1 \le l \le i$, from previous simulations, the model coefficients $\alpha_j^{(\nu)}$, $1 \le j \le q$, and $\beta_k^{(\nu)}$, $1 \le k \le r$, can be adjusted to interpolate the previous simulations results. Therefore, the interpolation condition

$$\mathcal{M}^{(\nu)}(x^{(l)}) = f_\nu^{\mathrm{sim}}(x^{(l)}),\ 1 \le l \le i, \tag{3}$$

is expressed as a linear equation system. Each solution to this equation system provides a set of coefficients which define an interpolation of the data points. To obtain a unique solution, the RBF coefficients are forced to satisfy an additional constraint:

$$\sum_{k=1}^{q} \alpha_l^{(\nu)} p_k(x^{(l)}) = 0,\ 1 \le k \le r. \tag{4}$$

This guarantees the uniqueness of the obtained coefficients, presuming that the RBF is either positive or negative definite [36].

Replacing $f^{\mathrm{sim}}(\tilde{x})$ by $\mathcal{M}(\tilde{x})$ in the definition of the loss functional gives a metamodel $\mathcal{M}_F(\tilde{x})$ of the loss functional.

3.1.2. Normalization of the metamodels

Since force field parameters as well as observables may have diverse quantity dimensions, normalization is a natural way to smooth the shape of metamodels and to simplify the choice of the metamodel parameters.

First, the search space, i.e. the space of force field parameters is normalized to the n-dimensional unit cube by a transformation

$$\tilde{x} := \left(\frac{x_k - x_k^{\min}}{x_k^{\max} - x_k^{\min}} \right)_{1 \le k \le n},\ x \in D. \tag{5}$$

Second, it is also useful to normalize the space of system observables. This simplifies the metamodeling process as a similar metamodeling approach can be applied to all different kinds of system observables regardless of the noise or range of a quantity. In most cases, there is detailed knowledge not only about the desired target values f_ν^{exp} but also about the standard deviations of the relative noise $s_\nu^{\text{sim}}, 1 \leq \nu \leq m$. The transformation

$$\tilde{f}_\nu^{\text{sim}} := \left(\frac{f_\nu^{\text{sim}} - f_\nu^{\text{exp}}}{s_\nu^{\text{sim}} f_\nu^{\text{exp}}} \right)_{1 \leq \nu \leq m}, \; f^{\text{sim}} \in \mathbb{R}^m, \tag{6}$$

sets the standard deviation to a value of about 1 and the desired target value to 0.
For all of the above-mentioned benefits, the metamodeling approach described in the previous subsection was applied to the transformed data instead of the original data. The metamodel of the loss functional was also reformulated, in terms of normalized parameters and unnormalized observables.

3.2. Intelligent sampling

In this section, two sampling mechanisms, CORS [19] and a new method for CoSMoS developed by the authors, are presented. Both exploit a metamodel of the loss functional to focus sampling on points were the loss functional values are comparatively small. Both also generate sampling points through solving a surrogate optimization problem in every iteration. By this means, sampling is directed towards the most promising parameter sets, aiming to accumulate sampling points in the neighborhood of a global optimum. This is illustrated by the black sampling points in Figure 2 (left).

3.2.1. CORS: Constrained Optimization using Response Surfaces

CORS is a metamodel-based optimization method for expensive black-box functions [19, 37]. It has been applied to various practical problems [46–48]. CORS exploits all of the previously produced data by minimizing the metamodel of the objective function. Exploration is granted by introducing a taboo region around the sampling points, which results in a constrained optimization problem.
Concretely, the taboo region is defined as

$$R_{\tilde{r},\tilde{Q}} := \bigcup_{\tilde{x} \in \tilde{Q}} U_{\tilde{r}}(\tilde{x}),$$

where $U_{\tilde{r}}(\tilde{x})$ is a ball of radius $\tilde{r} > 0$ around a point $\tilde{x}$ in the transformed search space, i.e. the n-dimensional unit cube. $\tilde{Q}$ is the set of transformed sampling points. In order to prevent $R_{\tilde{r},\tilde{Q}}$ from covering the whole unit cube, the radius has to fulfill the requirement

$$\tilde{r} < \delta_{\tilde{Q}}^{\max} := \max_{\tilde{x} \in [0,1]^n} \min_{\tilde{z} \in \tilde{Q}} \|\tilde{x} - \tilde{z}\|.$$

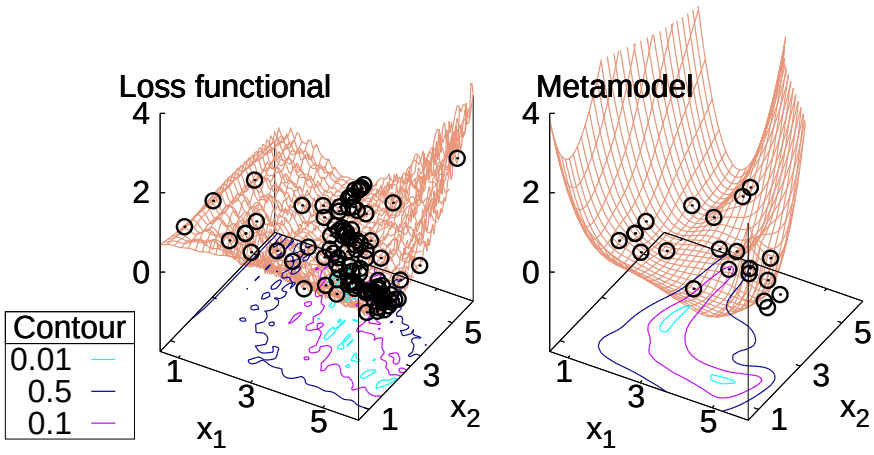

Figure 2: Left: loss functional for a test problem defined later. x_1 and x_2 denote two artificial force field parameters. The black sampling points were generated by CoSMoS (new) using metamodels composed by cubic RBFs with $d = 0$. It can be seen that the distribution of sampling points adapts the shape of the loss function. Right: Metamodel of the loss functional after 20 iterations of CoSMoS. The black sampling points are the sampling points generated throughout the first 20 iterations. It can be seen that the general trend of the loss functional (left) is captured by the preliminary metamodel (right). Intelligent sampling strategies implement preliminary metamodels to focus sampling on more interesting parts of the search space. Both plots contain contour lines of the plotted surfaces.

The radius $\tilde{r}_i$ in iteration i, considered as a measure of exploration, is varied throughout the optimization process. Its scaling repeatedly runs through the so-called *search-pattern*

$$V^{(1)} := (0.9,\ 0.75,\ 0.25,\ 0.05,\ 0.03,\ 0.0),$$

formally $\tilde{r}_i := V_{I(i)} \cdot \delta_{\tilde{Q}}^{\max}$ with $I(i) = (i \mod 6) + 1$.
To summarize, the candidate point in iteration i of CORS is generated by solving the constrained optimization problem

$$\min \mathcal{M}_F(\tilde{x}), \quad \tilde{x} \notin R_{\tilde{r}_i, \tilde{Q}}. \tag{7}$$

3.2.2. A new sampling technique for CoSMoS

Starting from CORS, a new sampling technique was tailored to the force field calibration problem in order to save explorative function evaluations. The new technique utilizes the fact that $\xi = 0$ is a lower bound for the loss functional, reducing exploration as soon as the sampled function values approach ξ. This was achieved by replacing the rigid taboo approach by a penalty technique.
The penalty term

$$\bar{p}(\tilde{x}) := \frac{\delta_{\tilde{Q}}^{\max}}{\min\limits_{\tilde{x}_i \in \tilde{Q}} \|\tilde{x} - \tilde{x}_i\|} \geq 1,$$

also defined for normalized variables, grows to infinity as $\tilde{x}$ approaches a sampling point. The penalty function and the metamodel of the loss were assembled

to

$$\tau_\gamma(\tilde{x}) := \bar{p}(\tilde{x})^\gamma \cdot (\mathcal{M}_F(\tilde{x}) - \xi).$$

The parameter γ controls the amount of exploration, similar to $\tilde{r}_i$ in CORS. Therefore, the search pattern idea was adapted from CORS, replacing γ by $\gamma_i := V_{I(i)}$.
While CORS samples candidates points according to the constrained minimization problem (7), the new method samples candidate points by solving

$$\min \tau_{\gamma_i}(\tilde{x}), \tilde{x} \in [0,1]^n$$

in each iteration i.

3.2.3. Parallel and abortive simulations

For both sampling techniques, parallel and abortive simulations were handeled in a similar way. The normalized parameter sets $\tilde{x}$ (for currently running and aborted simulations) were included into $\tilde{Q}$ to deflect the sampling from all previous sampling points, including abortive and currently running simulations. CoSMoS automatically classifies simulations as abortive when they do not deliver useful results in a time range defined by the user.
The overall algorithm, including all presented features, is presented in Figure 3.

4. Results

The presented algorithms were assessed for the global optimization of force field parameters. This was done in two steps: First, all of the algorithms were applied to a set of suitable test functions. Algorithm parameters were tuned for the most promising: RRS, CMA-ES and CoSMoS. Second, these three were applied to an example simulation system.

4.1. Comparison with other methods based on test functions

4.1.1. Definition of test problems

In order to assess the described algorithms, mathematical test functions have to be used instead of costly molecular simulations. The test functions defined below have properties similar to those of real simulation observables: noisy evaluations, a continuous dependency between parameters and mean function values, a "global trend", multiple local optima and, eventually, local bumpiness. They take the place of the simulated observables in the definition of the loss functional.
The first kind of test functions, denoted as ϕ, were taken from a study of two-center Lennard-Jones fluids with pointdipole [49]. Four force field parameters had been used: reduced dipolar momentum μ^2, reduced bond length L, and the reduced Lennard-Jones parameters σ, ε; $x := (\mu^2, L, \sigma, \varepsilon)$. Three observables had been considered: saturated liquid density, heat of vaporization, and vapor pressure at temperatures T. The nonlinear dependencies between force field

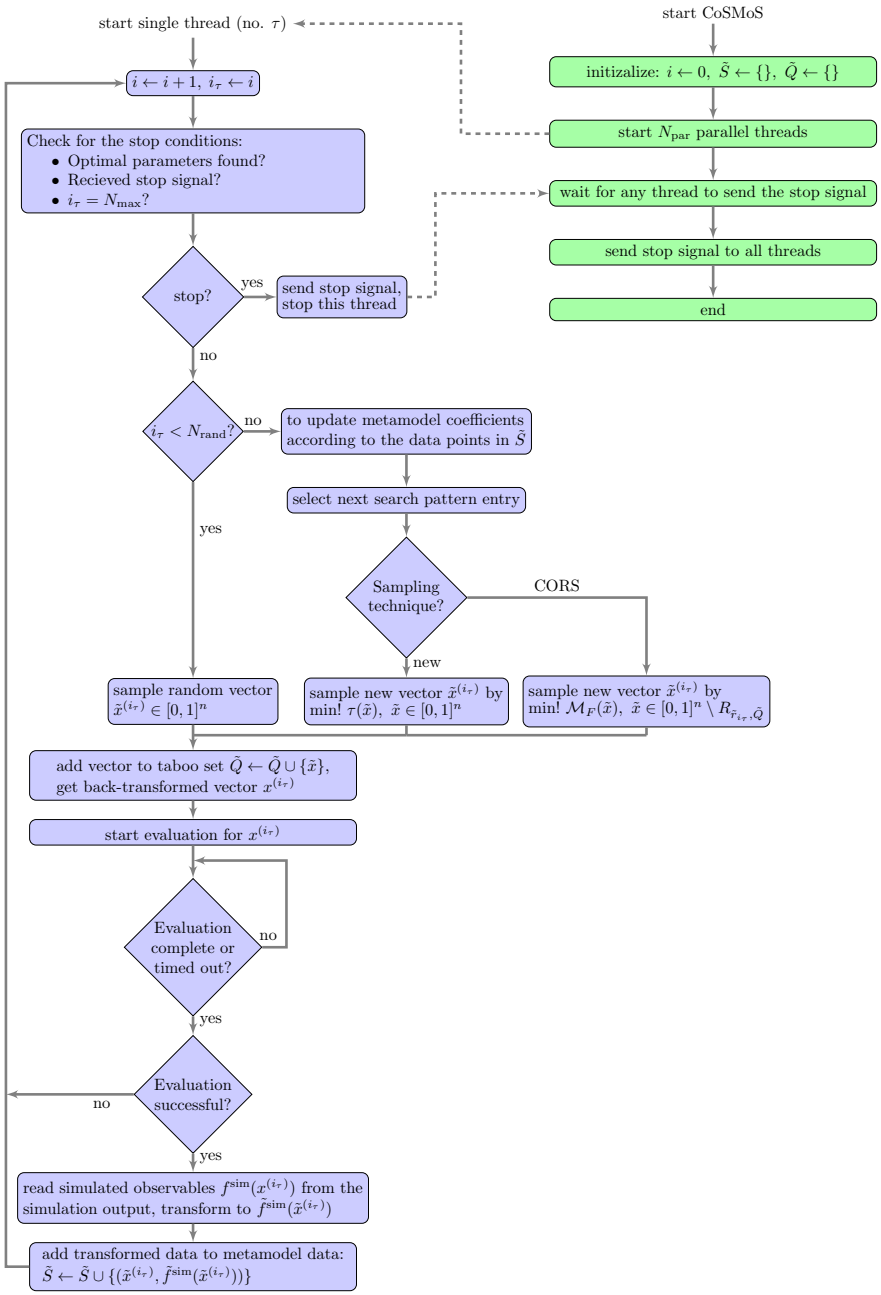

Figure 3: Flowchart of the parallel CoSMoS algorithm: The main routine (upper right) initializes the iteration count i, the set of simulated data $\tilde{S}$, and the taboo set $\tilde{Q}$ which are common to all threads. Then it creates $N_{\text{par}} \in \mathbb{N}$ parallel threads. Each of the runs through four steps: First, it checks for stopping conditions. These are: a threshold for the number of evaluations $N_{\max} \in \mathbb{N}$ and the detection of an optimal parameter vector. Second, it samples a new solution vector $\tilde{x}^{(i_\tau)}$, randomly (for the first $N_{\text{rand}} \in \mathbb{N}$ iterations) or according to one of the presented intelligent sampling strategies. Third, it evaluates the objective function, and fourth, it adds the result of the evaluation to $\tilde{S}$. Then it starts again from the beginning. CoSMoS contains further functionality to rule out conflicts between the threads. Those are not described by the flowchart.

parameters and observables had been approximated by functions $\phi_T^{(1)}$, $\phi_T^{(2)}$, and $\phi_T^{(3)}$. These functions have been used to assess GROW [16]. In the present work, $\phi_T^{(1)}(x)$ and $\phi_T^{(2)}(x)$ were considered at $T \in \{65\text{ K}, 75\text{ K}, \ldots, 115\text{ K}\}$ and compared with experimental data for Nitrogen [50]. The search space was defined by $x^{\min} := (0.29, 0.3, 0, 0)$ and $x^{\max} := (0.4, 0.4, 0.0468, 0.24)$.
The second kind of test functions, denoted as ψ, are artificial functions defined on spaces of arbitrary dimension n, in order to generalize conclusions to calibration problems with more or fewer than four parameters. They were defined as follows:

$$\psi_k^{(n)}(x) := x_{k \bmod n} \left(\log x_{k+1 \bmod n}\right)^{k+1} + \cos\left(x_{k+1 \bmod n}^{k+1} \sin(6 x_{k \bmod n})\right),\ k \in \{1, \ldots, m\}.$$

The target values for the calibration were defined as $\psi_k^{(n),\exp} := 4 + k,\ k \in \{1, \ldots, m\}$.
A set of eleven test problems was defined (cf. Table 1). The quadratic loss functional F_2 was used with uniform weights. Figure 2 shows the loss functional for test problem 1. Artificial Gaussian distributed noise with a relative standard deviation of s^{sim} was added to the functions. The amount of noise was varied to study the robustness of the algorithms.

Table 1: List of test problems based on the defined test functions. $\mathbf{f^{sim}}$ denotes the vector of test functions, which replace simulation observables in numerical tests. $\mathbf{s^{sim}}$ is the relative standard deviation of normal distributed noise which was added to the test function values.

No.	**n**	**m**	$\mathbf{f^{sim}}$	$\mathbf{s^{sim}}$ (rounded)
1	2	2	$(\psi_1^{(2)}, \psi_2^{(2)})$	no noise added
2	2	2	$(\psi_1^{(2)}, \psi_2^{(2)})$	0.8%
3	2	2	$(\psi_1^{(2)}, \psi_2^{(2)})$	8%
4	2	5	$(\psi_1^{(2)}, \ldots, \psi_5^{(2)})$	no noise added
5	2	5	$(\psi_1^{(2)}, \ldots, \psi_5^{(2)})$	0.8%
6	2	5	$(\psi_1^{(2)}, \ldots, \psi_5^{(2)})$	8%
7	4	12	$(\phi_{65\text{K}}^{(1)}, \phi_{65\text{K}}^{(2)}, \phi_{75\text{K}}^{(1)}, \phi_{75\text{K}}^{(2)}, \ldots, \phi_{115\text{K}}^{(1)}, \phi_{115\text{K}}^{(2)})$	no noise added
8	4	12	$(\phi_{65\text{K}}^{(1)}, \phi_{65\text{K}}^{(2)}, \phi_{75\text{K}}^{(1)}, \phi_{75\text{K}}^{(2)}, \ldots, \phi_{115\text{K}}^{(1)}, \phi_{115\text{K}}^{(2)})$	$(0.4\%, 0.8\%, 0.4\%, 0.8\%, \ldots)$
9	4	12	$(\phi_{65\text{K}}^{(1)}, \phi_{65\text{K}}^{(2)}, \phi_{75\text{K}}^{(1)}, \phi_{75\text{K}}^{(2)}, \ldots, \phi_{115\text{K}}^{(1)}, \phi_{115\text{K}}^{(2)})$	8%
10	6	8	$(\psi_1^{(6)}, \ldots, \psi_8^{(6)})$	0.8%
11	10	9	$(\psi_1^{(10)}, \ldots, \psi_9^{(10)})$	0.8%

4.1.2. Comparison of all algorithms
As a first step, all algorithms were applied to all test problems to roughly estimate their performance. The algorithms' parameters, such as the population size of evolution methods, were chosen by hand (cf. Table 2). Thereby, extensive optimizations of the algorithms' parameters were avoided for algorithms turning out less appropriate. Each optimization run was stopped after $N_{\max} = 100$

Table 2: Algorithm parameters for first application to the test problems

RRS		**TS**	
population size	4	increments of local optimizer	$h = 0.1 \cdot (x^{\max} - x^{\min})$
scaling factor	0.8	#eval until intensivation	10
initial rectangle	search space D	#eval until diversification	5
		#eval until reduction of h	25
CMA-ES		**CoSMoS (CORS)**	
population size	10	RBF type	cubic
#selected candidates	half of population	degree of polynomial	1
starting point	center of D	sampling technique	CORS
initial standard deviations	$(x^{\max} - x^{\min})/3$	search pattern	$V^{(2)}$
DE		**CoSMoS (new)**	
population size	10	RBF type	cubic
combination factor K	0.5	degree of polynomial	1
combination factor F	random in $[-2, 2]$	sampling technique	new
crossover probability	0.8	search pattern	(2.0)

Table 3: Comparison of all algorithms for test problems (TP) 1–6. Smallest detected loss functional value after 20, 40, 60, 80 and 100 function evaluations (#Eval), averaged over 10 replicates. Best values are printed bold.

		Algorithm						
TP	**#Eval**	PRS	RRS	CMA-ES	DE	TS	CoSMoS (CORS)	CoSMoS (new)
1	20	2.50E-2	7.54E-3	9.17E-3	2.15E-2	2.51E-2	7.01E-3	**4.93E-3**
	40	6.08E-3	3.49E-3	4.17E-3	8.34E-3	7.68E-3	1.64E-3	**1.58E-3**
	60	4.71E-3	1.60E-3	2.07E-3	4.43E-3	4.67E-3	1.04E-3	**6.58E-4**
	80	4.47E-3	5.59E-4	7.44E-4	1.75E-3	3.63E-3	**4.08E-4**	4.35E-4
	100	3.58E-3	2.02E-4	3.73E-4	1.32E-3	3.50E-3	**3.20E-4**	4.34E-4
2	20	2.29E-2	1.76E-2	7.37E-3	3.35E-2	2.46E-2	5.09E-3	**3.48E-3**
	40	3.93E-3	6.65E-3	4.76E-3	1.09E-2	1.00E-2	**1.63E-3**	1.89E-3
	60	3.32E-3	1.24E-3	1.18E-3	4.81E-3	5.49E-3	1.10E-3	**1.04E-3**
	80	3.24E-3	1.24E-3	1.03E-3	3.35E-3	2.65E-3	**2.65E-4**	6.54E-4
	100	2.86E-3	7.35E-4	7.84E-4	2.88E-3	2.65E-3	**1.85E-4**	4.92E-4
3	20	1.13E-2	6.13E-3	9.75E-3	4.22E-2	1.27E-2	**4.59E-3**	6.27E-3
	40	6.33E-3	1.90E-3	2.33E-3	6.66E-3	1.16E-2	**1.76E-3**	2.35E-3
	60	3.19E-3	1.20E-3	1.36E-3	3.96E-3	4.86E-3	1.01E-3	**7.32E-4**
	80	2.34E-3	9.24E-4	8.20E-4	2.93E-3	4.01E-3	5.90E-4	**4.27E-4**
	100	1.82E-3	7.46E-4	6.74E-4	2.45E-3	3.84E-3	5.90E-4	**3.13E-4**
4	20	1.49E-1	8.28E-2	3.57E-2	1.14E-1	8.38E-2	**1.55E-2**	1.89E-2
	40	7.83E-2	3.63E-2	2.21E-2	9.13E-2	3.14E-2	**9.87E-3**	1.27E-2
	60	6.67E-2	2.23E-2	1.64E-2	4.60E-2	2.79E-2	**6.20E-3**	9.46E-3
	80	5.37E-2	9.99E-3	1.14E-2	3.12E-2	2.79E-2	**5.01E-3**	8.46E-3
	100	4.72E-2	7.99E-3	8.36E-3	2.57E-2	2.79E-2	**4.05E-3**	8.46E-3
5	20	7.82E-2	1.09E-1	4.65E-2	1.47E-1	7.94E-2	**1.72E-2**	2.90E-2
	40	5.52E-2	3.95E-2	2.55E-2	1.02E-1	2.84E-2	**1.05E-2**	1.91E-2
	60	5.42E-2	2.03E-2	1.91E-2	6.51E-2	2.69E-2	**7.65E-3**	1.59E-2
	80	5.25E-2	1.37E-2	1.46E-2	4.81E-2	2.69E-2	**6.06E-3**	1.20E-2
	100	3.58E-2	1.06E-2	1.08E-2	3.87E-2	3.24E-2	**5.37E-3**	1.04E-2
6	20	1.78E-1	1.24E-1	3.73E-2	1.00E-1	8.32E-2	**1.49E-2**	2.37E-2
	40	9.29E-2	9.04E-2	2.21E-2	6.61E-2	3.09E-2	**9.58E-3**	1.95E-2
	60	6.39E-2	6.94E-2	1.89E-2	5.06E-2	2.99E-2	**8.00E-3**	1.73E-2
	80	6.39E-2	6.08E-2	1.74E-2	3.83E-2	2.89E-2	**7.84E-3**	1.26E-2
	100	4.84E-2	5.70E-2	1.01E-2	2.56E-2	2.89E-2	**7.55E-3**	9.92E-3

Table 4: Comparison of all algorithms for test problems (TP) 7–11. Smallest detected loss functional value after 20, 40, 60, 80 and 100 function evaluations (#Eval), averaged over 10 replicates. Best values are printed bold.

		Algorithm						
TP	**#Eval**	PRS	RRS	CMA-ES	DE	TS	CoSMoS (CORS)	CoSMoS (new)
7	20	8.59E-3	8.32E-3	1.09E-2	2.13E-2	4.18E-2	**1.83E-3**	2.89E-3
	40	5.19E-3	4.71E-3	6.19E-3	7.30E-3	1.35E-2	**9.22E-4**	1.21E-3
	60	3.95E-3	2.95E-3	6.13E-3	5.98E-3	9.50E-3	**8.68E-4**	1.05E-3
	80	3.77E-3	2.58E-3	4.82E-3	4.40E-3	3.37E-3	**8.68E-4**	1.01E-3
	100	3.39E-3	2.40E-3	3.93E-3	3.38E-3	3.35E-3	**8.56E-4**	9.86E-4
8	20	1.01E-2	8.27E-3	1.31E-2	1.85E-2	2.88E-2	**3.51E-3**	2.25E-3
	40	4.58E-3	3.05E-3	6.53E-3	1.07E-2	1.48E-2	**9.79E-4**	1.36E-3
	60	4.03E-3	2.31E-3	4.85E-3	7.96E-3	6.53E-3	**8.37E-4**	1.15E-3
	80	3.06E-3	1.86E-3	4.48E-3	5.83E-3	3.34E-3	**8.06E-4**	1.09E-3
	100	2.98E-3	1.68E-3	4.03E-3	4.50E-3	3.23E-3	**8.06E-4**	9.92E-4
9	20	1.51E-2	1.60E-2	1.36E-2	1.82E-2	3.28E-2	**5.31E-3**	6.97E-3
	40	1.07E-2	8.54E-3	9.61E-3	1.01E-2	1.09E-2	**4.53E-3**	5.01E-3
	60	8.19E-3	6.02E-3	8.88E-3	7.40E-3	1.02E-2	**3.84E-3**	4.50E-3
	80	7.16E-3	4.51E-3	7.57E-3	6.01E-3	8.09E-3	**3.48E-3**	4.20E-3
	100	6.21E-3	4.32E-3	7.11E-3	5.37E-3	7.09E-3	**3.31E-3**	4.04E-3
10	20	4.07E-1	4.73E-1	4.87E-1	5.32E-1	7.48E-1	3.19E-1	**2.50E-1**
	40	3.57E-1	3.54E-1	3.36E-1	4.21E-1	5.33E-1	2.10E-1	**2.00E-1**
	60	3.40E-1	3.01E-1	2.70E-1	3.08E-1	3.37E-1	**1.55E-1**	1.85E-1
	80	3.31E-1	2.71E-1	2.24E-1	2.69E-1	3.14E-1	**1.20E-1**	1.60E-1
	100	3.29E-1	2.56E-1	1.94E-1	2.36E-1	2.96E-1	**1.02E-1**	1.40E-1
11	20	6.94E-1	7.12E-1	5.96E-1	8.17E-1	7.39E-1	6.25E-1	**5.69E-1**
	40	6.35E-1	**4.13E-1**	5.11E-1	6.37E-1	6.71E-1	4.81E-1	4.64E-1
	60	5.77E-1	**3.44E-1**	4.22E-1	4.73E-1	5.74E-1	4.11E-1	4.29E-1
	80	5.73E-1	**3.15E-1**	3.51E-1	4.59E-1	5.23E-1	3.70E-1	4.29E-1
	100	5.42E-1	**2.95E-1**	3.36E-1	4.13E-1	4.42E-1	3.30E-1	4.19E-1

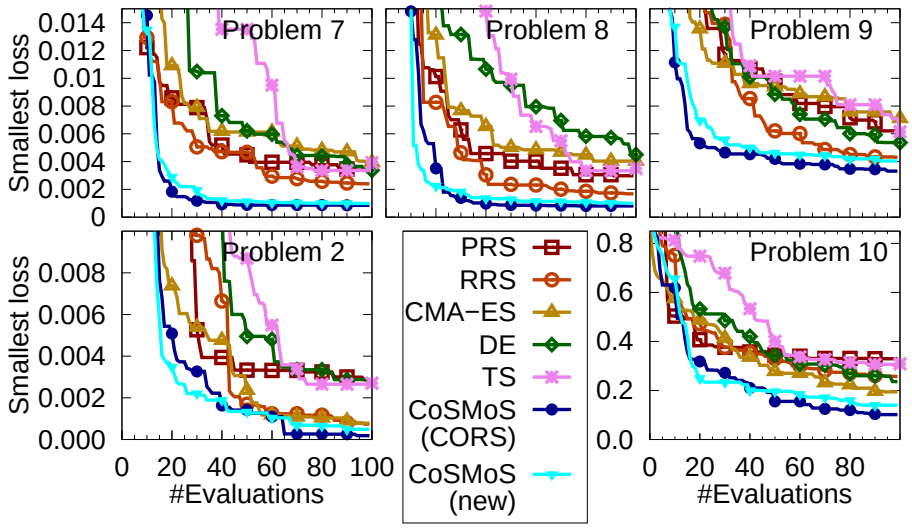

Figure 4: Comparison of all algorithms for test problems 2,7,8,9, and 10. Evolution of the smallest detected loss functional value during the optimization run, averaged over ten replicates.

evaluations of the loss functional. We observed the evolution of the smallest loss functional value found within the optimization process.
The values are averaged over ten replicates of each optimization run in order to respect the stochastic nature of the algorithms. TS, a fully determinstic method, was started from random initial points. For CoSMoS, the normalization of observables was not used, N_{rand} was set to 10. Parallelization was switched off, $N_{\text{par}} := 1$. The numerical results for all test problems are given in Tables 3 and 4. Figure 4 illustrates the results for test problems 2, 7, 8, 9 and 10.
Obviously, CoSMoS dominated the other algorithms on all test problems except no. 11. The CORS sampling with search pattern $V^{(1)}$ was slightly better than the new sampling technique with search pattern (2.0) in most applications, especially no. 4-11. At this point, it was not clear if this was due to the search pattern or due to the sampling technique. For test problem 8, both versions of CoSMoS reached a value smaller than 1E-3. Remarkably, alike values can only be found in the immediate neighborhood of the global optimum, which has a loss functional value of 7E-4. Test problem 8 is particularly interesting, because it is based on real molecular simulations, with a realistic amount of noise added. CoSMoS was outperformed on the test problem with the ten-dimensional search space, no. 11. Here, RRS performed best.
For impartial assessment, the algorithms are compared to PRS. Generally, RRS was clearly better than PRS, except for test problem 6. So was CMA-ES, except for problems 7–9, i.e. the test problems based on Nitrogen simulations. DE and

TS could not prove any significant benefit over PRS for the test problems. Hence, we rated them as unsuitable and sorted them out at this point.
Moderate noise had no noticeable effect on the algorithms (test problems 2, 5 and 8). High noise (no. 3,6 and 9) slowed down all of them. Especially for the simulation-based test problem (no. 9), none of them could find values smaller than 1E-3, any more.

4.1.3. Parameter tuning for RRS, CMA-ES and CoSMoS

In order to exhaust the full potential of RRS, CMA-ES and CoSMoS, the following parameters were tuned: RRS' population size and scaling factor; CMA-ES' population size; and CoSMoS' search pattern, RBF type, and RBF parameter. The tuning was carried out with view to test problem 8, which was considered most related to real applications.

Table 5: Parameter tuning for CoSMoS, CMA-ES and RRS. Best values are printed bold.

		RRS with population size 4; different scaling parameters						
TP	**#Eval**	**0.2**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
8	20	9.81E-3	9.00E-3	8.06E-3	**6.93E-3**	7.87E-3	8.23E-3	1.08E-2
	40	6.62E-3	3.87E-3	3.35E-3	**2.24E-3**	2.63E-3	3.27E-3	4.67E-3
	60	5.97E-3	1.13E-3	2.47E-3	**1.75E-3**	1.87E-3	2.16E-3	3.00E-3
	80	5.74E-3	2.97E-3	2.13E-3	**1.64E-3**	1.65E-3	1.81E-3	2.30E-3
	100	5.69E-3	2.90E-3	2.06E-3	1.59E-3	**1.58E-3**	1.66E-3	1.97E-3
		RRS with scaling parameter 0.6; different population sizes						
TP	**#Eval**	**3**	**4**	**5**	**6**	**8**	**10**	**15**
8	20	**7.86E-3**	8.13E-3	8.40E-3	9.61E-3	9.11E-3	1.06E-2	9.26E-3
	40	3.32E-3	**2.82E-3**	2.96E-3	3.56E-3	4.44E-3	4.44E-3	5.61E-3
	60	2.34E-3	2.01E-3	**1.91E-3**	2.19E-3	2.57E-3	2.70E-3	3.81E-3
	80	2.19E-3	1.82E-3	**1.66E-3**	1.86E-3	2.06E-3	2.11E-3	2.70E-3
	100	2.14E-3	1.78E-3	**1.59E-3**	1.75E-3	1.83E-3	1.80E-3	2.24E-3
		CMA-ES; different population sizes						
TP	**#Eval**	**3**	**5**	**8**	**10**	**15**	**20**	
8	20	1.21E-2	1.35E-2	1.22E-2	1.49E-2	**1.08E-2**	1.31E-2	
	40	6.92E-3	6.99E-3	6.55E-3	7.67E-3	**6.29E-3**	8.18E-3	
	60	5.15E-3	4.56E-3	**4.37E-3**	5.57E-3	4.68E-3	5.42E-3	
	80	4.05E-3	3.41E-3	**3.38E-3**	4.33E-3	3.81E-3	4.08E-3	
	100	3.19E-3	**3.01E-3**	3.11E-3	3.75E-3	3.32E-3	3.64E-3	
		CoSMoS (new) with cubic RBF and mp=0; different search patterns						
TP	**#Eval**	**(2)**	**(1)**	**(0)**	**(20)**	$\mathbf{V^{(1)}}$	$\mathbf{V^{(2)}}$	
8	20	2.82E-3	3.52E-3	**1.71E-3**	6.01E-3	1.87E-3	2.15E-3	
	40	1.45E-3	2.63E-3	**1.06E-3**	3.02E-3	1.16E-3	1.15E-3	
	60	1.32E-3	2.56E-3	9.37E-4	2.34E-3	**8.52E-4**	8.78E-4	
	80	1.14E-3	1.11E-3	9.18E-4	2.34E-3	**7.68E-4**	7.73E-4	
	100	1.09E-3	8.93E-4	7.85E-4	2.09E-3	**7.68E-4**	7.73E-4	
		CoSMoS (new) with search pattern B2; different RBFs						
TP	**#Eval**	**cubic**	**inv. multiquadric**		**multiquadric**		**Gaussian**	
8	20	4.80E-3	7.75E-3		**2.72E-3**		2.73E-3	
	40	2.34E-3	1.13E-3		**1.01E-3**		1.20E-3	
	60	1.83E-3	1.01E-3		**9.20E-4**		1.12E-3	
	80	1.08E-3	9.76E-4		**8.98E-4**		1.07E-3	
	100	**8.67E-4**	9.49E-4		8.78E-4		1.05E-3	

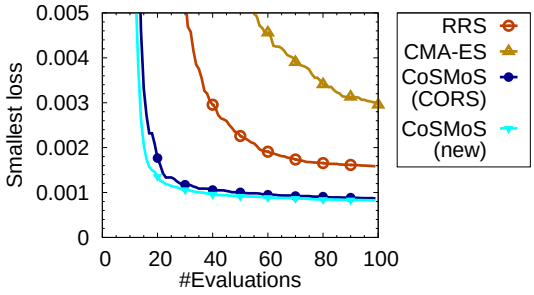

Figure 5: Comparison of RRS, CMA-ES, CoSMoS(CORS) and CoSMoS(new) with tuned algorithm parameters, applied to test problem 8 and averaged over 100 replicates.

RRS. The scaling parameter and population size were tuned sequentially: First, scaling parameters from 0.2 to 0.9 were used on test problem 8, 100 replicates for each setting. The population size was fixed to 4. According to table 5, 0.6 led to the best results. Setting the scaling parameter too small, e.g. 2, leads to a serious performance loss. Second, population sizes from 3 to 15 were used with the scaling parameter fixed to 0.6. The optimal population size was 5.

CMA-ES. CMA-ES population sizes from 3 to 20 were used on test problem 8, 100 replicates for each population size. None of the tested population sizes could enhance the previous efficiency significantly (cf. Table 5). A population size of 8 seemed most appropriate.

CoSMoS. The search pattern and metamodels were tuned sequentially, using the new sampling strategy: First, the search pattern was varied, 10 replicates for each. The candidates were search patterns with one element $((2), (1), (0), (20))$ and multiple elements ($V^{(1)}$ as before, $V^{(2)} = (2, 1, 0.5, 0.25, 0.13, 0)$). For the new sampling technique, switching to the multi-element pattern $V^{(1)}$ augmented the performance (cf. Table 5). Second, different metamodeling techniques were compared. RBF parameters were tuned via leave-one-out-cross validation [51], which we do not describe here in detail. Three promising RBF settings were selected: inverse multiquadrics with RBF parameter $\hat{\gamma} = 2$ and a polynomial of degree $d = 0$, multiquadrics with $\hat{\gamma} = 1.2$ and $d = 1$, and Gaussian RBFs with $\hat{\gamma} = 0.7$ and $d = 0$. In a final comparison, all of them outperformed the previously used cubic RBF (cf. Table 5) concerning pace of optimization. Multiquadrics were most appropriate.

Final comparison. Finally, the algorithms with tuned parameters were compared on test problem 8, 100 replicates each (cf. Figure 5). Changing the population size did not overcome the difficulties of CMA-ES concerning test problem 8. Tuning the parameters of RRS and CoSMoS brought a small gain of performance. In general, the overall picture remains: CoSMoS clearly dominates RRS and CMA-ES, but now with slight advantages for the new sampling technique.

4.2. Comparison of RRS, CMA-ES and CoSMoS based on simulations

4.2.1. Definiton of the simulation systems

The three selected methods were used to parameterize a molecular model for phosgene ($COCl_2$). The Lennard-Jones parameters of phosgene, $x = (\sigma_C, \sigma_O, \sigma_{Cl}, \varepsilon_C, \varepsilon_O, \varepsilon_{Cl})$, were optimized in order to reproduce the saturated liquid density ρ_l at temperatures ranging from 235 K to 280 K.

The molecular model includes fixed bond lengths and angles, an improper dihedral, four partial point charges at the atom centers, and Lennard-Jones-Parameters for three atom types [22]. The intermolecular parameters and partial atomic charges were derived from quantum mechanical calculations using GAMESS and $WOLF_2PACK$, respectively [52–54].

Parallel NpT simulations were carried out at vapor liquid equilibrium for seven temperatures and their corresponding pressures, using GROMACS [55]. The simulation box contained 750 molecules. We used the GROW interface and its capabilities to initialize and equilibrate the simulations [15]. Each equilibration cycle as well as the production run included 250000 time steps with a length of 2 fs. The pre-equilibration included 750000 time steps with a length of 2 fs.

4.2.2. Global and local optimization of phosgene

For each of the optimization algorithms, the global search was combined with the Steepest Descent (SD) local optimization method. First, the global optimization was carried out to find close-to-optimal parameters. Then, local SD optimizations were carried out with GROW to turn the close-to-optimal parameters into optimal ones. The objective function was a quadratic loss functional F_2 with uniform weights. The box constraints of the search space were set to $0.15 \leq \sigma_C, \sigma_O, \sigma_{Cl}[\text{nm}] \leq 0.85$; $0.005 \leq \varepsilon_C[\text{kJ/mol}] \leq 0.995$; $0.01 \leq \varepsilon_O, \varepsilon_{Cl}[\text{kJ/mol}] \leq 1.99$.

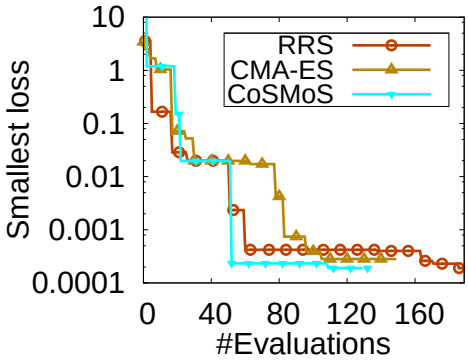

Figure 6: Comparison of RRS, CMA-ES and CoSMoS, applied to phosgene. Evolution of the smallest detected loss functional value during the optimization run.

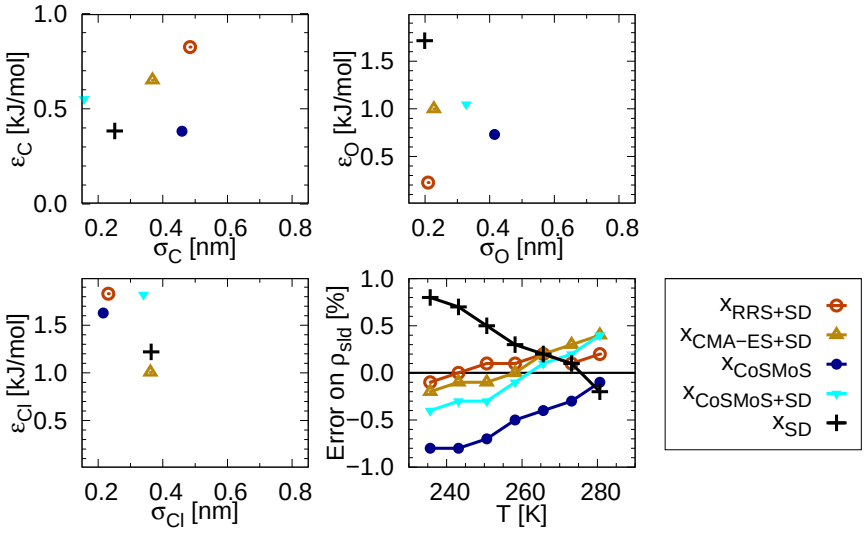

Figure 7: Lennard-Jones parameters for phosgene after optimization with Steepest Descent (SD), RRS+SD, CMA-ES+SD, CoSMoS, and CoSMoS+SD. x_{SD} is taken from [22]. Bottom right: Percentual errors on saturated liquid density.

RRS. The scaling parameter and population size were set to 0.75 and 7, respectively. They were set bigger than the tuned scaling parameter and population size from the previous section because of the higher-dimensional parameter space. The initial parameter vector had a loss functional value of 3.0. Within 17 function evaluations, RRS found a parameter vector with a loss functional value of 2.9E-2. In evaluation 52, it found a parameter vector with a loss functional value of 2.4E-3, $x_{\mathrm{RRS}} = (0.487, 0.209, 0.231, 0.824, 0.225, 1.832)$, cf. Figure 6. The subsequent local optimization could further improve the loss functional value to 8E-6 within only 24 evaluations of the loss functional. The final parameter vector was $x_{\mathrm{RRS+SD}} = (0.484, 0.209, 0.231, 0.824, 0.225, 1.832)$. Remarkably, the final parameters only differed marginally from x_{RRS}. Figure 7 shows that the percentual error of the saturated liquid density was optimal, i.e. as accurate as the experimental measurement. The experimental uncertainty is approximately 0.5%. RRS without local optimization could further improve x_{RRS} (Figure 6), but it did not reach the experimental accuracy.

CMA-ES. The population size was set to 6. The starting point and initial standard deviations were again set to the center of the search space and a third of its diagonal, respectively. Until evaluation 40, CMA-ES performs similarly efficiently as RRS. Thereafter, it takes longer to improve the loss functional values. The best parameter vector was found after 106 evaluations, $x_{\mathrm{CMA-ES}} = (0.368, 0.227, 0.362, 0.651, 0.999, 1.006)$. The subsequent local op-

timization could further improve the loss functional value to 3.8E-5 within 41 evaluations $x_{\mathrm{CMA-ES+SD}} = (0.368, 0.227, 0.363, 0.651, 0.999, 1.006)$. Again, the final parameters only differed marginally from $x_{\mathrm{CMA-ES}}$. The simulated saturated liquid density was again optimal.

CoSMoS. The search pattern was set to $V^{(1)}$. The metamodels were based on multiquadrics with parameters $\hat{\gamma} = 1.2, d = 0$. Interestingly, nearly half of the first 50 function evaluations failed due to unstable simulation systems. Nevertheless, within only 52 evaluations, CoSMoS found a parameter vector with loss functional value 2.3E-4, x_{CoSMoS}. The subsequent local optimization brought no further improvements. After 109 evaluations, CoSMoS found a different parameter vector with loss functional value 1.9E-4, $x_{\mathrm{CoSMoS,2}} = (0.156, 0.328, 0.34, 0.552, 1.049, 1.823)$. From here, the local optimization improved the loss functional value to 5.0E-5, $x_{\mathrm{CoSMoS+SD}} = (0.156, 0.327, 0.339, 0.552, 1.049, 1.823)$, again, a very small change in parameters. The simulated saturated liquid density was again optimal.

5. Discussion

A variety of six global optimization algorithms were tested for parameterizing molecular force fields. Through systematic test applications, RRS, CMA-ES and CoSMoS were selected to parameterize a molecular force field for phosgene. In combination with a simple local optimizer, they generated three diverse parameter vectors. Each of them was capable to reproduce the saturated liquid density of phosgene at vapor liquid equilibrium over a range of 50 K. However, without subsequent local optimizations, they could only produce almost optimal parameter vectors. This supports the combination of global and local optimization methods to obtain highly accurate force fields.
The detected optimal parameter vectors for phosgene were scattered over a big part of the search space. Obviously, the given parameterization problem was under-determined and thus quite easy to solve. This is not surprising after all, because six parameters were available to reproduce only one observable. Consequently, the optimized parameter vectors may not be transferable to go well with other simulation systems and observables. If less crucial or, as in this case, too many parameters are involved in the global optimization, they may have arbitrary values in the final (optimal) force field. This can lead to force fields that are highly accurate in a specific system but neither transferable nor physically interpretable. In order to prevent such situations, global optimization should incorporate more than one observable and focus on the decisive parameters.
However, it is not always easy to decide a priori which parameters are decisive. In the example of phosgene, the chlorine parameters affect the observables most, because they simply apply to more atoms in the system. Inserting an additional sensitivity analysis before the actual global optimization can help to detect parameters that have little influence, in order to keep them fixed or at least tighten their bounds. For CoSMoS, the generated meta models can be used to study the influences of the individual parameters. This enables the user

to reject or pick parameter sets that have been sampled by CoSMoS, according to a physical reasoning.
In nearly all applications, the newly designed CoSMoS dominated the other algorithms. This is partially due to the specific implementations of the algorithms. There are various implementations of DE and TS methods. The poor performance of DE and TS in this paper can not be generalized to all of them. There are also various implementations of CMA-ES, including a multi-objective version and a version using metamodels. Those might go better with force field parameterization. CoSMoS performed best for search patterns with multiple elements, e.g. $(0.9, 0.75, 0.25, 0.05, 0.03, 0.0)$, and in connection with multiquadric RBFs. The CORS and new sampling strategy worked comparably well for all of the test problems. Especially after the tuning of algorithm parameters, CoSMoS clearly outperformed the other algorithms.
A limitation of CoSMoS is foreshadowed by test problem 11: For optimizing more than ten parameters at once, its efficiency drops. This is rooted in the well-known fact that RBF networks converge slowly in higher-dimensional spaces, i.e. they require more sampling points to reliably reproduce the shape of the objective function. Here, direct search methods like RRS may be superior. Anyway, as discussed before, this has only little consequences for our field of application. In a typical parameterization, we optimize two to six parameters against two or three reference observables at a range of three to seven temperatures. Optimizing ten or more parameters will normally lead to tasks that are either under-determined or involve many unimportant parameters.
Taken together, the present results demonstrate that metamodel-based optimizers particularly suit the given quest. CoSMoS is likely to be efficient for a broad range of different parameterization problems and robust with respect to noise. Using sophisticated metamodeling techniques could possibly further enhance robustness. In the future, CoSMoS will be applied to more complex parameterization tasks.

6. Conclusions

Intermolecular force field parameters are the essence of every molecular simulation. For realistic simulations, they have to be optimized. As classical local optimizers can easily get stuck in local optima or due to noise, global optimization algorithms become an interesting alternative. In this paper, three diverse optimization algorithms were proven feasible for parameterizing force fields: RRS, CMA-ES and CoSMoS. The best results were achieved by combining global and local optimization algorithms.
Our specifically designed algorithm CoSMoS outperformed the other methods on a set of test problems and on parameterizing a molecular model for phosgene. Its particular efficiency is granted by exploiting metamodels of the simulated physical properties. CoSMoS supports parallel simulations, copes with noisy observables and abortive simulations and prevents preliminary convergence to local optima. Taken together, a combination of CoSMoS and GROW may fa-

cilitate the fully automated parameterization of intermolecular potentials for a broad range of molecular systems.

7. Acknowledgments

The authors thank Karl N. Kirschner and Astrid Maaß for many fruitful discussions. The quantum mechanical calculations of intramolecular parameters and partial atomic charges were done by Karl N. Kirschner.

[1] G. Guevara-Carrion, H. Hasse, J. Vrabec, Thermodynamic Properties for Applications in Chemical Industry via Classical Force Fields, in: B. Kirchner, J. Vrabec (Eds.), Multiscale Molecular Methods in Applied Chemistry, volume 307 of *Topics in Current Chemistry*, Springer Berlin Heidelberg, Berlin and Heidelberg, 2012, pp. 201–249.

[2] G. Guevara-Carrion, C. Nieto-Draghi, J. Vrabec, H. Hasse, Prediction of Transport Properties by Molecular Simulation: Methanol and Ethanol and their Mixture, Journal of Phyisical Chemistry B 112 (2008) 16664–16674.

[3] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, W. Wriggers, Atomic-Level Characterization of the Structural Dynamics of Proteins, Science 330 (2010) 341–346.

[4] Q. H. Zeng, A. B. Yu, G. Q. Lu, Multiscale modeling and simulation of polymer nanocomposites, Progress in Polymer Science 33 (2008) 191–269.

[5] Y. Cui, Using molecular simulations to probe pharmaceutical materials, Journal of Pharmaceutical Sciences 100 (2011) 2000–2019.

[6] J. A. Nelder, R. A. Mead, A Simplex Method for Function Minimization, Computer Journal 7 (1965) 308–313.

[7] R. Faller, H. Schmitz, O. Biermann, F. Müller-Plathe, Automatic Parameterization of Force Fields Liquids by Simplex Optimization, Journal of Computational Chemistry 20 (1999) 1009–1017.

[8] S. Girard, F. Müller-Plathe, Molecular dynamics simulation of liquid tetrahydrofuran: on the uniqueness of force fields, Molecular Physics 101 (2003) 779–787.

[9] T. J. Müller, S. Roy, W. Zhao, A. Maaß, D. Reith, Economic simplex optimization for broad range property prediction: Strengths and weaknesses of an automated approach for tailoring of parameters, Fluid Phase Equilibria 274 (2008) 27–35.

[10] P. Ungerer, C. Beauvais, J. Delhommelle, A. Boutin, B. Rousseau, A. H. Fuchs, Optimization of the anisotropic united atoms intermolecular potential for n-alkanes, Journal of Computational Physics 112 (1999) 5499–5510.

[11] E. Bourasseau, M. Haboudou, A. Boutin, A. H. Fuchs, P. Ungerer, New optimization method for intermolecular potentials: Optimization of a new anisotropic united atoms potential for olefins: Prediction of equilibrium properties, Journal of Chemical Physics 118 (2003) 3020–3034.

[12] J. Stoll, Molecular models for the prediction of thermophysical properties of pure fluids and mixtures, volume Nr. 836 of *Fortschritt-Berichte / VDI Verfahrenstechnik*, als ms. gedr ed., VDI-Verl., Düsseldorf, 2005.

[13] T. Merker, J. Vrabec, H. Hasse, Engineering Molecular Models: Efficient Parameterization Procedure and Cyclohexanol as Case Study, Soft Materials 10 (2012) 3–25.

[14] S. Deublein, P. Metzler, J. Vrabec, H. Hasse, Automated development of force fields for the calculation of thermodynamic properties: acetonitrile as a case study, Molecular Simulation 39 (2013) 109–118.

[15] M. Hülsmann, T. Köddermann, J. Vrabec, D. Reith, GROW: A Gradient-based Optimization Workflow for the Automated Development of Molecular Models, Computer Physics Communications 181 (2010) 499–513.

[16] M. Hülsmann, J. Vrabec, A. Maaß, D. Reith, Assessment of Numerical Optimization Algorithms for the Development of Molecular Models, Computer Physics Communications 181 (2010) 887–905.

[17] M. Hülsmann, T. J. Müller, T. Köddermann, D. Reith, Automated force field optimisation of small molecules using a gradient-based workflow package, Molecular Simulation 36 (2010) 1182–1196.

[18] T. Köddermann, D. Reith, R. Ludwig, Comparison of Force Fields on the Basis of Various Model Approaches-How To Design the Best Model for the [C n MIM][NTf 2] Family of Ionic Liquids, ChemPhysChem 14 (2013) 3368–3374.

[19] R. Regis, C. Shoemaker, Constrained global optimization of expensive black box functions using radial basis functions, Journal of Global Optimization 31 (2005) 153–171.

[20] A. Maaß, L. Nikitina, T. Clees, K. N. Kirschner, D. Reith, Multi-objective Optimisation on the Basis of Random Models for Ethylene Oxide, Molecular Simulation 36 (2010) 1208–1218.

[21] C. Bäuerle, C.-A. Thole, U. Trottenberg, DesParO - A Design Parameter Optimisation Toolbox using an Iterative Kriging Algorithm, ERCIM News (2004) 32–33.

[22] M. Hülsmann, Effiziente und neuartige Verfahren zur Optimierung von Kraftfeldparametern bei atomistischen Molekularen Simulationen kondensierter Materie, Ph.D. thesis, Universität zu Köln, Köln, 2012.

[23] A. Krämer, M. Hülsmann, J. Vrabec, D. Reith, Global optimization techniques utilized to generate new accurate VLE force fields for ethylene oxide as a test case, Abstracts of papers of the American Chemical Society 245 (2013) 362–COMP.

[24] W. L. Goffe, G. D. Ferrier, J. Rogers, Global optimization of statistical functions with simulated annealing, Journal of Econometrics 60 (1994) 65–99.

[25] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, Science 220 (1983) 671–680.

[26] T. Bäck, H.-P. Schwefel, An Overview of Evolutionary Algorithms for Parameter Optimization, Evolutionary Computation 1 (1993) 1–23.

[27] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[28] I. C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters 85 (2003) 317–325.

[29] N. Hansen, The CMA-Evolution Strategy: A Tutorial, 2011. URL: `https://www.lri.fr/ hansen/cmatutorial110628.pdf`.

[30] A. M. Connor, D. G. Tilley, A Tabu search method for the optimization of fluid power circuits, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 212 (1998) 373–381.

[31] D. Cvijovic, J. Klinowski, Taboo Search: An Approach to the Multiple Minima Problem, Science 267 (1995) 664–666.

[32] R. H. Leary, Global optimization on funneling landscapes, Journal of Global Optimization 18 (2000) 367–383.

[33] D R Jones, C C Perttunen, B E Stuckmann, Lipschitzian Optimization without the Lipschitz Constant, J. Optim. Theory Appl. 79 (1993) 157–181.

[34] T. G. Kolda, R. M. Lewis, V. Torczon, Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods, SIAM Review 45 (2003) 385–482.

[35] A. R. Conn, K. Scheinberg, L. N. Vicente, Introduction to derivative-free optimization, volume 8 of *MPS-SIAM series on optimization*, Society for Industrial and Applied Mathematics/Mathematical Programming Society, Philadelphia, 2009.

[36] H.-M. Gutmann, A radial basis function method for global optimization, Journal of Global Optimization 19 (2001) 201–227.

[37] R. Regis, C. Shoemaker, Parallel radial basis function methods for the global optimization of expensive functions, European Journal of Operational Research 182 (2007) 514–535.

[38] S. Ucyigitler, M. C. Camurdan, M. Turkay, J. R. Elliott, Optimization Of Transferable Site–Site Potentials Using A Combination Of Stochastic And Gradient Search Algorithms, Industrial & Engineering Chemistry Research (2012).

[39] N. Hansen, A. Ostermeier, Completely Derandomized Self–Adaptation in Evolution Strategies, Evolutionary Computation 9 (2001) 159–195.

[40] Nikolaus Hansen, Stefan Kern, Evaluating the CMA Evolution Strategy on Multimodal Test Functions, in: Xin Yao, Edmund K. Burke, José Antonio Lozano, Jim Smith, Juan J. Merelo Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, Hans-Paul Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference, Birmingham, UK, September 18-22, 2004, Proceedings, volume 3242 of *Lecture Notes in Computer Science*, Springer, 2004, pp. 282–291.

[41] M. Omidvar, X. Li, P. Mugunthan, C. A. Shoemaker, R. G. Regis, A Comparative Study of CMA-ES on Large Scale Global Optimisation: Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models, in: J. Li, J. a. Debenham (Eds.), Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence, volume 41 of *AI'10*, Springer, 2005, p. n/a.

[42] R. Ros, Black-box optimization benchmarking the IPOP-CMA-ES on the noisy testbed: comparison to the BIPOP-CMA-ES, in: Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, GECCO '10, ACM, 2010, pp. 1511–1518. URL: `http://doi.acm.org/10.1145/1830761.1830767`.

[43] D. Karaboğa, S. Ökdem, A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm, Turk J Elec Engin 12 (2004) 53–60.

[44] R. Hooke, T. Jeeves, Direct search solution of numerical and statistical problems, Journal of the Association for Computing Machinery (ACM) 8 (1961) 212–229.

[45] S. Parodat, MARABU - Ein Werkzeug zur Approximation nichtlinearer Kennlinien mit radialen Basisfunktionen, Paderborn, 1997.

[46] P. Mugunthan, C. A. Shoemaker, R. G. Regis, Comparison of function approximation, heuristic, and derivative-based methods for automatic calibration of computationally expensive groundwater bioremediation models, Water Resources Research 41 (2005) n/a.

[47] W. Nan, S. Lincheng, L. Hongfu, C. Jing, H. Tianjiang, Robust optimization of aircraft weapon delivery trajectory using probability collectives and meta-modeling, Chinese Journal of Aeronautics 26 (2013) 423–434.

[48] E. K. Murphy, V. V. Yakovlev, CAD Technique for Microwave Chemistry Reactors with Energy Efficiency Optimized for Different Reactants, APPLIED COMPUTATIONAL ELECTROMAGNETICS SOCIETY JOURNAL 25 (2010) 1108–1117.

[49] J. Stoll, J. Vrabec, H. Hasse, Comprehensive Study of the Vapour–Liquid Equilibria of The Pure Two–Centre Lennard–Jones plus Pointdipole Fluid, Fluid Phase Equilibria 209 (2003) 29–53.

[50] P.J. Linstrom and W.G. Mallard (Ed.), NIST Chemistry WebBook: NIST Standard Reference Database Number 69, 2014. URL: `http://webbook.nist.gov`.

[51] T. A. Runkler, Data analytics: Models and algorithms for intelligent data analysis, Springer Vieweg, Wiesbaden and New York, 2012.

[52] M. F. Guest, I. J. Bush, Van Dam, Huub J. J., P. Sherwood, Thomas, Jens M. H., Van Lenthe, Joop H., Havenith, Remco W. A., J. Kendrick, The GAMESS-UK electronic structure package: algorithms, developments and applications, Molecular Physics 103 (2005) 719–747.

[53] D. Reith, K. N. Kirschner, A Modern Workow for Force-Field Development – Bridging Quantum Mechanics and Atomistic Computational Models, Computer Physics Communications 182 (2011) 2184–2191.

[54] O. Krämer-Fuhrmann, J. Neisius, N. Gehlen, D. Reith, K. N. Kirschner, Wolf2Pack–portal based atomistic force-field development, Journal of chemical information and modeling 53 (2013) 802–808.

[55] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, Berendsen, Herman J. C., GROMACS: Fast, flexible, and free, Journal of Computational Chemistry 26 (2005) 1701–1718.