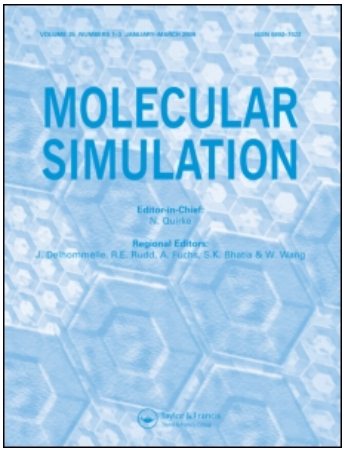

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Automated force field optimisation of small molecules using a gradient-based workflow package

Marco Hülsmann[a]; Thomas J. Müller[b]; Thorsten Ködderman[a]; Dirk Reith[a]
[a] Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Schloß Birlinghoven, St Augustin, Germany [b] Eduard-Zintl-Institut für Anorganische und Physikalische Chemie, Technische Universität Darmstadt, Darmstadt, Germany

Online publication date: 10 December 2010

To cite this Article Hülsmann, Marco , Müller, Thomas J. , Ködderman, Thorsten and Reith, Dirk(2010) 'Automated force field optimisation of small molecules using a gradient-based workflow package', Molecular Simulation, 36: 14, 1182 — 1196
To link to this Article: DOI: 10.1080/08927022.2010.513974
URL: http://dx.doi.org/10.1080/08927022.2010.513974

Molecular Simulation
Vol. 36, No. 14, December 2010, 1182–1196

Taylor & Francis
Taylor & Francis Group

Automated force field optimisation of small molecules using a gradient-based workflow package

Marco Hülsmann[a1], Thomas J. Müller[b2], Thorsten Ködderman[a3] and Dirk Reith[a]*

[a]*Fraunhofer Institute for Algorithms and Scientific Computing (SCAI), Schloß Birlinghoven, 53754 St Augustin, Germany;* [b]*Eduard-Zintl-Institut für Anorganische und Physikalische Chemie, Technische Universität Darmstadt, Petersenstr. 20, 64285 Darmstadt, Germany*

(*Received 21 December 2009; final version received 4 August 2010*)

In this study, the recently developed gradient-based optimisation workflow for the automated development of molecular models is for the first time applied to the parameterisation of force fields for molecular dynamics simulations. As a proof-of-concept, two small molecules (benzene and phosgene) are considered. In order to optimise the underlying intermolecular force field (described by the (12,6)-Lennard-Jones and the Coulomb potential), the energetic and diameter parameters ϵ and σ are fitted to experimental physical properties by gradient-based numerical optimisation techniques. Thereby, a quadratic loss function between experimental and simulated target properties is minimised with respect to the force field parameters. In this proof-of-concept, the considered physical target properties are chosen to be diverse: density, enthalpy of vapourisation and self-diffusion coefficient are optimised simultaneously at different temperatures. We found that in both cases, the optimisation could be successfully concluded by fulfillment of a pre-defined stopping criterion. Since a fairly small number of iterations were needed to do so, this study will serve as a good starting point for more complex systems and further improvements of the parametrisation task.

Keywords: force field development; numerical optimisation; gradient-based algorithms; molecular dynamics; Lennard-Jones potential

1. Introduction

In the last decades, molecular simulation and force field development have become indispensable in the field of computational chemistry, as they are applicable to a large variety of areas, such as thermodynamic properties of fluids [1–7], mechanic properties of solids [8–10], phase change phenomena [11–13], transport processes in biologic tissue [14,15], protein folding [16–18], transport processes in liquids [19–21], polymer properties by using different length scales [22–25] or generic statistic properties of soft matter [26].

A force field describes the intra- and intermolecular interactions between the particles and has to be defined such that the molecular model predicts a large number of physical properties simultaneously and correctly. The parameterisation of a force field for specific systems is a challenge an its own, and in recent years, there has been much effort to define and optimise force fields [27–35]. However, many practical approaches are based on quantum mechanics and only optimise intramolecular force field parameters and partial charges. As the computational time increases with the size of the molecular system, quantum mechanical methods fail in the case of the optimisation of intermolecular force fields, as in order to achieve this goal, a sufficiently large system size is required. Therefore, the application of empirical parameterisation methods has become more and more important in the field of molecular simulation, wherein some a priori selected physical properties are fitted to experimental target data. Even though some scientific groups have already dealt with this problem [36–46], a unique way to solve it for arbitrary systems has not been found yet. In fact, a trial-and-error investigation is still often practised by computational chemists in spite of the partial success of systematic optimisation strategies, leading to sub-optimal force field parameters. In order to overcome that situation, we find the usage of fast numerical optimisation algorithms and the development of automated optimisation procedures to be of prime importance. The most significant contributions in this regard shall be pointed out briefly.

Faller et al. [38] used the simplex algorithm by Nelder and Mead [47] as iterative optimisation procedure in order to detect a local minimum of a quadratic loss function between simulated and experimental target data. The simplex algorithm is quite robust with respect to noisy data but its convergency is very slow. Furthermore, it is not directed to the minimum and hence, in the first steps, it strays in the parameter space. In the last steps, it starts to hop around the local minimum. The simplex method

*Corresponding author. Email: dirk.reith@scai.fraunhofer.de

ISSN 0892-7022 print/ISSN 1029-0435 online

DOI: 10.1080/08927022.2010.513974
http://www.informaworld.com

delivered some very good results, predicting the density and enthalpy of vapourisation of small molecules [34,38] and acceptable results predicting radial distribution functions of polymers [48], but it was always used at one temperature only.

Some years later, Ungerer et al. [39] applied a gradient-based method derived from a linear Taylor expansion resulting in the solution of a linear equation system (LES). First, the partial derivatives were approximated by finite differences, and later, Bourasseau et al. [40] developed a new method to evaluate the partial derivatives. Their technique turned out to be more efficient than the simplex algorithm, but it also has some drawbacks: First, it is not clear *a priori* whether the solution of the LES consists of positive, i.e. meaningful, force field parameters, and second, the linearisation of the Taylor expansion can in practice only be used close to the minimum. Hence, the initial force field parameters must already be chosen quite accurately. Third, the method cannot be applied to underdetermined problems because the LES is singular. In the case of some specific small molecules, the method delivered very good results predicting the density, enthalpy of vapourisation and vapour pressure at different temperatures simultaneously, but especially for larger molecules, the results were not fully satisfying [40].

We wanted to revisit the parametrisation problem in order to achieve a sustained improvement of the modelling process. In order to do so, the overall problem is divided and studied along the main problem sub-classes: the development of a suitable program toolkit, the determination of the most efficient algorithm and the problem of controlled noise handling. In this work, the performance of our recently developed automated gradient-based optimisation workflow (GROW) is studied [49] for the first time in real application scenarios. The implementation of GROW contains a large number of efficient iterative numerical optimisation algorithms such as steepest descent and conjugate gradients (CGs) as well as algorithms which also use Hessian matrices such as Newton Raphson and trust region (TR). The main problem with this kind of algorithms is the fact that it is not clear how they can handle statistical noise, which is present in all molecular simulations. Therefore, some mathematical adjustment has to be made in order to deal with this problem. In [50], a detailed assessment of algorithm candidates was executed also with respect to noise. Thereby, molecular simulations were replaced by fit functions for vapour–liquid equilibrium (VLE) data [51], and artificial noise was added to the calculated physical properties. Doing so offered the possibility to exactly control the noise level and, hence, execute a deterministic test of the algorithms. It was practically proven that iterative gradient-based algorithms – and, therefore, GROW – can be applied to the optimisation of force fields. In this work, GROW is applied for the first time to real molecular simulations. The simulation tools integrated in the workflow are the molecular dynamics (MD) software packages *GROMACS*[4] (version 4.0) and *Moscito*[5] (version 4). The considered physical properties are density, enthalpy of vapourisation and self-diffusion coefficient.

The two substances benzene and phosgene were chosen for the following reasons: benzene is a simple molecule due to its quite simple structure and the fact that it is non-polar. Furthermore, benzene is symmetric with only two chemically independent atom types. Hence, the parameterisation of its force field was deemed to be a relatively easy task. However, it is still challenging because of the π-interactions. Phosgene is polar but at the same time quite polarisable. This means that it can be expected that the interaction surface of phosgene is temperature dependent. Therefore, we chose this molecule in order to test how GROW can deal with such kinds of systems.

2. Previous work

2.1 Optimisation task

In order to apply iterative numerical optimisation procedures, we need to define the optimisation task in a mathematical way. Hence, GROW [49] is aimed at minimising the following quadratic loss function between the calculated (e.g. from simulation) and target (e.g. from experiment) data:

$$F(x) = \sum_{i=1}^{n} w_i \left(\frac{f_i^{\exp} - f_i^{\mathrm{sim}}(x)}{f_i^{\exp}} \right)^2, \tag{1}$$

where $x = (x_1, \ldots, x_N)^T \in \mathbb{R}^N$ is the force field parameter vector, N is the number of force field parameters, n is the number of physical properties considered in the optimisation, $f_i^{\mathrm{sim}}(x)$ is the ith property dependent on the force field parameters and $f_i^{\exp}$ is the respective target value. The weights w_i account for the fact that some properties may be easier to reproduce than others. This loss function has to be minimised with respect to x.

Optimising with respect to different properties at different temperatures T from a temperature range $\mathcal{T}$ simultaneously, we can express the loss function as follows:

$$F(x) = \sum_{i=1}^{n} \sum_{T \in \mathcal{T}} w_{i,T} \left(\frac{f_{i,T}^{\exp} - f_{i,T}^{\mathrm{sim}}(x)}{f_{i,T}^{\exp}} \right)^2. \tag{2}$$

The gradient ∇F and the Hessian matrix $D^2 F$ required for the different gradient-based optimisation algorithms are calculated at each iteration $x^k, k = 0, 1, 2, \ldots,$ via

finite differences for the partial derivatives,

$$\frac{\partial f_{i,T}^{\mathrm{sim}}}{\partial x_j}(x) = \frac{f_{i,T}^{\mathrm{sim}}(x_1, \ldots, x_j + h, \ldots, x_N) - f_{i,T}^{\mathrm{sim}}(x)}{h} + \mathcal{O}(h), \quad i = 1, \ldots, n, \quad j = 1, \ldots, N, \tag{3}$$

starting from an initial force field parameter vector x^0. The latter must be situated in the sphere of influence of a – at least – local minimum x^{opt}, where $\nabla F(x^{\mathrm{opt}}) = 0$ and $D^2F(x^{\mathrm{opt}})$ is positive definite. The force field parameters, i.e. the components of the parameter vector x, are chosen from a compact admissible domain, as they obey certain boundary conditions. Hence, it must be guaranteed that the local or global minimum of F is situated inside this domain. Otherwise, it has to be increased afterwards.

2.2 Software tool GROW

The software tool GROW, whose implementation issues are published in [49], enables the user to perform an automated gradient-based iterative optimisation workflow as given in Section 1. It is modularly constructed and consists of a main control script, specific implementations and secondary control scripts for each numerical algorithm. Taken together, this machinery is able to automatically optimise force fields and is extensible by developers with regard to further optimisation algorithms and simulation tools.

The implementation of GROW contains the numerical optimisation algorithms given in Section 3 and acts as an interface between optimisation and simulation providing all necessary control routines for both tasks. On the optimisation side, it starts with an initial parameter vector, computes its loss function value, gradient and – if required – Hessian via simulation and then evaluates a pre-defined stopping criterion. If this criterion is fulfilled, the actual force field parameters are taken as final and the optimisation workflow terminates. Otherwise, the parameters must be updated by the gradient-based optimisation algorithm. In order to determine an adequate length of the search direction and not to leave the admissible domain, we had to apply a step length control method. For descent methods, i.e. methods with the updating formula

$$x^{k+1} = x^k + t_k d^k, \tag{4}$$

where d^k is the actual descent direction and t_k the step length; the Armijo step length control turned out to be suitable for the present problem. The Armijo step length t_{A} is defined as

$$t_A = \max\{\beta_{\mathrm{A}}^{\ell} | \ell = 0, 1, \ldots, F(x + \beta_{\mathrm{A}}^{\ell} d) \leq F(x) + \zeta_{\mathrm{A}} \beta_{\mathrm{A}}^{\ell} \langle \nabla F(x), d \rangle\}, \tag{5}$$

with $0 < \beta_{\mathrm{A}}, \zeta_{\mathrm{A}} < 1$. The Armijo step length only exists, if d is a descent direction, but then, the convergency of the whole optimisation algorithm is guaranteed to be monotonous.

As $\beta_{\mathrm{A}}^{\ell} \overset{\ell \to \infty}{\to} 0$ and β_{A}^{1} will already be very small in relation to the range of the force field parameters due to the boundary conditions, the resulting step length will be too small so that no profit is drawn. Hence, the usual stopping criterion $\nabla F(x^{\mathrm{opt}}) = 0$ will not be fulfilled within a reasonable amount of computation time. Therefore, the authors decided to use the stopping criterion

$$F(x) \leq \tau, \tag{6}$$

with $\tau > 0$ as small as possible (e.g. 10^{-3} or even smaller).

On the simulation side, there is a control script calling the simulation tool, i.e. all preparation routines, the main executable which manages the simulation itself and collocates the trajectory, and the computation routines writing out all considered physical properties into files. These are read by the optimisation side of GROW and the workflow continues. If properties at different temperatures are considered, the required simulations have to be executed in parallel. In this case, the optimisation side calls a script distributing K jobs at K temperatures. A script controlling the parallel environment and the simulation control script are called K times. If $m = n/K$ physical properties are considered, the result of each job consists of m properties files. The K results are finally passed on to the optimisation side in order to evaluate the loss function and the stopping criterion. Figure 1 shows the optimisation and the simulation side of GROW as well as their connections in the case of parallel jobs at different temperatures.

2.3 Assessment of gradient-based algorithms

As the most time-consuming step depicted in Figure 1 is the molecular simulation, efficient numerical algorithms with low complexity were applied in order to accelerate the convergency of the optimisation procedure. The following numerical methods were chosen:

(1) **Descent methods:** $x^{k+1} = x^k + t_k d^k$
- Steepest descent: $d^k = -\nabla F(x^k)$
- Newton Raphson: $d^k = -(D^2F(x^k))^{-1}\nabla F(x^k)$
- Quasi Newton: $d^k = -H_k^{-1}\nabla F(x^k)$. The approximation H_k of the Hessian is updated at every iteration, which can be achieved by different Quasi Newton methods.
  - Powell symmetric Broyden

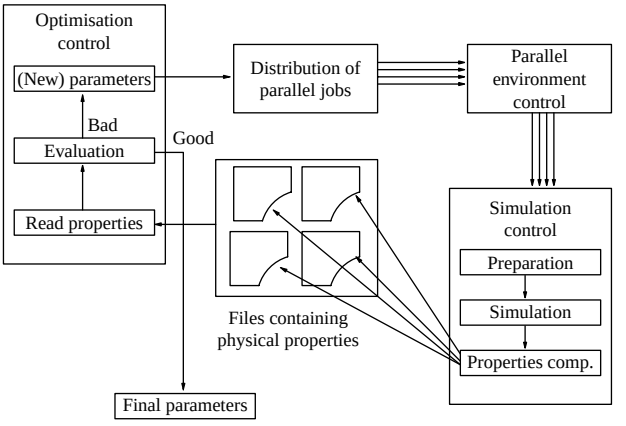

Figure 1. GROW as an interface between optimisation and simulation in the case of parallel jobs at different temperatures: on the left side, the optimisation control is performed. If the stopping criterion is not fulfilled, the actual parameters are passed on to a distribution control script submitting parallel jobs at different temperatures. After the execution of a parallel environment control script, a simulation control script is called performing all necessary preparations, the simulation itself, and the computation of the properties to be optimised. The latter are written into separate files which are read by the optimisation control script. After the evaluation of the loss function, the workflow continues.

- Davidon Fletcher Powell
- Broyden Fletcher Goldfarb Shanno

(2) **CG Methods:** $d^{k+1} = -\nabla F(x^{k+1}) + \beta_k d^k, \quad d^0 = -\nabla F(x^0)$.
- Fletcher Reeves: $\beta_k^{\mathrm{FR}} = \|\nabla F(x^{k+1})\|^2 / \|\nabla F(x^k)\|^2$
- Polak Ribière: $\beta_k^{\mathrm{PR}} = \langle \nabla F(x^{k+1}) - \nabla F(x^k), \nabla F(x^{k+1}) \rangle / \|\nabla F(x^k)\|^2$

(3) **TR methods:** $x^{k+1} = x^k + d^k$ with the so-called TR *subproblem*: $d^k = d^k(\Delta)$ for some pre-defined step length Δ. There are two ways to solve the subproblem:
- Double dog leg algorithm (DD): geometric approach
- Exact solution: eigenvalue decomposition of Hessian

The algorithms and the Armijo step length control to determine t_k are described and discussed in detail in [49,52].

In [50], the authors entirely focused on the best choice of the optimisation algorithms, which were assessed using a simple molecular model, which is reasonable for numerous real fluids. It is based on the quadrupolar two-centre Lennard-Jones (LJ) potential and hence, it is only valid for LJ fluids consisting of two centres with a quadrupolar moment. As an example, nitrogen was taken. The considered physical properties could be computed in a very fast way because molecular simulations were replaced by the fit functions for VLE data developed by [51]. Please note that in [50], no simulations had to be performed and the best and most efficient algorithm could be identified by carrying out a detailed algorithm assessment. In order to mimic molecular simulation runs as realistically as possible, artificial noise was added to the calculated physical properties, i.e. uniformly distributed random numbers within a certain interval around the actual calculated property. The properties used for the assessment were saturated liquid density, enthalpy of vapourisation and vapour pressure on the vapour–liquid coexistence curve. In total, there were eight different optimisation tasks considering different combinations of properties at only one and six different temperatures with and without artificial noise. Furthermore, the shape of the loss function (2) was studied and it was found that it has the form of a steep rain drain leading to difficulties for the gradient-based algorithms. This was another argument for the choice of the stopping criterion 6. Parameter τ was different for each optimisation task and it was determined using the steepest descent method as a reference method: whenever the Armijo step length control did not converge within 100 iterations, a small upper bound for the current loss function value was chosen for τ.

The results of the detailed assessment were the following: the quasi Newton methods and the solution of the TR subproblem by a DD algorithm turned out not to be suitable for the present optimisation tasks, as the matrices involved in those methods were not symmetric positive definite, and at some iteration steps even singular. The steepest descent, the Newton Raphson, the Fletcher Reeves, the Polak Ribière and the TR method with an exact solution of the subproblem mostly fulfilled the respective stopping criterion within a reasonable number of iterations, regardless of the presence of noise. However, the Newton Raphson method often used the steepest descent direction, as the Hessian was not symmetric positive definite in most cases.

The CG methods and the TR method with exact solution of the subproblem were found to be the best numerical optimisation algorithms, as they led to the best results in most cases and were very robust with respect to noise. The TR method converged within less iterations than the CG methods, but it needs more function evaluations due to the calculation of the Hessian. Some further improvements were performed in order to state how close to the minimum these methods can get. In the case of all optimisation tasks, the results could be improved significantly by one of the three methods.

So, it was concluded that gradient-based numerical optimisation algorithms are suitable, even if there is noise in the function to be minimised. It is recommended to start with a steepest descent method in order to reach a neighbourhood of the minimum as early as possible. Then, a CG method should be applied and afterwards – as it is too time-consuming before because of the Hessian

computations – the TR method with an exact solution of the subproblem is applied. Only at the last step, one can test the Newton Raphson method because this is the most efficient one in domains which are situated extremely close to the minimum and in which the Hessian can be assumed to be positive definite. Furthermore, it is part of the TR algorithm to use a Newton Raphson step when the radius of the TR is very small and the minimum is situated inside.

The assessment of the numerical optimisation algorithms was a practical proof that gradient-based algorithms can be used for such kinds of optimisation tasks. Hence, they are ready for the application to molecular simulations. In this paper, the first results are shown using MD simulations performed by *GROMACS* and *Moscito*.

3. Simulation details

3.1 Potential and force field parameters

Various intermolecular potentials are available to describe interactions in fluids. In order to test the gradient-based methods and to obtain a more consistent potential model, we decided to test only one widely used intermolecular potential consisting of the (12,6) – LJ and the Coulomb potential. The LJ potential contains two parameters for each atom or group of atoms, i.e. force centre, namely the energetic parameter ϵ and the diameter parameter σ. Let r_{ij} be the distance between two force centres. Then, the LJ potential has the following form:

$$U_{\mathrm{LJ}}(r_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right], \tag{7}$$

where ϵ_{ij} and σ_{ij} are combinations of ϵ_i and ϵ_j, and σ_i and σ_j, respectively, according to the Lorentz-Berthelot mixing rules:

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}, \tag{8}$$

$$\sigma_{ij} = \frac{1}{2}(\sigma_i + \sigma_j). \tag{9}$$

To take electrostatics into account, the common Coulomb potential was applied:

$$U_{\mathrm{Coul}}(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}, \tag{10}$$

where q_i and q_j are the partial atomic charges, r_{ij} is their distance and $\epsilon_0 = 8.854 \times 10^{-12}\,\mathrm{Fm}^{-1}$, the permittivity of vacuum.

Intramolecular potentials are the standard harmonic potentials for bond lengths, angles and dihedrals, which are often equal to 0 as the linear constraint solver for molecular simulations algorithm [53] is used in order to perform MD with constraints. In the case of benzene, the bond lengths, angles and dihedrals are fixed, and in the case of phosgene, only the bond lengths obey constraints. Torsional angles only appear in benzene, as in phosgene not more than three force centres exist.

In this work, the force field parameters to be optimised are ϵ and σ for each force centre. In the case of phosgene, the partial atomic charges are Mulliken charges obtained from a quantum mechanical geometry optimisation via *Gaussian 03*[6] with a HF/STO-3G base set. For benzene, bond lengths, angles, force constants and charges were taken from the optimised potentials for liquid simulations optimised potentials for liquid simulations (OPLS) force field [36] (also the torsional angles) and from [54] for phosgene, except the charges and the force constants for the bond angles, which were taken from the Gromos 43A1 force field [55].

Actually, i.e. in the mathematical sense, the potential parameters cannot be assumed to be independent from each other, as stated by [56]. Some of them are correlated or anticorrelated. Therefore, assuming independence anyway leads to inaccuracies in the total force field. However, a force field is an approximation of reality and its main advantage is its explicability. The origins of the parameters should differ from each other because the course of action should be traceable, explainable and reproducible. Any interdependences between parameters make the force field much more complex, which is not desired, as the computation of a force field as an approximate model must be fast and comprehensible. Moreover, especially for intramolecular forces, many state-of-the-art approaches exist leading to reliable parameters. Hence, the optimisation via GROW must begin when such methods fail.

Table 1 shows the initial force field parameters σ and ϵ for the three molecules considered in this work, which were taken from the OPLS and the Gromos 43A1 force field in the case of benzene and phosgene, respectively.

3.2 System configurations

The systems simulated in this work are isothermal–isobaric ensemble of benzene (500 particles) and phosgene (750 particles). The explicit simulation conditions for each system are summarised in Table 2. Vapour pressure is computed by the Antoine equation indicated in the *NIST Chemistry Webbook* [57]:

$$\log_{10}(P) = A - \frac{B}{T + C}, \tag{11}$$

where the coefficients *A, B* and *C* are calculated from data generated by [58] in the case of benzene. In the case of phosgene, Equation (11) was used as well, whereas the corresponding data were generated by [59].

Table 1. Initial force field parameters for the two considered substances, benzene and phosgene.

Substance	Atom	q (C)	m (u)	σ (nm)	ϵ (kJ/mol)
Benzene	Carbon (C)	−0.1150	12.0110	3.5500	0.2929
	Hydrogen (H)	0.1150	1.00787	2.4200	0.1256
Phosgene	Carbon (C)	0.3027	12.0110	0.3361	0.4059
	Oxygen (O)	−0.1362	15.9948	0.2626	1.7250
	Chlorine (Cl)	−0.0833	35.4530	0.3470	1.2556

The parameters were taken from the OPLS force field [36] and the Gromos 43A1 force field [55] in the case of benzene and phosgene, respectively. Only the LJ parameters σ and ϵ were considered within the optimisation.

Table 2. System settings for the two considered substances, benzene and phosgene.

System parameter	Benzene	Phosgene
Number of molecules	500	750
Temperature range (K)	303	235.65
	313	243.15
	323	250.65
		258.15
		265.65
		273.15
		280.65
Pressure range (bar)	0.16	0.12
	0.25	0.18
	0.32	0.27
		0.39
		0.55
		0.75
		1.01
Pre-pre-equilibration: number of steps	10,000	10,000
Pre-pre-equilibration: step length (ps)	0.002	0.002
Pre-equilibration: number of steps	250,000	50,000
Pre-equilibration: step length (ps)	0.002	0.002
Equilibration cycle: number of steps	250,000	50,000
Equilibration cycle: step length (ps)	0.002	0.002
Number of equilibration cycles:	1	1
Production run: number of steps	500,000	50,000
Production run: step length (ps)	0.002	0.002

Electrostatic interactions were computed using the particle mesh Ewald summation method with a real space cutoff of 0.9 nm and a mesh spacing of approximately 0.12 nm and fourth-order interpolations [60]. For benzene both Coulomb and LJ interactions 1–4 interactions are multiplied by a factor of 0.5. Temperature control was achieved using a Nosé–Hoover thermostat [61,62] and pressure control using the Rahman–Parrinello barostat [63,64] with coupling times $\tau_T = 0.5$ ps and $\tau_p = 0.2$ ps, respectively.

3.3 (Pre-)equilibration and production

An MD simulation workflow, as depicted in Figure 2, has to be organised well in order to provide successful simulations and accurate approximations of the considered physical properties, which are calculated via thermodynamic averages over a certain time period. Therefore, the system has to be *equilibrated*, i.e. specific physical properties are not allowed to be subject to high changes over time. This in turn means that their trend may not increase or decrease with time and that their oscillations must be bounded as strictly as possible. Within GROW, an MD simulation workflow using *GROMACS* is constructed as follows:

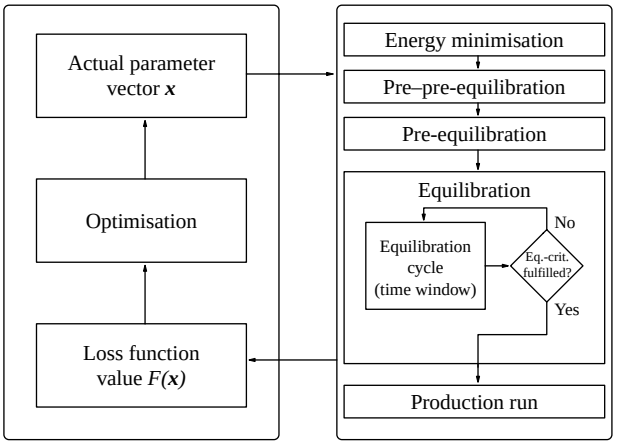

Figure 2. Construction of an MD simulation workflow using *GROMACS* for a parameter vector x within GROW: first, the potential energy surface is minimised so that the energy is not too high already at the beginning; second, a canonical ensemble pre-pre-equilibration is performed which is long enough to guarantee that the system will not enter into the gas phase; third, an isothermal–isobaric ensemble pre-equilibration is executed to move the system closer to equilibrium. Fourth, the equilibration is activated: in each equilibration cycle, a simulation is performed through a time window of a user-defined length and specified properties are monitored. When the equilibration criterion is fulfilled for each of these properties, the system is considered as equilibrated, and the considered physical properties are calculated within a production run which can be significantly shorter than an equilibration cycle. The loss function value $F(x)$ is computed by means of the properties and passed on to the optimisation procedure.

(1) First of all, a local minimum of the potential energy hyper surface is determined, in order to keep the potential energy low already from the beginning. This is done by 2000 iterations of a steepest descent algorithm implemented in *GROMACS*.

(2) Before the actual system equilibration is started, some previous simulations have to be performed for reasons of precaution so that a successful equilibration is guaranteed. Especially in the case of simulations on the vapour–liquid coexistence curve or if the temperature is so high that the considered substance is nearly a gas, a sort of *pre-pre-equilibration* has to be performed in order to make sure that the system does not become gaseous. This is done by quite a small number of canonical steps so that the box volume is kept constant. Dependent on the substance, the time step can also be reduced, e.g. by a factor of 10, so that big molecular jumps are avoided and the system is monitored much more accurately.

(3) A *pre-equilibration* is performed for the conventional reasons as stated for example by [2]: as the system is assumed to be far away from equilibrium at the beginning, there has to be a simulation before, which takes it closer to equilibrium. This is also important because the starting configuration consists of a crystalline structure, as the initial box is a cube. For the small molecules considered in this work, the pre-equilibration is not crucial after having performed the pre-pre-equilibration already. However, as this cannot be generalised, it is performed anyhow. Whenever GROW is applied to larger molecules such as ionic liquids or polymers, pre-equilibration will be a very important step in order to save computation time. The size and the number of time steps has to be defined by the user. In the case of the three substances considered in this paper, both variables were equal to the ones used for equilibration.

(4) An *equilibration* cycle consists of a time window of a user-defined length. During the simulation over this time window, specific physical properties are monitored and tested for equilibration. These always include the potential energy as recommended by [2]. If the density belongs to the physical properties to optimise, it is monitored as well. In the case of the enthalpy of vapourisation, the monitored property is the non-bonded energy consisting of the LJ and the Coulomb potential (Equations (7) and (10), respectively). The diffusion coefficient is not integrated in the equilibration process, as it can only be determined retroactively. This means that it is computed by averages over a certain time. It cannot be computed at a specific time step. Integrating the diffusion coefficient into equilibration would lead to a huge amount of computation time.

The equilibration criterion is formulated as follows: Let X be one of the properties mentioned above and $\mu = \langle X \rangle$ its (real) thermodynamic average. Let M be the total number of time steps within a simulation. It can be proven in a straightforward way that the quadratic error function between the approximation $(1/M)\sum_{i=1}^{M} X_i$, where $X_i, \quad i = 1, \ldots, M$, is independent samples, and μ is equal to the standard deviation of X divided by $\sqrt{M}$:

$$\left\langle \left(\frac{1}{M} \sum_{i=1}^{M} X_i - \mu \right)^2 \right\rangle = \frac{\sigma^M(X)}{\sqrt{M}}. \qquad (12)$$

The standard deviation $\sigma(X)$ can be estimated by the approximation

$$\sigma^M(X) \approx \frac{1}{M-1} \sum\nolimits_{i=1}^{M} \left(X_i - \frac{1}{M} \sum\nolimits_{i=1}^{M} X_i \right)^2,$$

and hence, the following equilibration criterion is taken:

$$\sigma^{\tilde{M}}(X) \leq \tilde{\tau} \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} X_i, \qquad (13)$$

where $\tilde{M}$ is the size of the time window. It is more efficient to use $\tilde{M}$ instead of M because otherwise another equilibration cycle may be started needlessly, as the equilibration criterion may not be fulfilled only because the initial values of X are involved in the calculation. The multiplication by $(1/\tilde{M})\sum_{i=1}^{\tilde{M}} X_i$ is due to the fact that the criterion must be defined relatively and not absolutely. The tolerance parameter $\tilde{\tau}$ depends on the specific property. Thereby, it is important to know which amount of noise in the case of which property can be tolerated by the gradient-based algorithms. Please note that the amount of noise in the simulation data can be measured by the quadratic error function (12). According to [50], $\tilde{\tau} = 0.005$ was set for the density and $\tilde{\tau} = 0.01$ for all energies.

Please note that the $X_i, \quad i = 1, \ldots, M$, must be statistically independent from each other. The independence can be measured by means of the autocorrelation function of X. See [2] for more details. However, in this application, where equilibration is only understood as the accuracy of the thermodynamic averages, it is sufficient to take a sample at each 1000th time step. Hence, instead of $X_i, \quad i = 1, \ldots, M$, only $M/1000$ samples are involved in the average calculations.

(5) When the equilibration criterion (13) is fulfilled for all properties, the *production run* can start in order to compute the physical properties to be optimised. This phase of the simulation workflow does not need to be very long, as the system is already equilibrated, except in the case of the diffusion coefficient, which is determined by a sufficiently long simulation in the production phase. The final properties are calculated

via averages over a few time steps only. Note again that the properties X_i involved in the average calculation must be independent.

Please note that this kind of simulation workflow does not depend on the simulation tool. It can also be constructed with other simulation tools than *GROMACS*. The number of time steps, the time step lengths and the number of equilibration cycles until equilibrium was reached are indicated in Table 2.

3.4 Experimental and simulated physical properties

As already mentioned, the physical properties considered were density (ρ in kg/m^3), enthalpy of vapourisation ($\Delta_v H$ in kJ/mol) and self-diffusion coefficient (D in 10^{-5} cm^2/s). For the two substances, experimental data on the vapour-liquid coexistence curve were taken. Hence, the experimental densities considered here are the saturated liquid densities ρ_l. Please note that the molecular models created by the isothermal–isobaric ensemble MD simulations do not necessarily produce accurate VLE data. Hence, the density is generally denoted by ρ. In the case of benzene, density and self-diffusion coefficient were considered. Experimental values were obtained from [65]. For phosgene, density and enthalpy of vapourisation were optimised simultaneously. The following nonlinear regression formula was taken for the experimental density of phosgene, which is valid at $P = 1$ atm:

$$\rho(T) = a + bT + cT^2. \tag{14}$$

The regression coefficients *a, b* and *c* were determined by [66]. The fact that equation (16) is only valid at air pressure is not problematic, since the influence of the isothermal compressibility on the density is negligible, as it is less than 10^{-3}%. Hence, the error on the density will be in the same order of magnitude as the statistical noise in the simulation data and it is not necessary to perform separate simulations for the density at $P = 1$ atm. The enthalpy of vapourisation of phosgene was calculated by a combination of the Clausius–Clapeyron equation and the Berthold state equation taken from [59]:

$$\Delta_v H(T) = RT^2 \frac{\mathrm{d}\ln P}{\mathrm{d}T}\left[1 - \frac{9PT_c}{128P_cT}\left(1 - 6\frac{T_c^2}{T^2}\right)\right] - \frac{PV_{m,c}}{RT}, \tag{15}$$

where $R = 8.3146\,\mathrm{J\,mol^{-1}\,K^{-1}}$ is the ideal gas constant, $T_c = 455\,\mathrm{K}$ the critical temperature, $P_c = 56$ atm the critical pressure, and $V_{m,c} = 69.2\,\mathrm{cm^3}$ is the molar critical volume. Pressure P is computed by the Antoine equation (11). Please note that for both density and enthalpy of vapourisation, conversions had to be carried out in order to get the correct units.

After the production runs, we computed the simulated values for ρ, $\Delta_v H$ and D are computed as thermodynamic averages over the production time, which are denoted by $\langle\cdots\rangle$. The density is simply computed by

$$\rho = \frac{m_l}{\langle V_l\rangle}, \tag{16}$$

where m_l is the mass and V_l the volume of the liquid phase. The computation of ρ is performed by a program implemented in *GROMACS*, as well as by the non-bonded energy terms from equations (7) and (10). The enthalpy of vapourisation is then approximated by

$$\Delta_v H \approx -\frac{\langle U_{nb}\rangle}{N_{Mol}} + RT = -\frac{\langle U_{LJ} + U_{Coul}\rangle}{N_{Mol}} + RT, \tag{17}$$

as it is assumed that the contribution of the gas phase is negligible, and N_{Mol} stands for the number of molecules within the system. Equation (17) is evaluated after the simulation phase by *GROW*.

For the self-diffusion coefficient, separate simulations were performed by *MOSCITO*, which also computed D using the Einstein relation:

$$D = \frac{1}{6}\lim_{t\to\infty}\frac{\mathrm{d}}{\mathrm{d}t}\langle r(t) - r(0)\rangle^2, \tag{18}$$

where $r(t)$ is the position vector of the centres of mass of the system of particles at time step t, and $\langle r(t) - r(0)\rangle^2$ is the mean square displacement. The latter can only be computed after a certain time window, as it is a thermodynamic average. The self-diffusion coefficient is then calculated by fitting the curves indicating how large is the distance of a molecule from its original position by regression lines. The average of the slopes of these lines divided by 6 equals D.

4. Results

In the two following subsections, practical results concerning simulation and optimisation of the two considered molecules, benzene and phosgene, are indicated, respectively. Please note that in all cases, the goal was to fulfill the stopping criterion $F(x) \leq 10^{-3}$ (Table 3).

4.1 Benzene

Figure 3 shows the different simulation parts mentioned in Section 3 in the case of benzene at $T = 303\,\mathrm{K}$. The LJ parameters used within the simulations are the initial ones indicated in Table 2. Two different physical properties are plotted, which are necessary to decide whether the system is equilibrated or not, according to the equilibration criterion (13), i.e. the density (Figure 3(a)) and the

Table 3. Origin of experimental values for the two considered substances, benzene and phosgene.

Substance	Property	Origin of experimental values
Benzene	ρ, D	From the literature [65]
Phosgene	ρ	Non-linear regression formula (14), Coefficients calculated by [66]
	$\Delta_v H$	Combination of Clausius–Clapeyron equation and Berthold state equation taken from [59]

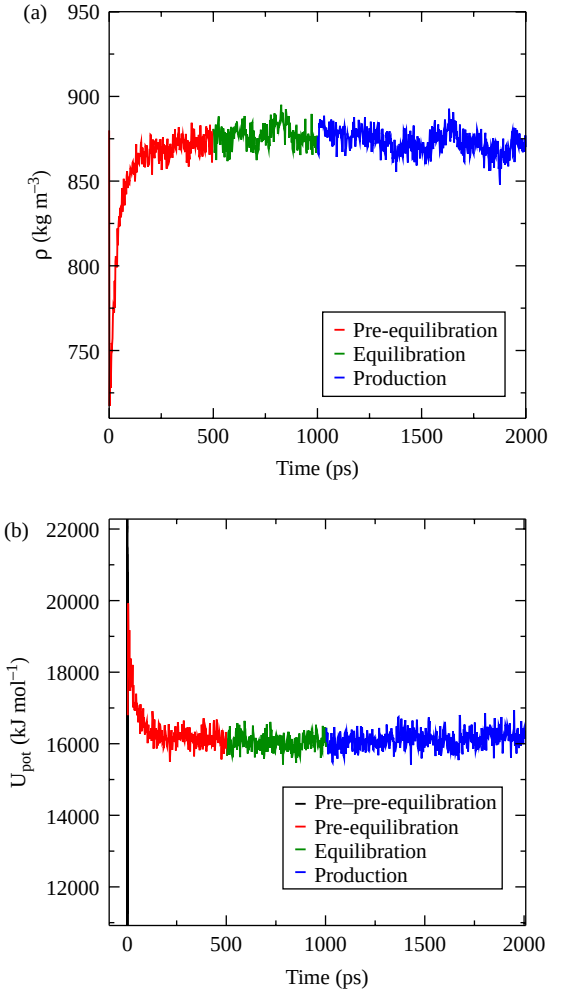

Figure 3. (a) Density and (b) potential energy within the different simulation phases of benzene at $T = 303\,\mathrm{K}$ with the initial LJ parameters indicated in Table 2. In the case of ρ_l, only the pre-equilibration, equilibration and production are plotted. Both properties can be considered as equilibrated at a very early stage.

potential energy (Figure 3(b)). For the former, only the pre-equilibration, the equilibration and the production phase are indicated because within the pre-pre-equilibration, the density is constant. For the latter, the energy minimisation and pre-pre-equilibration phase are very short, as benzene is the simplest system considered in this work and the time needed for equilibration is the shortest. It can be seen that after strong deviations and oscillations at early time steps, both time series reach a zero trend with small deviations, which can be considered as an equilibrated state. Table 4 shows the simulation results of the initial LJ parameters from Table 2, i.e. the simulated and experimental values for ρ_l and D at three different temperatures. The mean absolute percental error (MAPE) over all temperatures between experimental and simulated data is 0.81% for ρ_l and 22.05% for D with a loss function value of 0.1463. Hence, the value of D is still very poor and needs to be improved by all means.

Table 5 shows the optimisation results for benzene obtained by three iterations of the steepest descent method. The decrease of the loss function was in $\mathcal{O}(10^{-2})$. Thereby, the MAPE on the diffusion coefficient decreased drastically (by a factor 10^{-1}) and the MAPE on density was increased only marginally. Following the idea described in [49], the optimisation workflow was interrupted after a certain number of Armijo step length control iterations, because the resulting small enhancement is not worth the much higher amount of computation time. Here, the workflow was stopped after 10 Armijo iterations. As the stopping criterion has not been fulfilled, a subsequently applied Polak Ribière method was tried which did not improve the results. Then, a final trial was performed by decreasing the gradient discretisation parameter h from 10^{-2} to 10^{-3} in Equation (3). The idea behind this was that the loss function, could be smoother in a neighbourhood of the minimum, so that the gradient might be calculated more accurately than in steep domains of the loss functions where the amount of noise is so high that by calculating the gradient with a small h the algorithm gets stuck in a local minimum produced by the oscillations due to statistical noise. For more details, cf. [49]. As none of these trials led to success, $\sigma = 0.3447\,\mathrm{nm}$ and $\epsilon = 0.2882\,\mathrm{kJ/mol}$, i.e. $x^{(2)}$ in Table 5, are considered as the optimal LJ parameters for benzene. Figure 4 shows the development of ρ_l (Figure 4(a)) and D (Figure 4(b)) within the optimisation workflow performed with GROW over the considered

Table 4. Simulated and experimental physical properties for benzene.

T (K)	Sim. ρ_l (kg/m^3)	Exp. ρ_l (kg/m^3)	Sim. D (10^{-5} cm^2/s)	Exp. D (10^{-5} cm^2/s)
303	873.28	869.00	1.88	2.45
313	859.89	875.00	2.26	2.87
323	847.77	846.00	2.64	3.37

Note: The simulated ones were obtained with the initial LJ parameters from Table 2. The MAPE on ρ_l is 0.81% and on D is 22.05%. The value of the loss function is 0.1463.

Table 5. Optimisation results for benzene: within only two iterations of the steepest descent method, the loss function was improved by two orders of magnitude.

Iteration	LJ Pars.	MAPE on ρ_l	MAPE on D	Gradient	Norm of gradient	Loss function
0	0.3550	0.81%	22.05%	17.0692	18.9660	0.1463
	0.2929			8.2676		
1	0.3424	0.85%	7.77%	−9.9735	11.6433	0.0205
	0.2868			−6.0080		
2	0.3447	0.85%	2.15%	−2.7582	3.2193	0.0017
	0.2882			−1.6602		

Note: The MAPE on the diffusion coefficient is one tenth of the original one, whereas the density has fallen off in quality a little. As the Armijo step length control did not converge after 10 iterations and no further improvements were achieved, the LJ parameters of the second iteration, i.e. $x^{(2)}$, are considered as optimal.

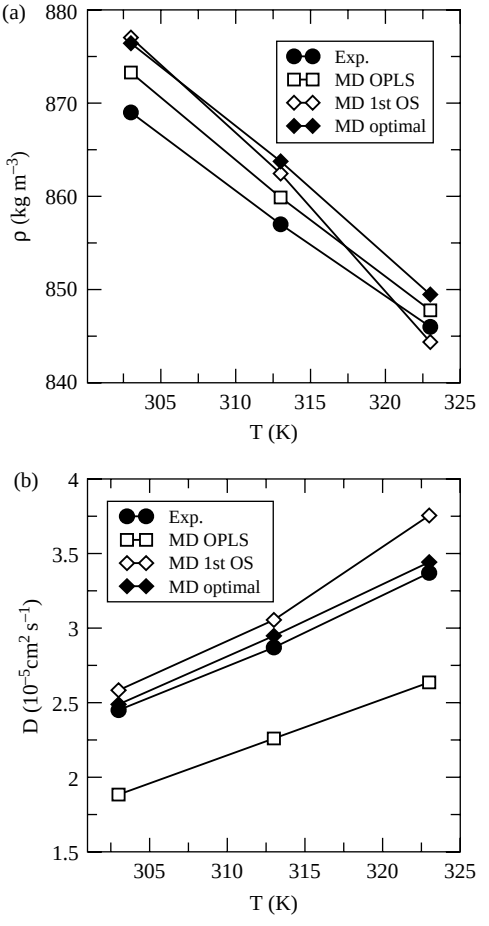

Figure 4. (a) Development of density and (b) diffusion coefficients within the optimisation workflow over the considered temperature range. The OPLS parameters produce diffusion coefficients which are situated below and far away from the experimental ones. After the first optimisation step, the parameters get closer but higher. Within three iterations performed with GROW, the diffusion coefficients are very close to experimental ones. However, the densities get marginally worse.

temperature range: although the densities get marginally worse, all experimental values of ρ_l and D are reproduced very well by GROW, although the stopping criterion has not been fulfilled (Table 6).

Table 6. Simulated and experimental physical properties for phosgene at seven different temperatures.

T (K)	Sim. ρ_l (kg/m^3)	Exp. ρ_l (kg/m^3)	Sim. $\Delta_v H$ (kJ/mol)	Exp. $\Delta_v H$ (kJ/mol)
235.65	1963.64	1502.80	38.12	26.69
243.15	1955.95	1486.92	37.71	26.30
250.65	1947.26	1470.71	37.27	25.92
258.15	1932.10	1454.17	36.71	25.55
265.65	1923.89	1437.32	36.27	25.17
273.15	1911.12	1420.14	35.81	24.80
280.65	1900.88	1402.64	35.35	24.42

Note: The simulated ones were obtained with the initial LJ parameters from Table 2. Here, three optimisation tasks are considered, at one (280.65 K), two (273.15 K, 280.65 K) and all seven different temperatures, respectively. The MAPEs on density are 32.88%, 31.90% and 33.06%, and on enthalpy, they are 42.96%, 42.22% and 44.68%. The values of the loss function are 0.2926, 0.5600 and 2.1675.

4.2 Phosgene

In the case of phosgene, three different optimisation tasks were solved starting with the same initial LJ parameters for each of them (see Table 1). These tasks differed in the number of temperatures considered for the simultaneous optimisation. The gradual improvements of the different optimisation workflows are shown in Table 7 (optimisation task 1: $T = 280.65$ K), Table 8 (optimisation task 2: $T = 280.65$ K and $T = 273.15$ K) and Table 9 (optimisation task 3: all seven temperatures specified in Table 2). All three setups converged towards a minimum of the loss function and the stopping criterion was fulfilled in the case of optimisation tasks 1 and 2. For optimisation task 3, the Armijo step length control did not converge after 30 iterations at $x^{(12)}$. The optimal LJ parameters discussed so far are indicated in Table 10 for all three optimisation tasks. In Figure 5, plots of density (Figure 5(b)) and enthalpy of vapourisation (Figure 5(a)) versus the temperature for different LJ parameters including the optimal ones are shown. A good agreement with the experimental data is shown.

The more the target values were taken into account, the more the iterations were required, each of which needed more individual simulations as well. Restarting GROW with preliminary LJ parameters from an optimisation

Table 7. Optimisation results for phosgene at $T = 280.65$ K: within 15 iterations of the steepest descent method, the loss function was improved by three orders of magnitude.

Iteration	MAPE on ρ_l	MAPE on $\Delta_v H$	Norm of gradient	Loss function
0	32.88%	42.96%	5.35	0.2926
1	25.05%	40.70%	5.20	0.2284
2	18.16%	37.32%	3.89	0.1723
3	12.51%	33.23%	3.12	0.1261
4	7.85%	29.71%	3.08	9.4×10^{-2}
5	4.68%	25.29%	2.06	6.6×10^{-2}
6	1.19%	22.49%	1.64	5.1×10^{-2}
7	0.23%	19.72%	1.31	3.9×10^{-2}
8	1.59%	17.56%	1.08	3.1×10^{-2}
9	2.12%	16.29%	1.05	2.7×10^{-2}
10	4.53%	11.92%	1.15	1.6×10^{-2}
11	5.40%	6.51%	0.96	7.2×10^{-2}
12	3.71%	6.09%	0.70	5.1×10^{-3}
13	1.23%	5.42%	0.40	3.1×10^{-3}
14	0.31%	3.75%	0.27	1.4×10^{-3}
15	0.36%	1.09%	0.07	1.3×10^{-4}

Note: The MAPEs on density and enthalpy of vapourisation were improved drastically and may already be situated within error bars caused by statistical noise. For reasons of brevity, the LJ parameters and the gradient are not indicated. As the stopping criterion $F(x^{(15)}) \leq 10^{-3}$ was fulfilled, the LJ parameters of the 15th iteration, i.e. $x^{(15)} = \left(\sigma_{\mathrm{C}}^{(15)}, \sigma_{\mathrm{O}}^{(15)}, \sigma_{\mathrm{Cl}}^{(15)}, \epsilon_{\mathrm{C}}^{(15)}, \epsilon_{\mathrm{O}}^{(15)}, \epsilon_{\mathrm{Cl}}^{(15)}\right)^{\mathrm{T}} = (0.24818, 0.21626, 0.39596, 0.38681, 1.71429, 1.22114)^{\mathrm{T}}$, are considered as optimal.

Table 8. Optimisation results for phosgene at two temperatures: within 14 iterations of the steepest descent method, the loss function was improved by three orders of magnitude.

Iteration	MAPE on ρ_l	MAPE on $\Delta_v H$	Norm of gradient	Loss function
0	31.90%	42.22%	10.91	0.5600
1	23.40%	38.52%	9.37	0.4063
2	16.84%	35.09%	7.56	0.3030
3	11.49%	31.98%	4.75	0.2310
4	8.58%	28.81%	4.41	0.1808
5	6.56%	24.19%	4.11	0.1254
6	0.74%	17.95%	2.44	6.5×10^{-2}
7	0.50%	14.87%	2.50	4.4×10^{-2}
8	0.10%	12.30%	1.95	3.0×10^{-2}
9	0.43%	9.89%	1.15	2.0×10^{-2}
10	0.08%	8.21%	1.06	1.3×10^{-2}
11	0.38%	6.53%	0.86	8.6×10^{-3}
12	0.52%	5.05%	0.73	5.2×10^{-3}
13	0.45%	2.97%	0.46	1.8×10^{-3}
14	0.32%	0.88%	0.12	1.9×10^{-4}

Note: The MAPEs on density and enthalpy of vapourisation were improved drastically. For reasons of brevity, the LJ parameters and the gradient are not indicated. As the stopping criterion $F(x^{(14)}) \leq 10^{-3}$ was fulfilled, the LJ parameters of the 14th iteration, i.e. $x^{(14)} = \left(\sigma_{\mathrm{C}}^{(14)}, \sigma_{\mathrm{O}}^{(14)}, \sigma_{\mathrm{Cl}}^{(14)}, \epsilon_{\mathrm{C}}^{(14)}, \epsilon_{\mathrm{O}}^{(14)}, \epsilon_{\mathrm{Cl}}^{(14)}\right)^{\mathrm{T}} = (0.25380, 0.20525, 0.39741, 0.38423, 1.71577, 1.22092)^{\mathrm{T}}$, are considered as optimal.

workflow at lower temperatures allowed a significant total speedup. As the number of time steps of the production run was quite low (50,000 ps), the optimal LJ parameters of optimisation task 2 were taken as initial parameters for a further optimisation with a longer production run (2,50,000 ps). Please note that at the beginning of the optimisation, a high accuracy of the simulated physical properties is not crucial because of the high amount of statistical noise. Hence, a speedup may be achieved by reducing the simulation time at the beginning and increasing it again in the neighbourhood of the minimum. Table 11 shows further improvements obtained by this approach. So the stopping criterion was also fulfilled for optimisation task 3.

Table 9. Optimisation results for phosgene at seven temperatures: within 12 iterations of the steepest descent method, the loss function was improved by three orders of magnitude.

Iteration	MAPE on ρ_l	MAPE on $\Delta_v H$	Norm of gradient	Loss function
0	33.06%	44.68%	8.52	2.1675
1	31.62%	37.34%	37.48	1.6803
2	23.82%	34.98%	31.70	1.2472
3	17.55%	32.30%	25.26	0.9492
4	2.73%	19.09%	9.35	0.2631
5	3.35%	14.63%	6.41	0.1600
6	2.80%	11.24%	4.76	9.7×10^{-2}
7	2.68%	8.51%	3.79	5.8×10^{-2}
8	2.19%	6.41%	2.79	3.4×10^{-2}
9	1.33%	4.41%	1.89	1.7×10^{-2}
10	0.91%	2.74%	1.17	7.8×10^{-3}
11	0.76%	1.46%	0.64	3.6×10^{-3}
12	0.71%	1.29%	0.17	2.1×10^{-3}

Note: The MAPEs on density and enthalpy of vapourisation were improved drastically. In order to speed up the optimisation process, the 10th iteration of optimisation task 1 was taken for $x^{(4)}$. For reasons of brevity, the LJ parameters and the gradient are not indicated. As the Armijo step length control did not converge after 30 iterations, the LJ parameters of the 12th iteration, i.e. $x^{(12)} = \left(\sigma_{\mathrm{C}}^{(12)}, \sigma_{\mathrm{O}}^{(12)}, \sigma_{\mathrm{Cl}}^{(12)}, \epsilon_{\mathrm{C}}^{(12)}, \epsilon_{\mathrm{O}}^{(12)}, \epsilon_{\mathrm{Cl}}^{(12)}\right)^{\mathrm{T}} = (0.27155, 0.23388, 0.40498, 0.39136, 1.71689, 1.23150)^{\mathrm{T}}$, are considered as optimal.

Comparing the loss function values of the three optimisation tasks, we could identify an increase of this value with an increasing number of temperature values, i.e. target values, which correlates with the number of terms in the loss function (1). This has been observed in [50] as well. Focusing on the other hand on the benchmark of the different physical properties to be optimised, i.e. considering the MAPE on ρ and $\Delta_v H$, we could observe that the ratio between these two was shifted considerably during the iterations. The enthalpy of vapourisation and the density started roughly with the same MAPE but one of the densities decreased much faster. This ratio was conserved until the end of the optimisation indicating that the density was much easier to fit than the enthalpy of vapourisation.

Finding a good agreement with the target properties in different optimisation tasks does not necessarily lead to identical LJ parameters. The six optimised LJ parameters of the different optimisation tasks differ from 0.2% (σ_{O}) to 12.2% (ϵ_{O}). The most similar parameters were obtained for chlorine parameters. In this case, a deviation of 2% for σ and 1% for ϵ was achieved. This gives the hint that

Table 10. Optimal LJ parameters in the case of phosgene for the three different optimisation tasks: 1 means one temperature, 2 means two temperatures and 3 means seven temperatures.

Optimisation task	σ_C (nm)	σ_O (nm)	σ_{Cl} (nm)	ϵ_C (kJ/mol)	ϵ_O (kJ/mol)	ϵ_{Cl} (kJ/mol)
1	0.24818	0.21626	0.39596	0.38681	1.71429	1.22114
2	0.25380	0.20525	0.39741	0.38423	1.71577	1.22092
3	0.27155	0.23388	0.40498	0.39136	1.71689	1.23150

Table 11. Further optimisation results for phosgene at seven temperatures achieved by a longer production run.

Iteration	MAPE on ρ_l	MAPE on $\Delta_v H$	Norm of gradient	Loss function
0	1.20%	2.03%	1.58	6.9×10^{-3}
1	0.77%	1.42%	0.57	2.2×10^{-3}
2	0.65%	0.68%	0.25	8.9×10^{-4}

The initial LJ parameters were the optimal ones from optimisation task 2 (see Table 10). Within only two iterations of the steepest descent method, the loss function was improved again by one order of magnitude. The MAPEs on density and enthalpy of vapourisation were improved as well. For reasons of brevity, the LJ parameters and the gradient are not indicated. As the stopping criterion $F(x^{(2)}) \leq 10^{-3}$ was fulfilled, the LJ parameters of the second iteration, i.e. $x^{(2)} = \left(\sigma_C^{(2)}, \sigma_O^{(2)}, \sigma_{Cl}^{(2)}, \epsilon_C^{(2)}, \epsilon_O^{(2)}, \epsilon_{Cl}^{(2)}\right)^T = (0.24846, 0.20142, 0.39833, 0.38309, 1.7155, 1.2190)^T$, are the optimal LJ parameters for phosgene achieved in this work.

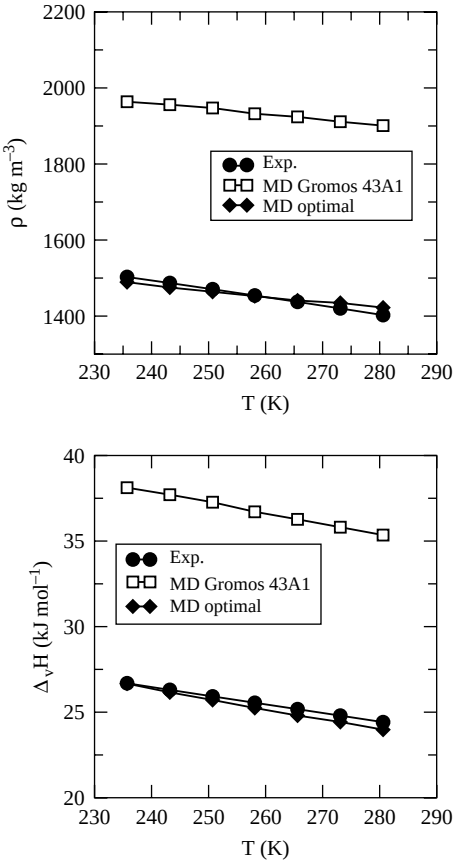

Figure 5. (a) Development of density and (b) enthalpy of vapourisation within the optimisation workflow performed by GROW (optimisation task 3). The Gromos 43A1 parameters produce properties which are situated far away from the experimental ones, whereas the properties obtained by a simulation with the optimal LJ parameters from Table 10 are very close to experimental ones.

chlorine has the most dominating LJ parameters in the current optimisation setups.

Another aspect influencing the optimal parameters is the choice of the target values. As a given force field cannot be perfect, parameterisation generally takes place at a chosen reference state. Altering these conditions may lead to inaccuracies in the predictions of the optimised force field. In the given setup, the reference state is clear for the optimisation with only one target temperature. Comparing the results of the optimisation at seven temperatures shows that the best agreement is found at the centre of the target values space (cf. Figure 5), which indicates that optimal calculation is achieved at a temperature of $T \approx 258\,\mathrm{K}$. This shift in the actual optimisation temperature influences the final parameters, cf. Figure 6.

The figure shows another feature of the optimisation: with the current simulation setup and the variation of these variables, it is obviously not possible to model a phosgene system over a wide temperature range. Optimisation showed what it can do at best. For further improvements, the model itself has to be evaluated. This includes the optimisation of different and more variables (e.g. partial charges), the consideration of other potentials or even other optimisation techniques.

5. Conclusion

In this work, the automated GROW was applied for the first time to molecular simulations. Before, the practical problem of providing a suitable program toolkit and the scientific problem to choose an efficient mathematical optimisation algorithm were treated separately. The main objective of this study was to utilise GROW in a first real application situation, especially characterised by the fact that unpredictable statistical noise would severely influence the course of the optimisation. Thereby, MD simulations were performed using the open-source software tools *GROMACS* and *Moscito* for each function evaluation within the optimisation workflow. During the optimisation procedure, a quadratic loss function had to be

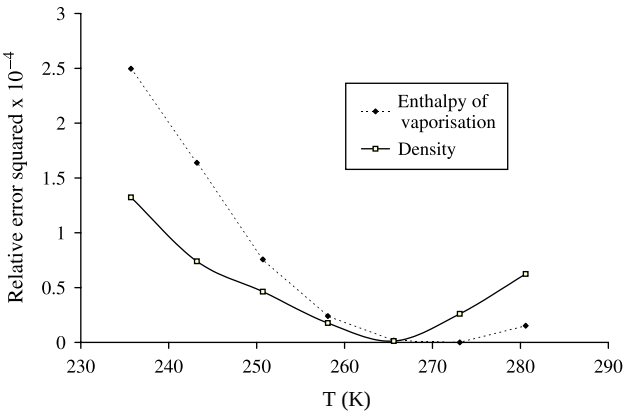

Figure 6. Squared relative errors on density and enthalpy of vapourisation over all seven temperatures in the case of phosgene. The curves are achieved by a simple spline interpolation. The plot shows that the best agreement between simulation and experiment is found at the centre of the target value space, i.e. at $T \approx 258$ K.

minimised. That the function contained the relative errors between simulated and experimental physical properties. The target properties considered in this work were the density, the enthalpy of vapourisation and the self-diffusion coefficient at different temperatures. The force field parameters to be fitted were the LJ parameters σ and ϵ, while the partial charges were pre-defined by quantum chemical calculations. The systems chosen were isothermal–isobaric ensemble of the small molecules benzene and phosgene, which were simple enough for the first application but challenging enough to say that the results achieved are precious not only from the mathematical but also from the physical point of view. Different optimisation tasks were considered with different physical properties and at different temperatures. The stopping criterion selected in this work was that the loss function reached some specified threshold τ (here, $\tau = 10^{-3}$). In most cases, this stopping criterion was fulfilled within a few iterations only, starting with LJ parameters from standard force fields. This shows the high performance of the gradient-based methods implemented in GROW. Hence, it could be proven that GROW is applicable to molecular simulations and can also handle the statistical noise appearing in the simulation data. The two main drawbacks are the quite high number of simulations required at each iteration (at least $N + 1$, i.e. one for the iteration itself and N for the gradient plus the evaluations required for the Armijo step length control) and the strong dependence on the initial force field parameters. Therefore, future work will focus on the augmentation of efficiency and the conceptual design of a global optimisation workflow which finds suitable initial parameters for GROW situated in the sphere of influence of a global minimum. Furthermore, scientifically and industrially more challenging substances will be considered in order to develop new force fields of high importance for both the scientific community and the industry. In this regard, different potential parameters, e.g. partial atomic charges, will also be included in the optimisation and different optimisation techniques will be considered.

Acknowledgements

We are grateful to Astrid Maaß, Axel Arnold, Karl N. Kirschner, Florian Müller-Plathe and Jadran Vrabec for valuable discussions and appreciate their intellectual support to our work. Marco Hülsmann acknowledges the financial support for his scholarship from the University of Cologne (Germany).

Notes

1. Email: marco.huelsmann@scai.fraunhofer.de
2. Email: t.mueller@theo.chemie.tu-darmstadt.de
3. Email: thorsten.koeddermann@scai.fraunhofer.de
4. http://www.gromacs.org
5. http://ganter.chemie.uni-dortmund.de/MOSCITO/
6. http://www.gaussian.com

References

[1] S.J. Singer and G.L. Nicolson, *The fluid mosaic model of the structure of cell membranes*, Science 175 (1972), pp. 720–731.
[2] M.P. Allen and D.J. Tildesley, *Computer Simulations of Liquids*, Oxford Science, Oxford, 1987.
[3] Y. Zhou and G. Stell, *Chemical association in simple models of molecular and ionic fluids. II. Thermodynamic properties*, J. Chem. Phys. 96 (1992), pp. 1504–1506.
[4] J.I. Siepmann, S. Karaborni, and B. Smit, *Simulating the critical behavior of complex fluids*, Nature 365 (1993), pp. 330–332.
[5] S. O'Connell and P.A. Thompson, *Molecular dynamics – continuum hybrid computations: A tool for studying complex fluid flows*, Phys. Rev. E 52 (1995), pp. 5792–5795.
[6] J. Kolafa, I. Nezbeda, and M. Lisal, *Effect of short- and long-range forces on the properties of fluids. III. Dipolar and quadrupolar fluids*, Mol. Phys. 99 (2001), pp. 1751–1764.
[7] R. Valiullin, S. Naumov, P. Galvosas, J. Kärger, H.-J. Woo, F. Porcheron, and P.A. Monson, *Exploration of molecular dynamics during transient sorption of fluids in mesoporous materials*, Nature 443 (2006), pp. 965–968.
[8] I.P. Batra, B.I. Bennett, and F. Herman, *Simple molecular model for crystalline tetrathiofulvalene-tetracyanoquinodimethane (TTF-TCNQ)*, Phys. Rev. B 11 (1975), pp. 4927–4934.
[9] T.P. Fehlner, *Molecular models of solid state metal boride structures*, J. Solid State Chem. 154 (2000), pp. 110–113.
[10] C.N. Della and S. Dongwei, *Mechanical properties of carbon nanotubes reinforced ultra high molecular weight polyethylene*, Diffus. Defect Data Pt. B Solid State Phenom. 136 (2008), pp. 45–48.
[11] S.-T. Lin, M. Blanco, and W.A. Goddard III, *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 (2003), pp. 11792–11805.
[12] D.E. Bien and V.A. Chiriac, *A novel molecular approach to modeling phase change in micro-fluidic systems*, 9th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 1–4 June 2004, Las Vegas, USA, IEEE, NJ, USA, 2004, pp. 598–604.
[13] J. Vrabec and J. Gross, *Vapor–liquid equilibria simulation and an equation of state contribution for dipole–quadrupole interactions*, J. Phys. Chem. B 112 (2008), pp. 51–60.

[14] A. Hodgkin and A. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, J. Physiol. 117 (1952), pp. 500–544.

[15] B.J. Barkla and O. Pantoja, *Physiology of ion transport across the tonoplast of higher plants*, Annu. Rev. Plant Physiol. Plant Mol. Biol. 47 (1996), pp. 159–184.

[16] M. Levitt and I. Warshe, *Computer simulation of protein folding*, Nature 253 (1975), pp. 694–696.

[17] J. Gsponer and A. Caflisch, *Molecular dynamics simulations of protein folding from the transition state*, Proceedings of the National Academy of Sciences, USA (2002), pp. 6719–6724.

[18] C.D. Snow, E.J. Sorin, Y.M. Rhee, and V.S. Pandel, *How well can simulation predict protein folding kinetics and thermodynamics?* Annu. Rev. Biophys. Biomol. Struct. 34 (2005), pp. 43–69.

[19] F. Müller-Plathe and D. Reith, *Cause and effect in reversed in non-equilibrium molecular dynamics: An easy route to transport coefficients*, Comput. Theor. Polym. Sci. 9 (1999), pp. 203–209.

[20] P. Bordat, D. Reith, and F. Müller-Plathe, *The influence of interaction details on the thermal diffusion in binary Lennard-Jones liquids*, J. Chem. Phys. 115 (2001), pp. 8978–8982.

[21] G. Guevara-Carrion, C. Nieto-Draghi, J. Vrabec, and H. Hasse, *Prediction of transport properties by molecular simulation: Methanol and ethanol and their mixture*, J. Phys. Chem. B 112 (2008), pp. 16664–16674.

[22] G.S. Grest and K. Kremer, *Molecular dynamics simulation for polymers in the presence of a heat bath*, Phys. Rev. A 33 (1986), pp. 3628–3631.

[23] F. Müller-Plathe, *Permeation of polymers – A computational approach*, Acta Polym. 45 (1994), pp. 259–293.

[24] K. Binder, *Monte Carlo and Molecular Dynamics Simulations in Polymer Science*, Oxford University Press, Oxford, 1995.

[25] K. Kremer and F. Müller-Plathe, *Multiscale simulation in polymer science*, Mol. Simul. 28 (2002), pp. 729–750.

[26] M. Praprotnik, C. Junghans, L. Delle Site, and K. Kremer, *Simulation approaches to soft matter: Generic statistical properties vs. chemical details*, Comput. Phys. Commun. 179 (2008), pp. 51–60.

[27] W.L. Jorgensen, D.S. Maxwell, and J. Tirado-Rives, *Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids*, J. Am. Chem. Soc. 118 (1996), pp. 11225–12236.

[28] J. Wang and P.A. Kollman, *Automatic parameterization of force field by systematic search and genetic algorithms*, J. Comput. Chem. 22 (2001), pp. 1219–1228.

[29] J. Wang, R.M. Wolf, J.W. Caldwell, P.A. Kollman, and D.A. Case, *Development and testing of a general Amber force field*, J. Comput. Chem. 25 (2004), pp. 1157–1174.

[30] D. Yin and A.D. MacKerell Jr, *Combined* ab initio*/empirical approach for optimization of Lennard-Jones parameters*, J. Comput. Chem. 19 (1998), pp. 334–348.

[31] I.J. Chen, D. Yin, and A.D. MacKerell Jr, *Combined* ab initio*/empirical approach for optimization of Lennard-Jones parameters for polar-neutral compounds*, J. Comput. Chem. 23 (2002), pp. 199–213.

[32] N. Foloppe and A.D. MacKerell Jr, *All-atom empirical force field for nucleic acids: I. Parameter optimization based on small molecule and condensed phase macromolecular target data*, J. Comput. Chem. 21 (2000), pp. 86–104.

[33] B. Eckl, J. Vrabec, and H. Hasse, *Set of molecular models based on quantum mechanical* ab-initio *calculations and thermodynamic data*, J. Phys. Chem. B 112 (2008), pp. 12710–12721.

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

[35] A. Maaß, L. Nikitina, T. Clees, K.N. Kirschner, and D. Reith, *Multi-objective optimization on basis of random models for ethylene oxide*, Mol. Simulat (in press).

[36] W.L. Jorgensen, J.D. Madura, and C.J. Swensen, *Optimized intermolecular potential functions for liquid hydrocarbons*, J. Am. Chem. Soc. 106 (1984), pp. 6638–6646.

[37] M.G. Martin and J.I. Siepmann, *Transferable potentials for phase equilibria. 1. United-atom description of n-alkanes*, J. Phys. Chem. B 102 (1998), pp. 2569–2577.

[38] R. Faller, H. Schmitz, O. Biermann, and F. Müller-Plathe, *Automatic parameterization of force fields for liquids by simplex optimization*, J. Comput. Chem. 20 (1999), pp. 1009–1017.

[39] P. Ungerer, C. Beauvais, J. Delhommelle, A. Boutin, B. Rousseau, and A.H. Fuchs, *Optimization of the anisotropic united atoms intermolecular potential for n-alkanes*, J. Phys. Chem. 112 (2000), pp. 5499–5510.

[40] E. Bourasseau, M. Haboudou, A. Boutin, A.H. Fuchs, and P. Ungerer, *New Optimization method for intermolecular potentials: Optimization of a new anisotropic united atoms potential for olefins: Prediction of equilibrium properties*, J. Chem. Phys. 118 (2003), pp. 3020–3034.

[41] J. Stoll, J. Vrabec, and H. Hasse, *A set of molecular models for carbon monoxide and halogenated hydrocarbons*, J. Chem. Phys. 119 (2003), pp. 11396–11407.

[42] D. Reith, M. Pütz, and F. Müller-Plathe, *Deriving effective mesoscale potentials from atomistic simulations*, J. Comput. Chem. 24 (2003), pp. 1624–1636.

[43] C. Oostenbrink, A. Villa, A.E. Mark, and W.F. van Gunsteren, *A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6*, J. Comput. Chem. 25 (2004), pp. 1656–1676.

[44] H. Sun, *Prediction of fluid densities using automatically derived VDW parameters*, Fluid Phase Equilibr. 217 (2004), pp. 59–76.

[45] K.N. Kirschner, A.B. Yongye, S.M. Tschampel, J. Gonzalez-Outeirino, C.R. Daniels, B.L. Foley, and R. Woods, *GLYCAM06: A generalizable biomolecular force field. Carbohydrates*, J. Comput. Chem. 29 (2008), pp. 622–655.

[46] B. Eckl, J. Vrabec, and H. Hasse, *On the application of force fields for predicting a wide variety of properties: Ethylene oxide as an example*, Fluid Phase Equilibr. 274 (2008), pp. 16–26.

[47] J.A. Nelder and R.A. Mead, *A simplex method for function minimization*, Comput. J. 7 (1965), pp. 308–313.

[48] D. Reith, H. Meyer, and F. Müller-Plathe, *CG-OPT: A software package for automatic force field design*, Comput. Phys. Commun. 148 (2002), pp. 299–313.

[49] M. Hülsmann, T. Köddermann, J. Vrabec, and D. Reith, *GROW: A gradient-based optimization workflow for the automated development of molecular models*, Comput. Phys. Commun. 181 (2010), pp. 499–513.

[50] M. Hülsmann, J. Vrabec, A. Maaß, and D. Reith, *Assessment of numerical optimization algorithms for the development of molecular models*, Comput. Phys. Commun. 181 (2010), pp. 887–905.

[51] J. Stoll, J. Vrabec, H. Hasse, and J. Fischer, *Comprehensive study of the vapour–liquid equilibria of the pure two-centre Lennard-Jones plus point quadrupole fluid*, Fluid Phase Equilibr. 179 (2001), pp. 339–362.

[52] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[53] B. Hess, H. Bekker, H.C. Berendsen, and J.G.E.M. Fraaije, *LINCS: A linear constraint solver for molecular simulations*, J. Comput. Chem. 18 (1997), pp. 1463–1472.

[54] M. Nakata, K. Kohata, T. Fukuyama, and K. Kuchitsu, *Molecular structure of phosgene as studied by gas electron diffraction and microwave spectroscopy*, J. Mol. Spectrosc. 83 (1980), pp. 105–117.

[55] W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hünenberger, P. Krüger, A.E. Mark, W.R.P. Scott, and I.G. Tironi, *Biomolecular Simulation: The GROMOS96 Manual and User Guide*, VDF Hochschulverlag AG ETH Zürich, Zürich/Groningen, 1996, p. II/36.

[56] P.H. Hünenberger and W.F. van Gunsteren, *Empirical classical interaction functions for molecular simulation*, in *Computer Simulation of Biomolecular Systems, Theoretical and Experimental Applications Vol. 3*, W.F. van Gunsteren, P.K. Weiner, and A.J. Wilkinson, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 3–82.

[57] NIST Chemistry Webbook *National Institute of Standards and Technology, USA*, Available at http://webbook.nist.gov/chemistry/

[58] D.D. Deshpande and M.V. Pandya, *Thermodynamics of binary solutions. Part 2. Vapour pressures and excess free energies of aniline solutions*, Trans. Faraday Soc. 63 (1967), pp. 2149–2157.
[59] W.F. Giauque and W.M. Jones, *Carbonyl chloride. Entropy. Heat capacity. Vapor pressure. Heats of fusion and vaporization. Comments on solid sulfur dioxide structure*, J. Am. Chem. Soc. 70 (1948), pp. 120–124.
[60] U. Essmann, L. Perera, M.L. Berkowitz, T.A. Darden, H. Lee, and L.G. Pedersen, *A smooth particle mesh Ewald method*, J. Chem. Phys. 103 (1995), pp. 8577–8593.
[61] S. Nosé, *A molecular dynamics method for simulating in the canonical ensemble*, Mol. Phys. 52 (1984), pp. 255–268.
[62] W.G. Hoover, *Canonical dynamics: Equilibrium phase space distributions*, Phys. Rev. A 31 (1985), pp. 1695–1697.
[63] M. Parrinello and A. Rahman, *Polymorphic transitions in single crystals: A new molecular dynamics method*, J. Appl. Phys. 52 (1981), pp. 7180–7182.
[64] S. Nosé and M.L. Klein, *Constant pressure molecular dynamics for molecular systems*, Mol. Phys. 50 (1983), pp. 1055–1076.
[65] K. Yoshida, N. Matubayasi, and M. Nakahara, *Self-diffusion coefficients for water and organic solvents at high temperatures along the coexistence curve*, J. Chem. Phys. 129 (2008), pp. 214501–214509.
[66] C.N. Davies, *The density and thermal expansion of liquid phosgene*, J. Chem. Phys. 14 (1945), pp. 48–49.