# Optimizing Molecular Models Through Force-Field Parameterization via the Efficient Combination of Modular Program Packages

**Marco Hülsmann, Karl N. Kirschner, Andreas Krämer, Doron D. Heinrich, Ottmar Krämer-Fuhrmann and Dirk Reith**

**Abstract** A central goal of molecular simulations is to predict physical or chemical properties such that costly and elaborate experiments can be minimized. The reliable generation of molecular models is a critical issue to do so. Hence, striving for semiautomated and fully automated parameterization of entire force fields for molecular simulations, the authors developed several modular program packages in recent years. The programs run with limited user interactions and can be executed in parallel on modern computer clusters. Various interlinked resolutions of molecular modeling are addressed: For intramolecular interactions, a force-field optimization package named $Wolf_2Pack$ has been developed that transfers knowledge gained from quantum mechanics to Newtonian-based molecular models. For intermolecular interactions, especially Lennard–Jones parameters, a modular optimization toolkit of programs and scripts has been created combining global and local optimization algorithms. Global optimization is performed by a tool named CoSMoS, while local optimization is done by the gradient-based optimization workflow named GROW or by a derivative-free method called SpaGrOW. The overall goal of all program packages is to realize an easy, efficient, and user-friendly development of reliable force-field parameters in a reasonable time. The various tools are needed

M. Hülsmann (✉) · A. Krämer · D.D. Heinrich · D. Reith
Department of Mechanical Engineering (EMT) and Institute for Technology, Renewables and Energy Efficiency (TREE), Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany
e-mail: marco.huelsmann@h-brs.de; marco.huelsmann@scai.fraunhofer.de

M. Hülsmann · O. Krämer-Fuhrmann · D. Reith
Department of Simulation Engineering, Fraunhofer-Institute for Algorithms and Scientific Computing (SCAI), Schloss Birlinghoven, 53757 Sankt Augustin, Germany

K.N. Kirschner
Department of Computer Science and the Institute of Visual Computing (IVC), Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany



R.Q. Snurr et al. (eds.), *Foundations of Molecular Modeling and Simulation*, Molecular Modeling and Simulation, DOI 10.1007/978-981-10-1128-3_4

and interlinked since different stages of the optimization process demand different courses of action. In this paper, the conception of all programs involved is presented and how they communicate with each other.

**Keywords** Molecular modeling · Force field · Numerical optimization · High-performance computing · Modular software packages

# 1 Introduction

## *1.1 Molecular Simulation and Its Tools*

Molecular simulation methods, most prominently molecular dynamics (MD) and Monte Carlo (MC), are powerful tools to gain insight into microscopic processes that govern the macroscopic behavior of matter. There is a long-standing tradition of studying molecular behavior for biomolecules (e.g., proteins, DNA, and carbohydrates) and for soft materials (e.g., plastics, fibers, carbon nanotubes, and ionic liquids). This is reflected by a long history of parameter and software development in this area, which is often distributed together as a collection of predefined parameters, molecular building blocks, and a simulation engine. However, in recent years, significant algorithmic progress has been made to enhance molecular simulation and analysis. There is a widespread utilization of GPUs in existing software packages (e.g., *Amber* [1], *Charmm* [2], *Gromacs* [3], and *LAMMPS* [4]) and automated procedures to derive force-field parameters [5, 6]. In addition, recent coarse-grained methods that access the mesoscale introduced new powerful scientific concepts to the field of molecular simulations (e.g., *HOOMD* [7], *ESPResSo++* [8], and *IBIsCO* [9]).

To gain a molecular-level understanding, chemical systems are modeled at atomistic or near atomistic (e.g., united atom, fine coarse graining) resolution levels. Since computable properties obey the laws of statistical physics, an ensemble of several ten thousands of atoms is necessary to compute the macroscopic observables. Furthermore, modern industrially relevant systems (e.g., chemically heterogeneous, surfaces, mixed phase states) require large models for accurate representations. This results in the necessity to implement the calculations in high-performance computing environments. Driven by the ongoing growth in computational power, it can be expected that these molecular methods will be increasingly useful in the coming decades.

One goal of our research is to provide a computational modeling service to external researchers, both in industry and academics, who wish to obtain a molecular understanding of their systems. As such, we have been faced with using, modifying, and optimizing all atom, united atom, and coarse-grained force fields for natural products, polymers, lipids, ionic liquids, and organic solvents. While the technique of molecular simulations has existed for decades and in spite of its obvious powers, only a few companies have in-house departments, that is due to

(a) the diversity of knowledge needed to do high-quality research (i.e., the method's core is mathematics and physics, the content is often being chemical, and the technical aspects require computational scientists) and (b) the high-performance hardware that is required to execute the simulation software.

## 1.2 Force Fields

One key requirement in molecular mechanics (MM)-based models is the need to be as accurate as possible. This accuracy is directly dependent upon the force field, which describes the intra- and intermolecular interactions. Force fields are a semiempirical approach to represent these interactions—that is a set of equations and associated parameters that model stretching, bending, internal rotations, van der Waals, and Coulombic interactions. In general, there is a consensus on what function form of the equations should be used. Coupled directly to the equations are the parameters, whose optimization is very important but often tedious to accomplish.

Over the past decades, many researchers have developed force fields for a variety of areas, such as thermodynamic properties of fluids [10–15], mechanic properties of solids [16–18], phase change phenomena [19–21], protein folding [22–24], transport processes in biological tissue [25, 26], transport processes in liquids [27–29], polymer properties using different length scales [30–33], and generic statistic properties of soft matter [34]. Some of these force fields have been molecule specific, while others have been transferable over a chemical class (e.g., hydrocarbons, alcohols). For our models, the criterion is that they accurately reproduce or predict the relevant observable(s) using the modeling software that is most appropriate for the investigation. Quantum mechanical methods are useful to determine some of the target observables used in parameter fitting (i.e., geometry, electrostatics, relative energies). However, weak short-range nonbonded interactions are difficult to isolate target quantum mechanical observables, particularly when the molecules are composed of heterogeneous atom types. Hence, the force-field parameters for these weak interactions are often fitted to experimental condense-phase target values. Thus, a manual parameter adjustment is usually not feasible or is, at best, extremely time-consuming.

## 1.3 Goal of This Work

What has become clear is that a user-friendly and versatile software package, which facilitates the optimization of force-field parameters for a given MM or MD engine, is very important. Hence, automated and semiautomated parameterization process can reduce the time required for optimization and subsequently allow researchers more time to explore their ideas. We contribute to this field by creating modular

software packages that follow our ideas for force-field development and by efficiently and systematically combining these programs for the (semi)automated optimization of bonded and nonbonded parameters.

The benefits of utilizing scientific workflows are numerous, and they represent a major improvement in how one approaches force-field development. These benefits include (a) saving time by automating certain optimization tasks; (b) making force-field development quasi-deterministic; (c) reducing human error; (d) enabling tasks to be executed in a distributed environment; (e) accommodating ideas, algorithmic changes, and updates easier; and finally (f) accelerating and transforming the process of scientific analysis. From a scientific perspective, workflows enable researchers to focus more on scientific issues, and due to its hierarchical organization, new advancement in theories can be easily incorporated. In addition to this, errors within the force field and models are better avoided, making the simulation results become more trustworthy and reliable. Moreover, the algorithms involved within the workflow can handle overdetermined and underdetermined optimization problems. From a community service perspective, our workflows significantly reduce the real time needed for force-field development and allow nonspecialists access to more standardized optimization procedures.

For the determination of the intramolecular parameters, we developed a tool named $Wolf_2Pack$, and for the intermolecular parameters, we use a combination of a global optimization procedure with a local one. For the former, we developed a global optimization tool named CoSMoS, and for the latter, we developed a gradient-based optimization toolkit named GROW and a derivative-free sparse grid-based algorithm named SpaGrOW. The three tools are described in more detail in the next subsections.

## 2 Goal-Driven Software Conception

### 2.1 *$Wolf_2Pack$: Intramolecular Parameters*

The concept for $Wolf_2Pack$[1] came from our goals to have a tool that would

(a) allow for quick optimization of bonded parameters,
(b) enable one to qualify observed MD structural results,
(c) allow one to evaluate existing force fields,
(d) allow for the systematic generation and archiving of QM target data for reuse,
(e) enable nonforce-field experts the opportunity to generate their own parameters, and
(f) enable reproducibility of reported force-field research results (e.g., molecule-specific QM and MM energy curves).

[1] http://www.wolf2pack.com.

To achieve these goals, a scientific workflow was developed that provided a guiding architecture for software development [35]. Each step of the workflow was realized through shell scripts, whose output data are organized, as illustrated in Fig. 1, into subdirectories. This modular construct has the advantage that individual scripts can be easily updated, discovered errors in the scripts and generated data can be efficiently corrected, and the generated data are organized in a systematic manner that easily allow for the inclusion of new computations, archiving, and reuse.

To enable nonforce-field experts the chance to check and optimize parameters, a Web site was created that serves as a front-end to $Wolf_2Pack$ [36]. This Web site guides users in the parameter optimization process, starting from selecting an appropriate molecule to the determination of a suitable parameter. The site also provides a collection of "Knowledge Modules" that are a combination of tutorials and examples. Currently, the Web site only provides access to a truncated amount of the existing data within the $Wolf_2Pack$'s database. In the near future, we intend to provide users' access to the full database and enable them to upload a molecule and compute the QM curves that they desire.

An important component of $Wolf_2Pack$ is its molecular database. The database contains molecules of diverse chemical functionalities for which bond, angle, and torsion relative energies curves have been generated. This database naturally grows over time as new functional groups and combinations thereof are investigated. Thus, the statistical evaluation of force fields improves as the database expands. Due to its systematic development, the database also enables users to reproduce results in published force-field papers, which is currently a difficult task to accomplish. We believe this will become an important feature in the future as users make use of $Wolf_2Pack$ for optimizing parameters. The challenge will be to continually update the database for the new QM theories that are reported in the

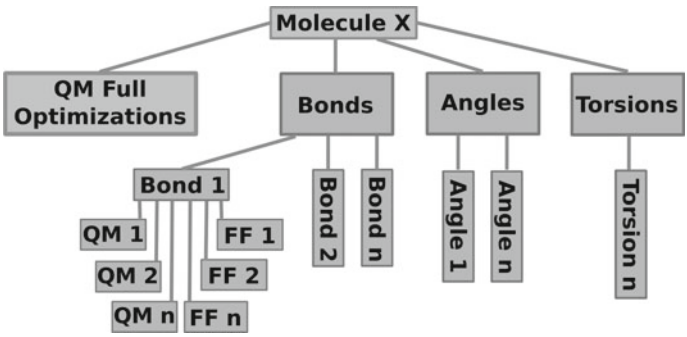


**Fig. 1** Illustration of the basic directory structure within $Wolf_2Pack$. Each molecule with a given conformation has its own parent directory. The *number* of bond, angle, and torsion subdirectories is dependent upon the molecule's unique internal coordinates. The "QM n" and "FF n" labels indicate data from constraint QM and MM optimizations using a specific theory level (e.g., HF/6-31G(d)//HF/6-31G(d)) or force field (e.g., Parm14SB)

literature, which will be an increasingly demanding task as the number of molecules and internal coordinates grow.

Considering parameterization philosophy, we are pursuing new ideas in addition to the traditional fitting of continuous relative potential energy curves. Through the assistance of the Balloon algorithm [37], $Wolf_2Pack$ can quantum mechanically generate and identify unique conformations automatically. For illustration, we recently predicted 76 unique octane conformations at the HF/6-31G(d) using Balloon and $Wolf_2Pack$ algorithms. While this does not represent the complete set of unique octane conformations, which have been determined to be 95 [38], it does impressively cover a wide range of relative energies (0.0–8.9 kcal/mol). These high numbers of conformations for a flexible molecule allow for a unique way to validate force fields. Traditionally, nonbonded and bonded force-field terms are optimized by reproducing experimental observables (e.g., density) and relative energy curves (i.e., transition states, minima), which rarely consider more than a few high energy minima. By having access to a large number of minima, one can observe how a given force field's parameters transfer to higher energy minima and conformations not originally considered during the optimization process.

Researchers usually strive to generate continuous QM rotational energy curves. A continuous curve is one whose incremented internal coordinate changes, while all other unconstrained torsion angles remain in their original position (e.g., within ±5°). The advantage of this is that the obtained relative energies directly reflect the rotation around a single bond. The subsequent parameter optimization is then fairly straightforward. A discontinuous rotational curve would be when a second torsion undergoes significant rotation at some point during the interested torsion rotation (e.g., Fig. 2). The resulting energy curve then reflects contribution from changes

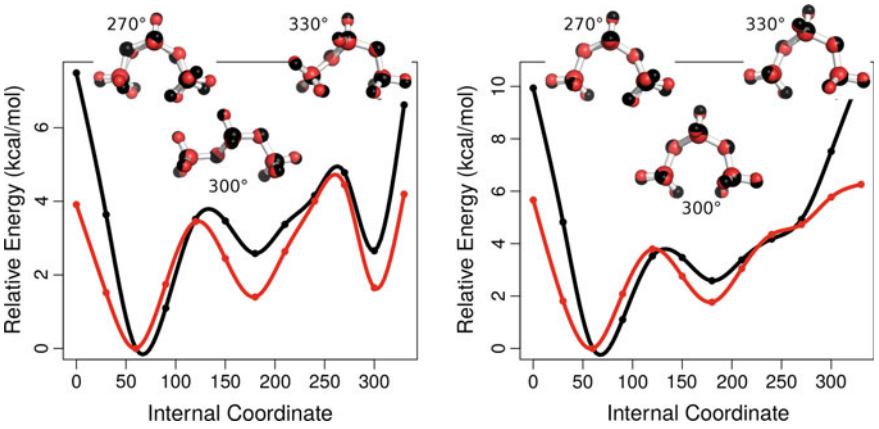


**Fig. 2** Potential energy curves and geometric overlays for dimethoxymethane as determined by HF/6-31G(d) (*red*) and the Gaff (*black*) force field. In this case, the C–C–O–C torsion on the *left side* of the molecule is systematically rotated. The *left image* shows the discontinuous curve where the *right side* C–O–C–C adopted a transconformation at 300°, while the *right image* shows the continuous curve. The continuous curve was generated by constraining the mobile torsion

within two torsion angles, making parameter optimization more convoluted. In $Wolf_2Pack$, we strive to generate continuous curves and will apply a secondary torsion constraint if necessary to obtain one for parameter optimization purposes. Nevertheless, we also make use of the discontinuous curves that are produced for testing the robustness of the optimized parameters. Fundamentally, the discontinuous curve represents significant coupling between internal coordinates, for which force fields should ideally reproduce. We believe that reproduction of discontinuous curves is a more rigorous test of a force field's performance in comparison with the reproduction continuous curves. In addition to investigated torsion angles, discontinuous curves also occur when generating bond stretching and angle bending energy profiles. Typically, a close contact occurs between atoms, resulting in the rotation about a bond to relieve the high energy strain.

## 2.2 *CoSMoS, GROW, and SpaGrOW: Intermolecular Parameters*

The optimization of nonbonded parameters is difficult since one can rarely isolate the parameters for a specific atom type, with the notable exception of the noble gases. If one considers simple saturated hydrocarbons, the carbon and hydrogen Lennard–Jones parameters are often optimized simultaneously. This results in a large possible parameter space, making an a priori understanding of the loss function's shape impossible. For this reason, as illustrated in Fig. 3, we have developed both global (i.e.,

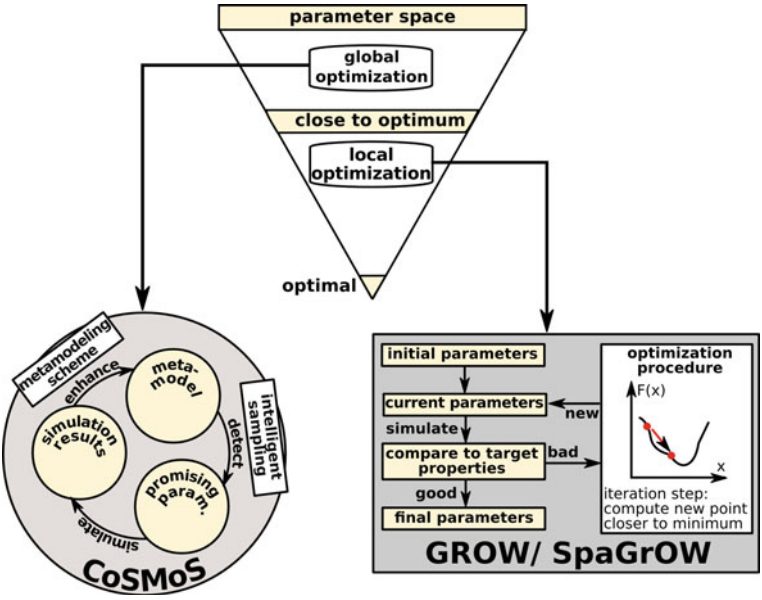


**Fig. 3** The funnel workflow approach for optimizing nonbonded parameters

CoSMoS) and local (i.e., GROW and SpaGrOW) tools that are implemented in a funnel workflow. CoSMoS is based on metamodeling that enables rough identification of potential optimal values, while either a gradient-based (GROW) or derivative-free (SpaGrOW) approach is used to refine the identified parameters.

In the last two decades, substantial research occurred for the optimization of intermolecular force-field parameters [39–54]. In most cases, intermolecular parameters, especially Lennard–Jones parameters, cannot be strictly derived via physical considerations since they parameterize semiempirical models (i.e., based on classical mechanics) whom themselves only approximate reality. Hence, they are usually adjusted so that the resulting model is able to reproduce physical or chemical experimental target properties as accurately as possible.

The overall optimization task is to find a solution to the following mathematical optimization problem:

$$\min_{x\in\Omega} F(x) := \left\| W(f^{\text{sim}}(x) - f^{\text{exp}}) \right\|_p^2,\ p \in [1,\infty], \tag{1}$$

where $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$ is a vector consisting of the force-field parameters to be adjusted, $N \in \mathbb{N}$ is the number of parameters, $n \in \mathbb{N}$ is the number of physical properties to be fitted, $f^{\text{sim}}(x) \in \mathbb{R}^n$ is the vector containing all properties calculated by simulation, $f_i^{\text{sim}}$, $i = 1, \ldots, m$, and $f^{\text{exp}} \in \mathbb{R}^n$ is the vector containing the experimental target values $f_i^{\text{exp}}$, $i = 1, \ldots, m$. For reasons of brevity, $\|\cdot\|$ indicates an arbitrary $p \in [1, \infty]$. If a particular norm is considered, this will be expressed explicitly (e.g., $\|\cdot\|_2$ or $\|\cdot\|_\infty$). The weighting matrix is defined as:

$$W = \begin{pmatrix} \frac{w_1}{f_1^{\text{exp}}} & 0 & \cdots & 0 \\ 0 & \frac{w_2}{f_2^{\text{exp}}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{w_n}{f_n^{\text{exp}}} \end{pmatrix} \tag{2}$$

with specific weights $w_i,\ i = 1, \ldots, n$, for each property, accounting for the fact that some properties may be easier to reproduce than others due to statistical noise on both simulation and experimental data. The loss function $F(x)$ has to be minimized with respect to $x$ within an admissible domain $\Omega \subset \mathbb{R}^N$. Hence, the optimization problem is constrained.

The loss function does not have any analytical form with respect to the force-field parameters, and the simulated properties are affected by statistical noise. Hence, it cannot be assumed to be smooth or differentiable. Its shape is not known a priori and is often jagged in real applications. Moreover, as the optimization problem may be overdetermined, the loss function may form a rain drain, where many global optima are located at the bottom. Additionally, the evaluations of the loss function may be costly, in particular if molecular simulations have to be performed. For all these reasons, the solution of the optimization problem (1) is

challenging and not possible using standard line-search methods. In order to jump over intermediate local minima, an efficient global optimization that focuses into a close neighborhood of the global minimum is indispensable. Mostly, global optimization algorithms get stuck at a certain iteration because the points in the parameter space are generated via random sampling methods. In this case, local optimization procedures are more reliable and faster because they are directed to the minimum, especially when they are gradient based. Hence, the combination of global with local optimization algorithms turned out to be much more reliable and efficient in order to solve the present optimization task than the usage of a single global or local algorithm [55].

## 2.3 Methodological Aspects of CoSMoS

The recently developed global optimization tool for the Calibration of molecular force fields by Simultaneous Modeling of Simulated data (CoSMoS) [56] uses a metamodeling procedure based on radial basis functions (RBFs). It has been shown in [56] that metamodel-based optimizers particularly suit the quest for quickly finding nearly optimal force-field parameters. The metamodels constructed by CoSMoS describe functional dependencies between the force-field parameters and the relative deviations of the simulated properties to experimental data so that the minimization task is easier to solve. The RBFs are rational symmetric functions $\Phi : \mathbb{R}^N \to \mathbb{R}$ of the form $\Phi(x) = \Phi(\|x\|)$ for $x \in \mathbb{R}^N$. For the present optimization problem, inverse multiquadric RBFs, i.e., $\Phi(x) = (\|x\|^2 + \gamma^2)^{-\frac{1}{2}},\ \gamma \in \mathbb{R}$, turned out to perform best. However, CoSMoS also offers the possibility to use other RBFs, e.g., cubic $\left(\Phi(x) = \|x\|^3\right)$ and Gaussian $(\Phi(x) = \exp(-(\gamma\|x\|)^2))$ functions, thin-plate splines $(\Phi(x) = \|x\|^2 \log\|x\|)$, or multiquadrics $\left(\Phi(x) = \sqrt{||x||^2 + \gamma^2}\right)$. The metamodel $\mathcal{M}^v(x)$ interpolating a target property $v \in \{1, \ldots, n\}$ is then given by

$$\mathcal{M}^v(x) = \sum_{j=1}^{q} \alpha_j^v \Phi(\left\|x - x_j\right\|) + \sum_{k=1}^{r} \beta_k^v p_k(x), \tag{3}$$

where $x_j,\ j = 1, \ldots, q,\ q \in \mathbb{N}$ are sampling points that fulfill the interpolation condition $\mathcal{M}^v(x_j) = f_v^{\text{sim}}(x_j),\ j = 1, \ldots, q$. The $p_k(x),\ k = 1, \ldots, r,\ r \in \mathbb{N}$ are low-order polynomials, and the coefficients $\alpha_j^v \in \mathbb{R},\ j = 1, \ldots, q,\ v = 1, \ldots, n$ and $\beta_k^v \in \mathbb{R},\ k = 1, \ldots, r, v = 1, \ldots, n$ are obtained by solving a linear equation system (LES): The radial basis function matrix of the sampling points is given by $H = (H)_{li} := (\Phi(||x_l - x_i||))_{l,i=1,\ldots,q} \in \mathbb{R}^{q \times q}$, and the polynomial matrix is given by $P := (P)_{lk} = p_k(x_l)_{l=1,\ldots,q,k=1,\ldots,r} \in \mathbb{R}^{q \times r}$. The right hand side is as follows:

$$d_v^{\text{sim}} := (d_v^{\text{sim}})_l = \left( \frac{f_v^{\text{sim}}(x_l) - f_v^{\text{exp}}}{s_v^{\text{sim}} f_v^{\text{exp}}} \right)_{l=1,\ldots,q}, \tag{4}$$

where $s_v^{\text{sim}}, v \in \{1, \ldots, n\}$ is the standard deviation of the relative noise of the property $v$. Hence, the following linear equation system (LES) has to be solved:

$$\begin{pmatrix} \mathrm{H} & \mathrm{P} \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \alpha^v \\ \beta^v \end{pmatrix} = \begin{pmatrix} d^{\text{sim}} \\ 0 \end{pmatrix}, \tag{5}$$

where $\begin{pmatrix} \alpha^v \\ \beta^v \end{pmatrix}$ is the vector containing the coefficients $\alpha_j^v \in \mathbb{R},\ j = 1, \ldots, q,\ v = 1, \ldots, n$, and $\beta_k^v \in \mathbb{R},\ k = 1, \ldots, r,\ v = 1, \ldots, n$. The second line mirrors an additional orthogonality to render the coefficients unique. However, this procedure may lead to large RBF coefficients, resulting in wavy metamodels that do not reflect the underlying data properly. This is particularly severe for noisy data, which demands proper smoothing approaches. Thus, in this work, CoSMoS was extended by two different smoothing methods: The *smoothest* metamodel is the one with the smallest RBF coefficients, which can be calculated by solving

$$\min_{\alpha^v} \quad \|\alpha^v\|^2, \tag{6}$$

$$\text{where} \quad f_l^{\text{sim}} - \xi \leq b_l \leq f_l^{\text{sim}} + \xi, l = 1, \ldots, q, \tag{7}$$

where $\xi > 0$ is a small tolerance value, and $b$ is the vector $(\mathrm{H} \quad \mathrm{P}) \begin{pmatrix} \alpha^v \\ \beta^v \end{pmatrix}$. As the statistical noise is taken into account by the method due to Eq. (4), confidence intervals are drawn around the sampling points so that overfitting can be avoided during interpolation. Hence, the method searches for metamodels which are as smooth as possible.

The *weighted* smoothing method tries to find a compromise between the two contradictory requirements of high smoothness and low smoothing error. This compromise is controlled via an additional weighting parameter $\chi > 0$, and the following constrained minimization problem is solved:

$$\min_{\alpha^v, \beta^v} \left\| (\mathrm{H} \quad \mathrm{P}) \begin{pmatrix} \alpha^v \\ \beta^v \end{pmatrix} - d_v^{\text{sim}} \right\|^2 + \chi \|\alpha^v\|^2, \tag{8}$$

which is equivalent to solving the LES

$$\begin{pmatrix} H^T H + \chi I & H^T P \\ P^T H & P^T P \end{pmatrix} \begin{pmatrix} \alpha^v \\ \beta^v \end{pmatrix} = \begin{pmatrix} H^T d_v^{\text{sim}} \\ P^T d_v^{\text{sim}} \end{pmatrix}. \tag{9}$$

An optimal choice of $\chi$ would lead to a perfect metamodel fulfilling both criteria. However, the parameter is problem-dependent and thus difficult to optimize in practice.

Furthermore, CoSMoS provides an intelligent sampling procedure extending the approach of the Constrained Optimization using Response Surfaces (CORS) [57]. The latter focuses the sampling onto potentially optimal regions, avoiding previously sampled regions. This neighborhood is a ball around a sampling point $x \in \tilde{\Omega}$, where $\tilde{\Omega} \subset \Omega$ is the set of the already sampled points, of radius

$$r < \delta_{\tilde{\Omega}}^{\max} := \max_{x \in \Omega} \min_{\tilde{x} \in \tilde{\Omega}} \|x - \tilde{x}\|. \tag{10}$$

This taboo search approach is then realized by solving the constrained minimization problems:

$$\min_{x \in \Omega} \quad ||W \cdot \mathcal{M}^v(x)||, \tag{11}$$

$$\text{where} \quad x \in \bigcup_{\tilde{x} \in \tilde{\Omega}} U_r(\tilde{x}), \quad v = 1, \ldots, n. \tag{12}$$

CoSMoS extends this approach by introducing a penalty term

$$p(x) := \frac{\delta_{\tilde{\Omega}}^{\max}}{\min_{\tilde{x} \in \tilde{\Omega}} \|x - \tilde{x}\|} \geq 1, \tag{13}$$

which grows to infinity, whenever $x$ approaches a sampling point. In contrast to CORS, CoSMoS minimizes the penalized metamodels

$$\tau_{\tilde{\gamma}}^v(x) := p(x)^{\tilde{\gamma}}(\mathcal{M}^v(x) - c), v = 1, \ldots, n. \tag{14}$$

where $\tilde{\gamma}$ and $c$ are control parameters. For more algorithmic details, see reference [56]. Figure 4 demonstrates the adaptive nature of the intelligent sampling strategy. The plot shows a preliminary metamodel after 20 evaluations (right) compared to the actual loss function (left). The metamodel generally captures the optimal region of the loss function, i.e., the vicinity of the minimum. The intelligent sampling strategy takes advantage of this and preferably samples points in the optimal region. In return, each function evaluation further improves the accuracy of the metamodel, improving the sketch of the optimal region. This circular procedure within CoSMoS, which is also depicted in Fig. 3, reduces the number of required simulations and thus the time-to-solution substantially.

An additional advantage of CoSMoS is the fact that it can handle abortive simulations. Whenever a simulation goes wrong due to a bad selection of the force-field parameters, the corresponding sampling points are penalized in the same way so that they are not triggered anymore by the sampling algorithm. Within one

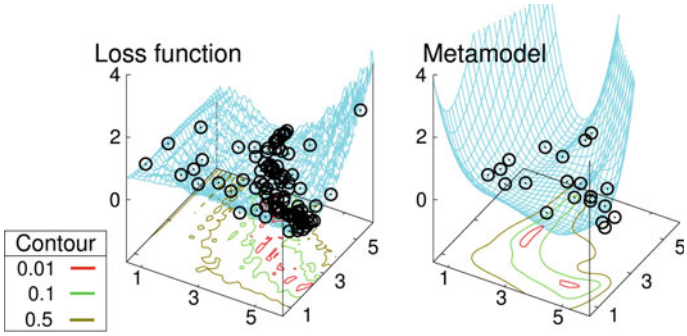


**Fig. 4** *Left* The original loss function for a test problem is shown. The *black points*, sampled by CoSMoS, adapt the shape of the loss function. *Right* The metamodel of the loss function after 20 CoSMoS iterations is depicted, with the first 20 sampling points

CoSMoS iteration, all belonging sampling points are evaluated in parallel via a simple job threading.

## 2.4 Methodological Aspects of GROW

The GRadient-based Optimization Workflow (GROW) [58] explicitly considers the euclidean norm for the loss function in Eq. (1). GROW is a collection of gradient-based numerical optimization algorithms (e.g., steepest descent, conjugate gradients, and trust region) combined with an efficient Armijo step length control. The latter prevents GROW from both jumping over the minimum and leaving the admissible domain of the force-field parameters. For more details of the algorithms involved in GROW, see Ref. [59].

The gradient at an iteration $x \in \Omega$ is given by the partial derivatives

$$\frac{\partial F}{\partial x_j}(x) = -2\sum_{i=1}^{n} w_i \frac{f_i^{\text{exp}} - f_i^{\text{sim}}(x)}{\left(f_i^{\text{exp}}\right)^2} \frac{\partial f_i^{\text{sim}}}{\partial x_j}(x),\ j = 1, \ldots, N.$$

The partial derivatives of the properties are approximated numerically by

$$\frac{\partial f_i^{\text{sim}}}{\partial x_j}(x) = \frac{f_i^{\text{sim}}(x_1, \ldots, x_j + h, \ldots, x_N) - f_i^{\text{sim}}(x)}{h},\ h > 0, j = 1, \ldots, N.$$

On the one hand, due to the statistical uncertainties on the simulated properties $f_i^{\text{sim}}(x)$, GROW can get stuck in an intermediate local minimum caused by the noise, if the discretization parameter $h$ is chosen too small. On the other hand, if $h$ is too large, the estimations of the gradient might be incorrect. Hence, a good compromise has to be found, and the choice of $h$ is problem-dependent and thus difficult

to optimize in practice. However, GROW turned out to be very successful for the parameterization of force fields in many applications [55, 60–62]. For more algorithmic details concerning GROW, see reference [58].

Local optimization procedures always start with an initial guess $x^0 \in \Omega$, which must be situated in the sphere of influence of the minimum. By evaluating the loss function, the simulated properties are compared with the experimental target data. If a specified stopping criterion is fulfilled, the parameters are final and the workflow ends. Otherwise, for the current iteration $x^k \in \Omega, \quad k \in \mathbb{N}$, GROW searches for a iteration $x^{k+1} \in \Omega$ with a lower loss function. At each iteration, a gradient has to be calculated, whose components are evaluated in parallel together with the original iteration $x^k$. Note that the force-field parameters for the gradient components are the same as in $x^k$ except for one component which deviates by $h$ from the original one. Hence, at each iteration, $N + 1$ loss function evaluations are parallelized. The Armijo steps are parallelized as well. For each job, time-consuming molecular simulations are required, and parallelization of these simulations reduces the real computation time significantly. Another approach to reduce computational effort consists in efficient gradient computations, which do not require new function evaluations. This is achieved by computing directional derivatives instead of the partial derivatives so that previously performed loss function evaluations can be used again. The same approach can be applied to Hessians (i.e., for the trust region) method as well [63, 64].

The stopping criterion depends on the specific properties to be fitted. For example, if the density deviates by less than 0.5 % from experiment, the corresponding force field is considered as optimal because the experiment is not more accurate either. The same holds for all other properties. However, the experimental accuracy is much lower for transport properties like diffusion coefficients or viscosity.

## 2.5 SpaGrOW as an Enhanced GROW-Alternative

The Sparse Grid-based Optimization Workflow (SpaGrOW) [65] counteracts the drawbacks of local gradient-based optimization mentioned above. It approximates the loss function near the minimum and filters out the statistical noise by regularization methods using naive elastic nets [66]. In order to reduce the computational effort, this approximation is performed on sparse grids [67], meaning that simulations only have to be performed for sparse grid points. As sparse grids are fully occupied at their boundary, transformations onto the unit hypercube is performed, followed by multiplications of the loss function values with sine functions so that they vanish at the boundary and no simulation has to be performed. Afterward, interpolations from sparse to full grids are performed via a combination technique [68], and the loss function is discretely minimized on the resulting full grids.

The integrated trust region approach [59] makes SpaGrOW an iterative procedure: At each iteration, the loss function is considered on a trust region of a certain size. It must be large enough in order to increase the speed of convergence and to distinguish different loss function values despite the statistical noise, and it must be small enough such that the loss function can be reproduced accurately by the sparse grid interpolations. The discrete minimum of the model on the full grid is compared to the corresponding original loss function value. If both coincide well, then the trust region is increased, if not then it is decreased. Due to the grid-based approach, SpaGrOW is able to find a much more direct path to the minimum than GROW. The practical proof that SpaGrOW is able to outperform gradient-based methods for the present optimization task and all algorithmic details can be found in reference [65].

Note that the loss function evaluations for the different sparse grid points are independent from each other. Hence, they are evaluated in parallel like the gradient components within GROW. Due to its derivative-free approach and due to the fact that it leads more directly to the optimum, SpaGrOW is always preferred to GROW within the funnel workflow. However, one or two steepest descent directions may also be reliable after the CoSMoS's global optimization, leading to faster force-field parameters with a lower loss function value. Moreover, SpaGrOW is not suitable for high-dimensional problems due to the involved smoothing and interpolation procedures, whose computation effort increases exponentially with the dimension.

## 3 Software Realization

### *3.1 Wolf$_2$Pack*

Wolf$_2$Pack is a software package that uses a series of shell scripts that interlink already existing and specialized software (e.g., for computing QM data, statistical analysis, visualization). It enables researchers to optimize intramolecular parameters by fitting to target QM data (i.e., relative energies and geometries) [35, 36]. The QM theories that are possible for generating target data include HF, B3LYP, MP2, AM1, and PM3, while both basis sets proposed by Pople [69] (e.g., 6-31G (d)) and correlation consistent [70] (e.g., aug-cc-pVDZ) basis sets can be specified to describe the orbital space. Currently, *Amber* force fields are available (i.e., Parm14SB [71], Gaff [72], Glycam06j [52], and Lipid14 [73]), as well as our own force field (ExTrM) that is continually being refined and extended.

Parameters optimization can be done using an algorithm or by hand in an iterative process. Several algorithms already exist for intramolecular parameter optimization [1, 6, 53, 74–82]. Currently, we have integrated the algorithms published in Refs. [78, 79]. However, Wolf$_2$Pack strongly encourages the user to perform the optimization by hand in an iterative manner. Doing so allows the users to explore the parameter space and thus build their intuition of how the parameters influence

the resulting curves. With gained experience, one can better decide the importance of specific parameters (e.g., a $V_3$ term in HC–CT–CT–HC), which ones have little influence on given energy curves. For example, an optimization algorithm may determine nonzero values for torsions $V_1$, $V_2$, and $V_3$, while during a manual adjustment, the user observes that the $V_2$ has little effect on the resulting fit. In such a case, setting the $V_2$ to zero should lead to an increase in the parameter transferability over diverse molecules. And due to $Wolf_2Pack$'s molecular database, such a transferability test can be done easily.

Within $Wolf_2Pack$, all QM calculations are performed by *GAMESS* [83], while all MM calculations are performed by *AmberTools* [1] (i.e., *Sander*). Partial atomic charges are determined using *R.E.D.* [54]. File format conversions are executed using *OpenBabel* [84] and shell scripts. Statistical analysis and image generation are done using *Ptraj* [1], *R statistical language* [85], and *pymol* [86]. LATEX typesetting language, with the graphics and animate packages sourced, is used to generate PDF documents with embedded images of relative energy curves and animations that display an overlay of the resulting QM and MM geometries of each conformation [87]. These PDF files serve to archive the final data and allow for easy dissemination of the results to other researchers.

## 3.2 CoSMoS, GROW, and SpaGrOW

CoSMoS, GROW, and SpaGrOW are integrated into a fully modular program structure. The program is implemented in a generic manner such that modules can be easily exchanged. This modular structure allows a developer to easily exchange the optimization algorithm, the optimization problem, the objective function, and the constraints. An interface to a new simulation tool can also be easily implemented. The overall structure is object-oriented and easy to extend. All three tools are written in *python* (version 2.6.6). The program is categorized into the following four layers, whereas the first two layers are related to general optimization problems and the last two are related to the execution of molecular simulations:

- Generic Optimization,
- Force-Field Parameterization,
- Parallel Jobs, and
- Simulation.

As shown in Fig. 5, each layer considers two independent optimization sections: the Solver and the Problem Formulation section. The former regards the optimization algorithm itself, while the latter regards the evaluation of the objective function (i.e., the function to be minimized and the constraints). Within the Generic Optimization layer, there are two abstract upper classes, which are the *OptimizationAlgorithm* and *OptimizationProblem* in the Solver and Problem Formulation sections. These two classes are connected in the sense that the *OptimizationAlgorithm* requires a defined

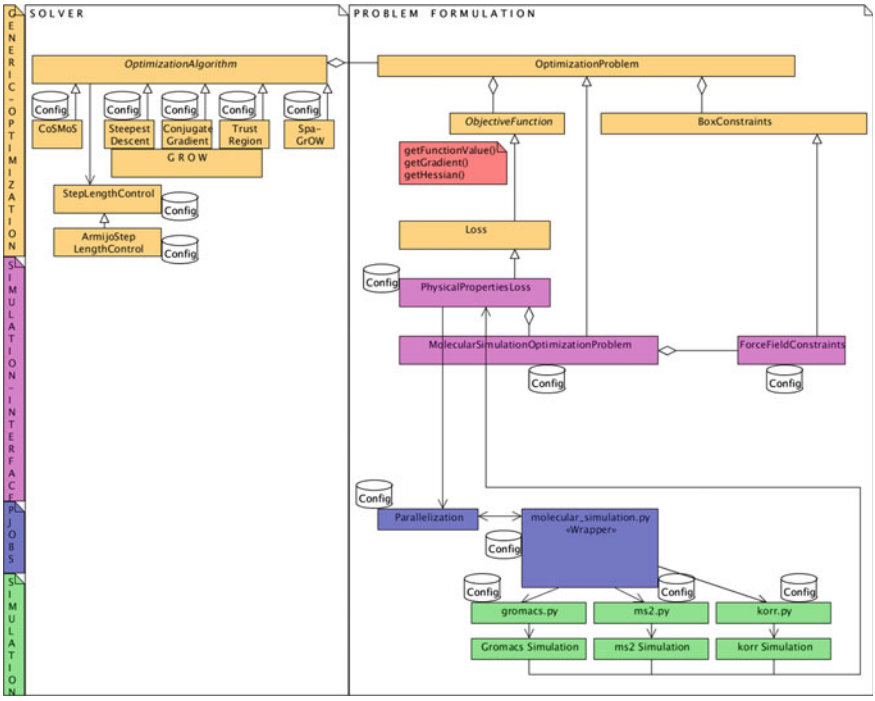


**Fig. 5** Generic modular structure of the overall intermolecular optimization toolbox consisting of the abstract layer Generic Optimization and the three specific layers Force-Field (FF) Parameterization, Parallel Jobs (PJOBS), and Simulation. Most of the modules require input parameters, which are defined in the configuration file (i.e., "Config")

problem to solve from *OptimizationProblem*. For *OptimizationProblem,* it is irrelevant which optimization algorithm is used to solve the optimization problem.

Within the Solver section, the class *OptimizationAlgorithm* defines an object of the class *StepLengthControl*, which steers the step length control. The specific class *ArmijoStepLengthControl* is derived from it and can be exchanged by another step length control method other than Armijo. The CoSMoS, GROW, and SpaGrOW algorithms are steered by specific child classes derived from *OptimizationAlgorithm*. GROW itself encompasses the classes *SteepestDescent*, *ConjugateGradients*, and *TrustRegion*.

The optimization problem for *OptimizationAlgorithm* is defined within the Problem Formulation as an objective function to be minimized and box constraints to be met, which are represented by abstract classes *ObjectiveFunction* and *BoxConstraints*. These two classes contain getter and setter functions (e.g., for the function value, the gradient, the Hessian), which have to be overwritten by specific derived child classes in the layer Force-Field Parameterization. A generic loss function class (i.e., *Loss*) is derived from *ObjectiveFunction* implementing a general loss function between calculated and target values (Eq. 1). Its child class *PhysicalPropertiesLoss* steers the molecular simulations

that are executed in parallel and collects the simulation results. This module interacts with a wrapper script for the molecular simulation steering calling specific *python* scripts for the desired simulation tools. Currently, interfaces to the simulation tools *Gromacs* [3], *ms2* [88], and *korr* (simulated simulations) [89] are implemented. The molecular simulations can be replaced by so-called *simulated simulations* based on equations of state defining functional dependencies between specific force-field parameters and certain physical observables. This makes it possible to compute physical properties without performing time-consuming molecular simulations (see Refs. [60, 89] for further details).

Finally, an abstract class named *BoxConstraints* is used by *OptimizationProblem* with the specific child class *ForceFieldConstraints* implementing the admissible domain $\Omega$ for the force-field parameters. An object of the latter is given to the class *MolecularSimulationOptimizationProblem* derived from the abstract class *OptimizationProblem*. Once the simulation results (i.e., the simulated physical properties) have been calculated, they are given back to the class *PhysicalPropertiesLoss*.

A majority of the modules requires certain input parameters, which have to be defined in a user-written configuration file, and is read by the main python module *main.py*. The configuration file specifies all class objects, modules, and submodules that are desired for optimization process. It also contains important preferences concerning the system (e.g., input/output paths, number of computer cores, batch system), the optimization (e.g., algorithm, step length control, stopping criterion, initial parameters, constraints), and the optimization problem (e.g., objective functions, the loss function's target values). When molecular simulations are performed, all desired properties and parameters of the thermodynamic system have to be defined (e.g., ensemble, temperatures, pressures, physical properties to be fitted, number of molecules, box size, number of MD/MC steps, time step). Hence, the file is divided into three blocks. If more than one substance is considered in the optimization, one block for each substance has to be indicated.

The final output file contains an evaluation in tabular form of all simulation and optimized force-field parameters, the simulated properties along with their actual deviations from the experimental reference data at each temperature, the loss function values, and algorithm-specific information.

The steering of parallel molecular simulations requires special consideration. This is realized by three different modules: the producer, the executer, and the collector. The main function of the *producer*, illustrated in Fig. 6, is to generate all configuration files for the molecular simulations. In order to generate transferable force fields, a variation level was added to the producer. This allows researchers to vary their optimization jobs by the force-field parameters, number of ensembles, temperatures, and molecular models (i.e., different substances). Before running the producer, the user must define all model systems with their properties in the initial configuration file, which contains several sections for each system. The relevant

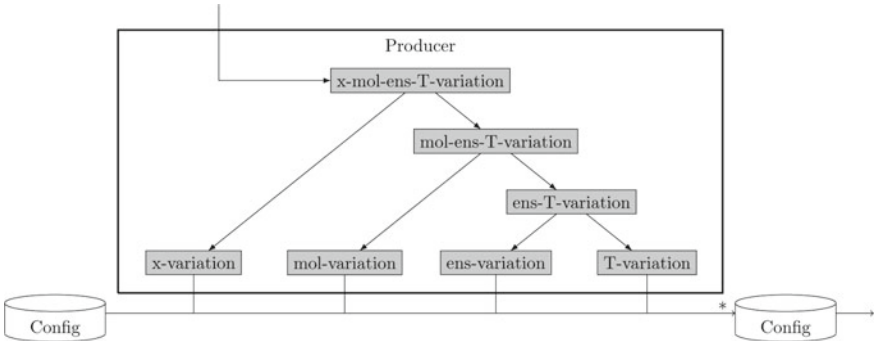


**Fig. 6** Illustration of the producer module comprising the *x-variation*, *mol-variation*, *ens-variation*, and *T-variation* scripts

properties for the producer are the force-field parameters, substances, ensembles, and temperatures.

Generally, all necessary configuration files are realized in the following manner. First, the *x-mol-ens-T-variation* script is started, which calls the *x-variation* script. This script then reads the initial configuration file and generates subdirectories that contain new configuration files with the new force-field parameters as varied by the optimization algorithm. Second, the *mol-ens-T-variation* script calls the *mol-variation* script, which varies the new configuration files with respect to different substances and stores them in new subdirectories. Third, the *ens-T-variation* script calls the *ens-variation* script. This script then reads the new configuration files and varies the ensembles as well. The new files are stored into subdirectories. Finally, the *T-variation* script is called, varying the temperature and storing the new configuration files into a new subdirectory. In summary, the producer generates a four-level subdirectory structure with varied configuration files, as exemplified in Fig. 7, according to the following pattern: force-field parameters–substances–ensembles–temperatures.

After this procedure, the *executer* starts the parallel molecular simulations based on the set of configuration files. After completion, the executer reports the status and results of all simulations to the *collector*. The latter collects the simulation results of each single molecular simulation being stored in the leaf subdirectory level. The main idea is that the collector runs through all result folders, collects the simulated physical properties, and stores them together in a result file within the highest directory level. Afterward, the result file is used for the evaluation of the loss function.

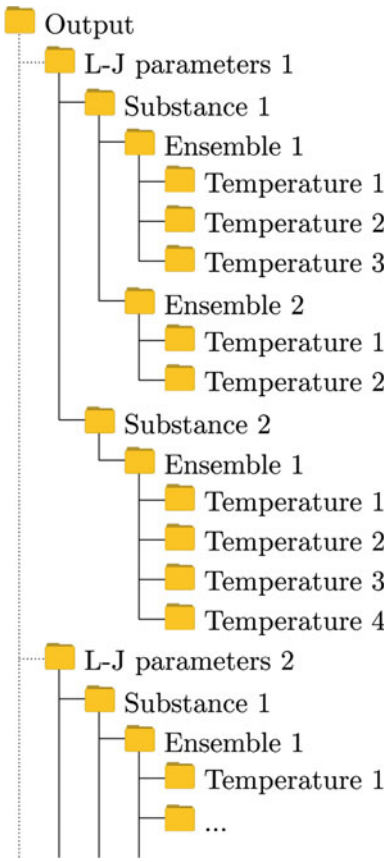


**Fig. 7** Illustration of the four-level subdirectory structure that is generated by the producer module. A unique configuration file is stored in all subdirectories

## 4 Interlinking Aspects of Bonded and Nonbonded Parameter Optimization

It is well known that bonded and nonbonded parameters are coupled to each other. For a given set of nonbonded parameters, there will be an optimal set of bonded parameters and vice versa. This implies that through a successive iteration of bonded and nonbonded parameter optimization, a self-consistent force field should be achieved. Figure 8 shows the interaction between intramolecular and intermolecular parameter optimization tools. Often, an initial set of Lennard–Jones parameters is chosen based on existing force fields and atom types. One then optimizes the bonded parameters using $Wolf_2Pack$. The resulting parameters are then transferred to the intermolecular optimization tools, which optimizes the nonbonded parameters. Depending on the algorithm used, the transferred Lennard–Jones parameters are used as an initial guess (i.e., GROW and SpaGrOW) or they are discarded (i.e., CoSMoS). Once new nonbonded parameters are generated, they

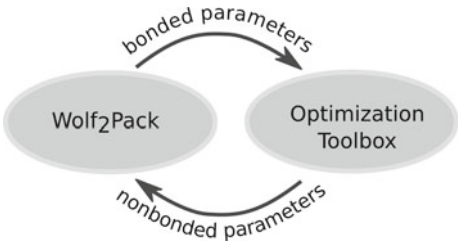


**Fig. 8** Interaction between intramolecular (i.e., $Wolf_2Pack$) and the intermolecular parameter optimization tools (i.e., CoSMoS, GROW, SpaGrOW)

are then transferred back to $Wolf_2Pack$, and the cycle is repeated until all investigated parameters converge. Currently, we are improving our understanding of the sensitivity of this global optimization routine by performing it on selected saturated hydrocarbons (e.g., octane).

## 5 Future Work: Methods and Applications

In addition to researching how to best realize the bonded–nonbonded optimization cycle described in the last section, we are currently working toward the inclusion of solution-phase models (e.g., pure solvent PBC box, ionic liquid PBC boxes) into $Wolf_2Pack$'s database. Experimentally known condense-phase observables (e.g., density, enthalpy of vaporization) will also be included into the database. These models and target experimental values will be accessible to CoSMoS, GROW, and SpaGrOW. This will allow future users to have a common access point and starting models for nonbonded parameter optimization. Once this is realized, the next step will be to extend $Wolf_2Pack$'s online portal to include these condensed-phase models and our nonbonded optimization algorithms, thus unifying our bonded and nonbonded software packages.

With regard to application, we will apply our tools to optimize a force field specific for fluorinated alcohols. Fluorinated alcohols are highly relevant in industrial applications (e.g., as solvents used in chemical separation processes). Their attractiveness is that they can be extracted from the reaction medium and be reused, which makes them both environmentally friendly and economically attractive [90]. The challenge in optimizing such a force field arises from the lack of experimental data and lacks previously published parameters that can be used as an initial input [91–93]. The goal will be to fit both vapor–liquid equilibrium data (e.g., saturated liquid density, vapor pressure) and transport properties (e.g., diffusion coefficients) simultaneously and at different temperatures. Hence, not only parallelization over different substances but also over different ensembles and temperatures are required.

Furthermore, a new force field for carbon dioxide will be developed that reproduces bulk densities, vapor–liquid equilibrium data, and overcritical transport properties (e.g., diffusion coefficients and viscosities) simultaneously. New force

fields for alkaline earth salts, including a transferable parameters, are about to be published.

## 6 Conclusion

In this work, the conception and implementation of recently developed modular program packages applied for force-field parameterizations was described in detail. Intramolecular parameters (i.e., bond length, angles, and torsions) are obtained using the software package $Wolf_2Pack$. Intermolecular parameters, especially Lennard–Jones parameters, are computed via a new set of software tools, implementing a so-called funnel workflow combing global and local optimization procedures. The global metamodeling package CoSMoS is combined with gradient-based (GROW) or derivative-free methods (SpaGrOW). The derivative-free method, based on smoothing procedures and sparse grid interpolation, tends to be much more efficient near the global optimum. The mathematical optimization problem is formulated through the minimization of a loss function between simulated physical properties and experimental reference data. It was shown how the individual software is interlinked with each other within the overall optimization package. These tools form the basis for user-friendly and highly efficient parallelized force-field parameterizations. Finally, several applications are planed in order to obtain industrially relevant force fields (i.e., for solution-phase models, ionic liquids, fluorinated alcohols, alkaline earth salts, and overcritical $CO_2$).

## References

1. Case, D.A., Babin, V., Berryman, J.T., Betz, R.M., Cai, Q., Cerutti, D.S., Cheatham III, T.E., Darden, T.A., Duke, R.E., Gohlke, H., Goetz, A.W., Gusarov, S. Homeyer, N., Janowski, P., Kaus, J., Kolossváry, I., Kovalenko, A., Lee, T.S., LeGrand, S., Lucko, T., Luo, R., Madej, B., Merz, K.M., Paesani, F., Roe, D.R., Roitberg, A., Sagui, C., Salomon-Ferrer, R., Seabra, G., Simmerling, C.L., Smith, W., Swails, J., Walker, R.C., Wang, J., Wolf, R.M., Wu, X., Kollmann, P.A.: AMBER 14. http://ambermd.org. University of California, San Francisco (2014)
2. Brooks, B.R., Brooks III, C.L., Mackerell, A.D., Nilsson, L., Petrella, R.J., Roux, B., Won, Y., Archontis, G., Bartels, C., Caflisch, S.B.A., Caves, L., Cui, Q., Dinner, A.R., Feig, M., Fischer, S., Gao, J., Hodoscek, M., Im, W., Kuczera, K., Lazaridis, T., Ma, J., Ovchinnikov, V., Paci, E., Pastor, R.W., Post, C.B., Pu, J.Z., Schaefer, M., Tidor, B., Venable, R.M., Woodcock, H.L., Wu, X., Yang, W., York, D.M., Karplus, M.: Charmm: the biomolecular simulation program. J. Comp. Chem. **30**, 1545–1615 (2009)
3. Hess, B., van der Spoel, D., Lindahl, E.: Gromacs user manual 4.5.4. http://www.gromacs.org/Documentation/Manual/manual-4.5.4.pdf (2010)
4. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. J. Comp. Phys. **117**, 1–19 (1995)

5. Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., Myers, J.: Examining the challenges of scientific workflows. Computer **40**, 24–32 (2007)
6. Waldher, B., Kuta, J., Chen, S., Henson, N., Clark, A.E.: ForceFit: a code to fit classical force fields to quantum mechanical potential energy surfaces. J. Comp. Chem. **12**, 2307–2316 (2010)
7. Highly optimized object-oriented many-particle dynamics—blue edition. http://codeblue.umich.edu/hoomd-blue/ (2011)
8. Halverson, J.D., Brandes, T., Lenz, O., Arnold, A., Bevc, S., Starchenko, V., Kremer, K., Stuehn, T., Reith, D.: ESPResSo++: a modern multiscale simulation package for soft matter systems. Comput. Phys. Commun. **184**, 1129–1149 (2013)
9. Karimi-Varzaneh, H., Qian, H., Chen, X., Carbone, P., Müller-Plathe, F.: Ibisco: a molecular dynamics simulation package for coarse-grained simulation. J. Comp. Chem. **32**, 1475–1487 (2011)
10. Singer, S.J., Nicolson, G.L.: The fluid mosaic model of the structure of cell membranes. Science **175**, 720–731 (1972)
11. Zhou, Y., Stell, G.: Chemical association in simple models of molecular and ionic fluids II. Thermodynamic properties. J. Chem. Phys. **96**, 1504–1506 (1992)
12. Siepmann, J.I., Karaborni, S., Smit, B.: Simulating the critical behaviour of complex fluids. Nature **365**, 330–332 (1993)
13. O'Connell, S.T., Thompson, P.A.: Molecular dynamics-continuum hybrid computations: a tool for studying complex fluid flow. Phys. Rev. E **52**, 5792–5795 (1995)
14. Kolafa, J., Nezbeda, I., Lisal, M.: Effect of short- and long-range forces on the properties of fluids. III. dipolar and quadrupolar fluids. Mol. Phys. **99**, 1751–1764 (2001)
15. Valiullin, R., Naumov, S., Galvosas, P., Kärger, J., Woo, H.-J., Porcheron, F., Monson, P.A.: Exploration of molecular dynamics during transient sorption of fluids in mesoporous materials. Nature **443**, 965–968 (2006)
16. Batra, I.P., Bennett, B.I., Herman, F.: Simple molecular model for crystalline tetrathiofulvalene-tetracyanoquinodimethane (TTF-TCNQ). Phys. Rev. B **11**, 4927–4934 (1975)
17. Fehlner, T.P.: Molecular models of solid state metal boride structure. J. Solid State Chem. **154**, 110–113 (2000)
18. Della, C.N., Dongwei, S.: Mechanical properties of carbon nanotubes reinforced ultra high molecular weight polyethylene. Solid State Phenom. **136**, 45–49 (2008)
19. Lin, S.-T., Blanco, M., Goddard III, W.A.: The two-phase model for calculating thermodynamic properties of liquids from molecular dynamics: validation for the phase diagram of Lennard-Jones fluids. J. Chem. Phys. **119**, 11792–11805 (2003)
20. Bien, D.E., Chiriac, V.A.: A novel molecular approach to modeling phase change in micro-fluidic systems. In: Proceedings of the 9th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, pp. 598–604. IEEE, New Jersey (2004)
21. Vrabec, J., Gross, J.: Vapor–liquid equilibria simulation and an equation of state contribution for dipole-quadrupole interactions. J. Phys. Chem. B **112**, 51–60 (2008)
22. Levitt, M., Warshel, A.: Computer simulation of protein folding. Nature **253**, 694–698 (1975)
23. Gsponer, J., Caflisch, A.: Molecular dynamics simulations of protein folding from the transition state. In: Fersth, A. (ed.) Proceedings of the National Academy of Sciences (PNAS), vol. 99, pp. 6719–6724. Washington (2002)
24. Snow, C.D., Sorin, E.J., Rhee, Y.M., Pandel, V.S.: How well can simulation predict protein folding kinetics and thermodynamics? Annu. Rev. Biophys. Biomol. Struct. **34**, 43–69 (2005)
25. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. **117**, 500–544 (1952)
26. Barkla, B.J., Pantoja, O.: Physiology of ion transport across the tonoplast of higher plants. Annu. Rev. Plant Physiol. Plant Mol. Biol. **47**, 159–184 (1996)
27. Müller-Plathe, F., Reith, D.: Cause and effect reversed in non-equilibrium molecular dynamics: an easy route to transport coefficients. Comput. Theor. Polymer Sci. **9**, 203–209 (1999)

28. Bordat, P., Reith, D., Müller-Plathe, F.: The influence of interaction details on the thermal diffusion in binary Lennard-Jones liquids. J. Chem. Phys. **115**, 8978–8982 (2001)
29. Guevara-Carrion, G., Nieto-Draghi, C., Vrabec, J., Hasse, H.: Prediction of transport properties by molecular simulation: methanol and ethanol and their mixture. J. Phys. Chem. B **112**, 16664–16674 (2008)
30. Grest, G.S., Kremer, K.: Molecular dynamics simulation for polymers in the presence of a heat bath. Phys. Rev. A **33**, 3628–3631 (1986)
31. Müller-Plathe, F.: Permeation of polymers—a computational approach. Acta Polymer. **45**, 259–293 (1994)
32. Binder, K.: Monte Carlo and molecular dynamics simulations in polymer science. Oxford University Press, Oxford (1995)
33. Kremer, K., Müller-Plathe, F.: Multiscale simulation in polymer science. Mol. Sim. **28**, 729–750 (2002)
34. Praprotnik, M., Junghans, C., Delle Site, L., Kremer, K.: Simulation approaches to soft matter: generic statistical properties vs. chemical details. Comput. Phys. Commun. **179**, 51–60 (2008)
35. Reith, D., Kirschner, K.N.: A modern workflow for force field development—bridging quantum mechanics and atomistic computational models. Comput. Phys. Commun. **182**, 2184–2191 (2011)
36. Krämer-Fuhrmann, O., Neisius, J., Gehlen, N., Reith, D., Kirschner, K.N.: $Wolf_2Pack$ – Portal based atomistic force field development. J. Chem. Inf. Mod. **53**, 802–808 (2013)
37. Vainio, M.J., Johnson, M.S.: Generating conformer ensembles using a multiobjective genetic algorithm. J. Chem. Inf. Mod. **47**, 2462–2474 (2007)
38. Tasi, G., Mizukami, F., Csontos, J., Gyõrffy, W., Pálinkó, I.: Quantum algebraic–combinatoric study of the conformational properties of *n*-alkanes. II. J. Math. Chem. **27**, 191–199 (2000)
39. Jorgensen, W.L., Madura, J.D., Swensen, C.J.: Optimized intermolecular potential functions for liquid hydrocarbons. J. Am. Chem. Soc. **106**, 6638–6646 (1984)
40. Martin, M.G., Siepmann, J.I.: Transferable potentials for phase equilibria. 1. United-atom description of *n*-alkanes. J. Phys. Chem. B **102**, 2569–2577 (1998)
41. Ungerer, P., Beauvais, C., Delhommelle, J., Boutin, A., Rousseau, B., Fuchs, A.H.: Optimization of the anisotropic united atoms intermolecular potential for *n*-alkanes. J. Phys. Chem. **112**, 5499–5510 (2000)
42. Bourasseau, E., Haboudou, M., Boutin, A., Fuchs, A.H., Ungerer, P.: New optimization method for intermolecular potentials: optimization of a new anisotropic united atoms potential for olefins: prediction of equilibrium properties. J. Chem. Phys. **118**, 3020–3035 (2003)
43. Stoll, J., Vrabec, J., Hasse, H.: A set of molecular models for carbon monoxide and halogenated hydrocarbons. J. Chem. Phys. **119**, 11396–11407 (2003)
44. Reith, D., Pütz, M., Müller-Plathe, F.: Deriving effective mesoscale potentials from atomistic simulations. J. Comp. Chem. **24**, 1624–1636 (2003)
45. Oostenbrink, C., Villa, A., Mark, A.E., van Gunsteren, W.F.: A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. J. Comp. Chem. **25**, 1656–1676 (2004)
46. Sun, H.: Prediction of fluid densities using automatically derived VDW parameters. Fluid Phase Eq. **217**, 59–76 (2004)
47. Eckl, B., Vrabec, J., Hasse, H.: On the application of force fields for predicting a wide variety of properties: ethylene oxide as an example. Fluid Phase Eq. **274**, 16–26 (2008)
48. Cacelli, I., Cimoli, A., Livotto, P.R., Prampolini, G.: An automated approach for the parameterization of accurate intermolecular force-fields: pyridine as a case study. J. Comp. Chem. **33**, 1055–1067 (2012)
49. Ucyigitler, S., Camurdan, M.C., Elliott, J.R.: Optimization of transferable site–site potentials using a combination of stochastic and gradient search algorithms. Ind. Eng. Chem. Res. **51**, 6219–6231 (2012)
50. Eckelsbach, S., Janzen, T., Köster, A., Mirshnichenko, S., Muñoz Muñoz, Y.M., Vrabec, J.: Molecular models for cyclic alcanes and ethyl acetate as well as surface tension data from molecular simulation. In: Nagel, W.E., Kröner, D.E., Resch, M.M. (eds.) High Performance

Computing in Science and Engineering '14, Transactions of the High Performance Computing Center, HLRS, Stuttgart (2014), pp. 645–659. Springer, Berlin (2015)
51. Muñoz Muñoz, Y.M., Guevara-Carrion, G., Llano-Restrepo, M., Vrabec, J.: Lennard–Jones force field parameters for cyclic alkanes from cyclopropane to cyclohexane. Fluid Phase Eq. **404**, 150–160 (2015)
52. Kirschner, K.N., Yongye, A.B., Tschampel, S.M., Gonzalez-Outeirino, J., Daniels, C.R., Foley, B.L., Woods, R.J.: GLYCAM06: a generalizable biomolecular force field. Carbohydrates. J. Comp. Chem. **29**, 622–655 (2008)
53. Faller, R., Schmitz, H., Biermann, O., Müller-Plathe, F.: Automatic parameterization of force fields for liquids by simplex optimization. J. Comp. Chem. **20**, 1009–1017 (1999)
54. Dupradeau, F.-Y., Pigache, A., Zaffran, T., Savineau, C., Lelong, R., Grivel, N., Lelong, D., Rosanski, W., Cieplak, P.: The R.E.D. tools: advances in RESP and ESP charge derivation and force field library building. Phys. Chem. Chem. Phys. **12**, 7821–7839 (2010)
55. Hülsmann, M.: Effiziente und neuartige Verfahren zur Optimierung von Kraftfeldparametern bei atomistischen Molekularen Simulationen kondensierter Materie. In: Fraunhofer SCAI (ed.) Fraunhofer-Verlag, Ph.D. thesis, University of Cologne, Germany (2012)
56. Krämer, A., Hülsmann, M., Köddermann, T., Reith, D.: Automated parameterization of intermolecular pair potentials using global optimization techniques. Comput. Phys. Commun. **185**, 3228–3239 (2014)
57. Regis, R., Shoemaker, C.: Constrained global optimization of expensive black box functions using radial basis functions. J. Glob. Opt. **31**, 153–171 (2005)
58. Hülsmann, M., Köddermann, T., Vrabec, J., Reith, D.: GROW: A gradient-based optimization workflow for the automated development of molecular models. Comput. Phys. Commun. **181**, 499–513 (2010)
59. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
60. Hülsmann, M., Vrabec, J., Maaß, A., Reith, D.: Assessment of numerical optimization algorithms for the development of molecular models. Comput. Phys. Commun. **181**, 887–905 (2010)
61. Hülsmann, M., Müller, T.J., Köddermann, T., Reith, D.: Automated force field optimization of small molecules using a gradient-based workflow package. Mol. Sim. **36**, 1182–1196 (2011)
62. Köddermann, T., Kirschner, K.N., Vrabec, J., Hülsmann, M., Reith, D.: Liquid-liquid equilibria of dipropylene glycol dimethyl ether and water by molecular dynamics. Fluid Phase Eq. **310**, 25–31 (2011)
63. Hülsmann, M., Kopp, S., Huber, M., Reith, D.: Efficient gradient and Hessian calculations for numerical optimization algorithms applied to molecular simulations. In: Proceedings of the International Conference on Mathematical Modeling in Physical Sciences (IC-MSQUARE), Budapest, Hungary (2012), IOP Publishing, Journal of Physics: Conference Series 410, 012007 (2013)
64. Hülsmann, M., Kopp, S., Huber, M., Reith, D.: Utilization of efficient gradient and Hessian computations in the force field optimization process of molecular simulations. Comput. Sci. Disc. **6**, 015005 (2013)
65. Hülsmann, M., Reith, D.: SpaGrOW—a derivative-free optimization scheme for intermolecular force field parameters based on sparse grids methods. Entropy **15**, 3640–3687 (2013)
66. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. Roy. Stat. Soc. Ser. B **67**, 301–320 (2005)
67. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. Sov. Math. Doklady **4**, 240–243 (1963)
68. Griebel, M., Schneider, M., Zenger, C.: A combination technique for the solution of sparse grid problems. Technical Report, Institute for Computer Science, Technical University of Munich, Germany (1990)
69. Ditchfield, R., Hehre, W.J., Pople, J.A.: Self consistent molecular orbital methods. IX. An extended Gaussian type basis for molecular orbital studies of organic molecules. J. Chem. Phys. **54**, 724–728 (1971)

70. Dunning, T.H.: Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. J. Chem. Phys. **90**, 1007–1023 (1989)
71. Maier, J.A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K.E., Simmerling, C.: ff14SB: improving the accuracy of protein side chain and backbone parameters from ff99SB. J. Chem. Theory Comput. **11**, 3696–3713 (2015)
72. Wang, J., Wolf, R.M., Caldwell, J.W., Kollman, P.A., Case, D.A.: Development and testing of a general amber force field. J. Comput. Chem. **25**, 1157–1174 (2004)
73. Dickson, C.J., Madej, B.D., Skjevik, Å.A., Betz, R.M., Teigen, K., Gould, I.A., Walker, R.C.: Lipid14: the amber lipid force field. J. Chem. Theory Comput. **10**, 865–879 (2014)
74. Wang, J.M., Kollman, P.A.: Automatic parameterization of force field by systematic search and genetic algorithms. J. Comp. Chem. **22**, 1219–1228 (2001)
75. Vaiana, A.C., Cournia, Z., Costescu, I.B., Smith, J.C.: AFMM: a molecular mechanics force field vibrational parametrization program. Comput. Phys. Commun. **167**, 34–42 (2005)
76. Guvench, O., MacKerell Jr, A.D.: Automated conformational energy fitting for force field development. J. Mol. Model. **14**, 667–679 (2008)
77. Mayne, C.G., Saam, J., Schulten, K., Tajkhorshid, E., Gumbart, J.C.: Rapid parameterization of small molecules using the force field toolkit. J. Comp. Chem. **32**, 2757–2770 (2013)
78. Hopkins, C.W., Roitberg, A.E.: Fitting of dihedral terms in classical force fields as an analytic linear least-squares problem. J. Chem. Inf. Mod. **54**, 1978–1986 (2014)
79. Burger, S.K., Ayers, P.W., Schofield, J.: Efficient parameterization of torsional terms for force fields. J. Comp. Chem. **35**, 1438–1445 (2014)
80. Betz, R.M., Walker, R.C.: Paramfit: automated optimization of force field parameters for molecular dynamics simulations. J. Comp. Chem. **36**, 79–87 (2015)
81. Vanommeslaeghe, K., Mingjun, Y., MacKerell, A.D.: Robustness in the fitting of molecular mechanics parameters. J. Comp. Chem. **36**, 1083–1101 (2015)
82. Vanduyfhuys, L., Vandenbrande, S., Verstraelen, T., Schmid, R., Waroquier, M., Van Speybroeck, V.: QuickFF: a program for a quick and easy derivation of force fields for metal-organic frameworks from ab initio input. J. Comp. Chem. **36**, 1015–1027 (2015)
83. Gordon, M.D., Schmidt, M.W.: Advances in electronic structure theory: GAMESS a decade later. In: Gordon, M.S., Schmidt, W., Dykstra, C.E. (eds.) Theory and Applications of Computational Chemistry: The First Forty Years, pp. 1167–1189. Elsevier Amsterdam Boston (2005)
84. O'Boyle, N., Banck, M., James, C., Morley, C., Vandermeersch, T., Hutchison, G.: Open babel: an open chemical toolbox. J. Cheminf. **3**, 33 (2011)
85. R: A language and environment for statistical computing. manual. http://www.R-project.org. The R Foundation for Statistical Computing, Vienna, Austria (2009)
86. PyMOL(TM) Molecular Graphics System, Version 1.6.0.0. http://pymol.org && http://sourceforge.net/projects/pymol/ (2009)
87. The LaTeX Project. http://latex-project.org/
88. Deublein, S., Eckl, B., Stoll, J., Lishchuk, S.V., Guevara-Carrion, G., Glass, C.W., Merker, T., Bernreuther, M., Hasse, H., Vrabec, J.: ms2: a molecular simulation tool for thermodynamic properties. Comput. Phys. Commun. **182**, 2350–2367 (2011)
89. Stoll, J., Vrabec, J., Hasse, H., Fischer, J.: Comprehensive study of the vapour–liquid equilibria of the pure two–centre Lennard-Jones plus point quadrupole fluid. Fluid Phase Eq. **179**, 339–362 (2001)
90. Bégué, J.-P., Bonnet-Delpon, D., Crousse, B.: Fluorinated alcohols: anew medium for selective and clean reaction. Synlett, 18–29 (2004)
91. Rochester, C.H., Symonds, J.R.: Densities of solutions of four fluoralcohols in water. J. Fluorine Chem. **4**, 141–148 (1974)
92. Gross, T., Karger, N., Price, W.E.: p, T dependence of self-diffusion in 2-fluoroethanol, 2,2 difluoroethanol and 2,2,2-trifluoroethanol. J. Mol. L. **75**, 159–168 (1998)
93. Meeks, A.C., Goldfarb, I.J.: Vapor pressure of fluoroalcohols. J. Chem. Eng. Data **12**, 196 (1967)