Robin Fingerhut, Gabriela Guevara-Carrion, Isabel Nitzke, Denis Saric, Joshua Marx, Kai Langenbach, Sergei Prokopev, David Celný, Martin Bernreuther, Simon Stephan, Maximilian Kohns, Hans Hasse, Jadran Vrabec

ms2: A molecular simulation tool for thermodynamic properties, release 4.0

Fingerhut, R., Guevara-Carrion, G., Nitzke, I., Saric, D., Marx, J., Langenbach, K., Prokopev, S., Celný, D., Bernreuther, M., Stephan, S., Kohns, M., Hasse, H., & Vrabec, J. (2021). ms2: A molecular simulation tool for thermodynamic properties, release 4.0. Computer Physics Communications, 262, 107860. https://doi.org/10.1016/j.cpc.2021.107860

Accepted manuscript of: Fingerhut, R., Guevara-Carrion, G., Nitzke, I., Saric, D., Marx, J., Langenbach, K., Prokopev, S., Celný, D., Bernreuther, M., Stephan, S., Kohns, M., Hasse, H., & Vrabec, J. (2021). ms2: A molecular simulation tool for thermodynamic properties, release 4.0. Computer Physics Communications, 262, 107860. https://doi.org/10.1016/j.cpc.2021.107860

*ms*2: A molecular simulation tool for thermodynamic properties, release 4.0

Robin Fingerhut[a], Gabriela Guevara-Carrion[a], Isabel Nitzke[a], Denis Saric[a], Joshua Marx[b], Kai Langenbach[b], Sergei Prokopev[c], David Celný[d], Martin Bernreuther[e], Simon Stephan[b], Maximilian Kohns[b], Hans Hasse[b], Jadran Vrabec[a,*]

[a] *Thermodynamics and Process Engineering, Technical University Berlin, 10587 Berlin, Germany*
[b] *Laboratory of Engineering Thermodynamics, University Kaiserslautern, 67653 Kaiserslautern, Germany*
[c] *Computational Fluid Dynamics Laboratory, Institute of Continuous Media Mechanics UB RAS, 614013 Perm, Russia*
[d] *Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 11519 Prague, Czech Republic*
[e] *High Performance Computing Center Stuttgart (HLRS), 70550 Stuttgart, Germany*

Abstract

A new version release (4.0) of the molecular simulation tool *ms*2 (Deublein *et al.*, 2011; Glass *et al.*, 2014; Rutkai *et al.*, 2017) is presented. Version 4.0 of *ms*2 features two additional potential functions to address the repulsive and dispersive interactions in a more versatile way, i.e. the Mie potential and the Tang-Toennies potential. This version further introduces Kirkwood-Buff integrals based on radial distribution functions, which allow the sampling of the thermodynamic factor of mixtures with up to four components, orientational distribution functions to elucidate mutual configurations of neighboring molecules, thermal diffusion coefficients of binary mixtures for heat, mass as well as coupled heat and mass transport, Einstein relations to sample transport properties with an alternative to the Green-Kubo formalism, dielectric constant of non-polarizable fluid models, vapor-liquid equilibria relying on the second virial coefficient and cluster criteria to identify nucleation.

Keywords: Molecular simulation; Molecular dynamics; Monte Carlo

*Corresponding author.
E-mail address: vrabec@tu-berlin.de

New version programm summary

*Program Title: ms*2

Program Files doi: http://dx.doi.org/10.17632/nsfj67wydx.3

Licensing provisions: CC by NC 3.0

Programming language: Fortran95

Supplemental material: A detailed description of the parameter setup for the introduced methods, properties, functionalities etc. is given in the supplemental material. Furthermore, all molecular force field models developed by our group are provided by the MolMod Database: Stephan *et al.*, Mol. Sim. 45 (2019) 806

Journal reference of previous version: Deublein *et al.*, Comput. Phys. Commun. 182 (2011) 2350 and Glass *et al.*, Comput. Phys. Commun. 185 (2014) 3302 and Rutkai *et al.*, Comput. Phys. Commun. 221 (2017) 343

Does the new version supersede the previous version?: Yes

Reasons for the new version: Introduction of new features as well as enhancement of computational efficiency

Summary of revisions: Two new potential functions to address repulsive and dispersive interactions (Mie and Tang-Toennies potential), new properties (Helmholtz energy, Kirkwood-Buff integrals, thermodynamic factor, thermal diffusion coefficients, dielectric constant, mean-squared displacement and non-Gaussian parameter), new functionalities (Kirkwood-Buff integration with extrapolation to the thermodynamic limit, van der Vegt correction for the radial distribution function, orientational distribution function, Einstein relations, vapor-liquid equilibria estimations, cluster criteria to identify nucleation).

Nature of problem: Calculation of application-oriented thermodynamic properties: vapor-liquid equilibria of pure fluids and multi-component mixtures, thermal, caloric and entropic data as well as transport properties and data on microscopic structure

Solution method: Molecular dynamics, Monte Carlo, various ensembles, Grand equilibrium method, Green-Kubo formalism, Einstein formalism, Lustig formalism, OPAS method, Smooth-particle mesh Ewald summation

1. Introduction

Significant increases in computing power have led to a broader usage of molecular modeling and simulation, which simultaneously widens the ability to tackle challenges in physics, chemistry and engineering in a sound and detailed manner. Over the last decades, it has often been shown that these computer-based methods may predict physical reality very successfully. Thus, the long-standing

obstacle of sparse or lacking experimental information on thermophysical data can be overcome by trustworthy and rapid predictions with massively-parallel high performance computing (HPC) hardware and scalable codes.

The program *ms*2 (molecular simulation 2) was developed to compute thermophysical equilibrium properties of pure fluids and mixtures with Monte Carlo (MC) or molecular dynamics (MD) simulations that are both implemented in a single source code. Licenses are freely available for all purposes which concern academic research and teaching under www.ms-2.de together with a substantial set of molecular force field models [1]. *ms*2 [2, 3, 4] supports the microcanonical (NVE), canonical (NVT), isobaric-isenthalpic (NpH), isobaric-isothermal (NpT) and grand canonical (μVT) ensembles as well as the simulation of vapor-liquid equilibria (VLE) with the Grand equilibrium method. Moreover, *ms*2 facilitates the sampling of numerous thermodynamic bulk properties, including transport data, like Maxwell-Stefan (MS) and Fick diffusion coefficients, for molecular models consisting of Lennard-Jones (LJ) interaction sites, point charges, point dipoles and point quadrupoles. It allows for the sampling of the chemical potential with Widom's particle insertion and thermodynamic integration as well as osmotic pressure, hydrogen bond statistics and other features. Next to these thermophysical properties, it was focused on an efficient parallelization of *ms*2 using the message passing interface (MPI), open multi-processing (OpenMP) and its hybrid form (MPI+OpenMP).

There is a series of molecular simulation tools, such as CHARMM, DL_POLY, ESPResSo, GIBBS, GROMACS, IMD, LAMMPS, ls1 mardyn, NAMD, TINKER or Towhee, that is being developed for a range of communities. Both industrial and academic users are addressed by *ms*2 with a focus on applications of molecular modeling and simulation in process and energy engineering. In contrast to most of the tools listed above, *ms*2 is limited to rigid force field models which are appropriate for small molecular species only. However, the implementation of the internal degrees of freedom into *ms*2 is underway for some time in an unpublished version of *ms*2 [5].

Aiming at high accuracy and short response time, *ms*2 is characterized by the large variety of properties that are sampled on the fly. This user-friendly design was extended by the ability to concurrently sample an arbitrary number of state points in one program execution. Concurrent sampling was optimized in the present *ms*2 version such that communication between ensembles was removed entirely. Combining this with its dedication to generate large sets of Helmholtz energy derivative data for the development of equations of state [6], *ms*2 is very much suited to be executed on HPC infrastructure.

A more versatile molecular model development was prioritized in this version release 4.0 such that the traditional LJ 12-6 potential was generalized to the Mie

potential and the more complex Tang-Toennies potential [7] was introduced. Consequently, the basis of molecular modeling and simulation may be improved by a more accurate description of the repulsive and dispersive interactions. Moreover, $ms2$ is now able to yield the Fick diffusion coefficient of mixtures constituted by up to four components due to the concurrent sampling of the MS diffusion coefficient and Kirkwood-Buff integrals (KBI) [8] that give access to the thermodynamic factor [9, 10, 11]. Additionally, with this release, more rapid VLE estimations can be made by carrying out a single NpT ensemble simulation sampling the chemical potential of the liquid and using the second virial coefficient for the vapor. These and further new features were implemented in the source code and the toolset provided at www.ms-2.de. The present work discusses the fourth major release of $ms2$ and its most important innovations, which are presented in the following sections.

2. Mie potential

Addressing repulsive and dispersive interactions in a more versatile way with $ms2$, the standard LJ 12-6 potential function was generalized with the Mie potential function [12]. The pairwise interaction between different sites i and j in a distance r_{ij} is modeled by

$$u_{ij}(r_{ij}) = \frac{n}{n-m}\left(\frac{n}{m}\right)^{m/(n-m)} \cdot \varepsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{n} - \left(\frac{\sigma}{r_{ij}}\right)^{m}\right], \tag{1}$$

where σ and ε are the Mie size and energy parameters, respectively, and n, m are the repulsive and dispersive exponents.

In $ms2$, the interactions between two different Mie sites are described by the Lorentz-Berthelot combining rules for pure components, while for mixtures the modified Lorentz-Berthelot rules are applied [2]. The unlike repulsive and dispersive exponents are determined according to Lafitte *et al.* [13] by

$$k_{ij} = 3 + \sqrt{(k_i - 3)(k_j - 3)} \qquad \text{for } k = n, m\,. \tag{2}$$

Long-range interactions beyond the cutoff radius r_c are considered analytically with the angle averaging formalism derived for the Mie potential by Lustig [14].

The generalization from LJ 12-6 to Mie was introduced throughout the entire code so that all properties and functionalities are accessible with it.

3. Tang-Toennies potential

To describe the intermolecular interactions, an additional potential function based on the work of Tang and Toennies (TT) [7] was introduced into $ms2$. Con-

siderations for its use are outlined in the following, whereas proceedings which are the same as for the Mie potential are not discussed in detail.

The pairwise interaction between different sites i and j in a distance r_{ij} is modeled by

$$u_{ij}(r_{ij}) = A_{ij}\exp(-\alpha_{ij}r_{ij}) - f_6(r_{ij})\frac{C_{6,ij}}{r_{ij}^6} - f_8(r_{ij})\frac{C_{8,ij}}{r_{ij}^8}, \tag{3}$$

where $f_n(r_{ij})$ denotes a damping function of the form

$$f_n(r_{ij}) = 1 - \exp(-b_{ij}r_{ij})\sum_{k=0}^{n}\frac{(b_{ij}r_{ij})^k}{k!} \qquad \text{for } n = 6, 8. \tag{4}$$

At first glance, the reader might have the impression that this functional form entails high computational costs, but this is not the case. In fact, despite its greater complexity, the TT potential is computed equally fast as the Mie potential and scales very well for larger ensembles, cf. Fig. 1.

The TT potential contains five parameters A_{ij}, α_{ij}, b_{ij}, $C_{6,ij}$, $C_{8,ij}$. In case of mixtures or pure fluid molecular models constituted by different TT sites, the parameters of the interactions between unlike sites are determined by the following combination rules [15, 16]

$$\alpha_{ij} = 2\frac{\alpha_i \cdot \alpha_j}{\alpha_i + \alpha_j}, \tag{5}$$

$$A_{ij} = \frac{1}{\alpha_{ij}}\left((A_i\alpha_i)^{1/\alpha_i} \cdot (A_j\alpha_j)^{1/\alpha_j}\right)^{\alpha_{ij}/2}, \tag{6}$$

$$b_{ij} = 2\frac{b_i \cdot b_j}{b_i + b_j}, \tag{7}$$

$$C_{n,ij} = \left(C_{n,i}^{1/b_i} \cdot C_{n,j}^{1/b_j}\right)^{b_{ij}/2} \qquad \text{for } n = 6, 8, \tag{8}$$

$$s_{ij} = \frac{s_i + s_j}{2}. \tag{9}$$

Subscripts i and j denote the parameters for the like interactions, whereas ij indicates the unlike interactions. In the remainder of this section, the indices ij will be omitted for brevity. The parameter s refers to the shielding for the short range correction that is discussed below.

Long-range interactions beyond the specified cutoff radius r_c are not calculated explicitly. Instead, analytical correction terms are used, which result in the necessity to compute

$$\int_{r_c}^{\infty} r^2 u(r)\mathrm{d}r. \tag{10}$$

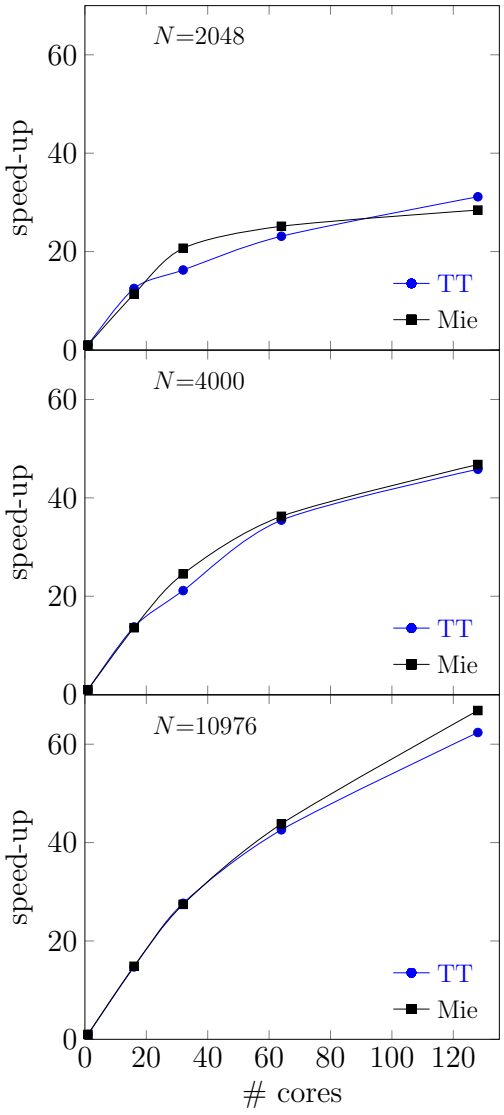

Figure 1: Performance of the TT and Mie potentials for 5000 MD simulation steps with varying particle number N using MPI parallelization.

To this end, the application of angle averaging according to Lustig [14] is desired, which reduces the computational effort to one logical operation per pair of interacting molecules to verify whether $r < r_c$. Hence, the potential function needs to be expressed in terms of r^{2m}. Note that $m < -3/2$ must be satisfied to guarantee convergence of Eq. (10). For simplicity, the repulsive term $A\exp(-\alpha r)$ was neglected. Further, since $f_n(r) \to 1$ as $r \to \infty$, it is reasonable to assume

$$f_n(r)\frac{C_n}{r^n} \approx f_n(r_c)\frac{C_n}{r^n} \qquad \text{for } n = 6, 8. \tag{11}$$

Numerical experiments suggest that this is a suitable approximation as long as r_c exceeds three to four times the distance of the potential well minimum.

Widom's test particle insertion and thermodynamic integration [2, 4] are methods to determine the chemical potential of a given component. However, in combination with the TT potential, both techniques require some caution. For the application of Widom's insertion method, a short range correction has to be applied. Numerical inaccuracies associated with the calculation of the exponential terms on computer hardware via Taylor series expansion cause the damping functions to exhibit oscillations in their entire domain. Their amplitude is negligible, except for small values of r. In case of $r \to 0$ the oscillations may lead to a change of sign of the otherwise strictly positive functions, which in turn produces erratic results for the chemical potential.

This can be prevented by choosing a representation of the dispersive interactions which avoids the exponential term

$$f_n(r)\frac{C_n}{r^n} = -C_n b^n \sum_{k=n+1}^{\infty} \eta_{n,k}(br)^{k-n} \qquad \text{for } n = 6, 8, \tag{12}$$

where the coefficients $\eta_{n,k}$ are given by

$$\eta_{n,k} = \frac{1}{k!}\sum_{m=0}^{n}(-1)^{k+m}\binom{k}{m} \qquad \text{for } n = 6, 8;\ k > n. \tag{13}$$

In *ms*2, series (12) is computed up to $k = 18$. With regard to the decreasing absolute value of the summands and therefore fading contribution of higher order terms, this proved to be an appropriate and sufficiently accurate choice.

To determine which representation of the dispersive terms is used, the additional parameter *shielding* was introduced into *ms*2. It denotes a lower bound of r up to which the usual form of the TT potential (3) is used. Note that the required shielding depends solely on parameter b.

Thermodynamic integration [4] is another technique to compute the chemical potential. In *ms*2, a non-linear scaling was implemented, i.e. $u(\lambda) = \lambda^d u$ for

$\lambda \in [0\,,1]$, with d being an input parameter. The default value is set to $d = 4$ in order to prevent the occurrence of singularities at $\lambda = 0$ or 1 when the LJ potential is used [17].

However, d has to be chosen under consideration of the shape of the repulsive interaction. Clearly, the slope of the repulsive term $\exp(-\alpha r)$ of the TT potential (3) depends on α. A comparison to the slope of r^{-12} as used in the LJ potential, yields

$$\left|\frac{\mathrm{d}\exp(-\alpha r)}{\mathrm{d}r}\right| = \alpha\exp(-\alpha r) \ll 12r^{-13} = \left|\frac{\mathrm{d}r^{-12}}{\mathrm{d}r}\right|, \tag{14}$$

for $\alpha < 5$ and sufficiently small values of r, where each potential function is dominated by its repulsive part. Hence, it is recommended to adjust d accordingly, since the statistical uncertainty of the obtained chemical potential rises as d increases.

4. Thermodynamic factor through Kirkwood-Buff integration

Both the Fick diffusion coefficient $\boldsymbol{D}$ and the MS diffusion coefficient $\boldsymbol{Đ}$ matrices are of central importance when describing mass transport in liquid mixtures. While the former can be measured in the lab due to its composition dependence, the latter cannot be acquired by experiments because it is related to the chemical potential. The thermodynamic factor $\boldsymbol{\Gamma}$ connects these diffusion coefficient matrices and is given for a multi-component mixture by

$$\Gamma_{ij} = \frac{x_i}{k_\mathrm{B}T}\left(\frac{\partial\mu_i}{\partial x_j}\right)_{T,p,\Sigma}, \tag{15}$$

where μ_i is the chemical potential of component i, x_j the mole fraction of component j, T the temperature and k_B the Boltzmann constant. This equation applies at constant temperature and pressure as well as $\sum_{i=1}^{n} x_i = 1$, with number of components n [18].

For mixtures containing three or more components, $\boldsymbol{D} = \boldsymbol{B}^{-1}\boldsymbol{\Gamma}$, where matrix $\boldsymbol{B}$ is determined by the MS diffusion coefficient matrix $\boldsymbol{Đ}$ [18]. Describing mixtures constituted of many components, matrix $\boldsymbol{D}$ asymptotically requires about twice as much information than matrix $\boldsymbol{Đ}$, i.e. $\boldsymbol{\Gamma}$ connects nine Fick with six MS diffusion coefficient elements in case of a quaternary mixture [10]. This is an important advantage of MS theory, but when mass transport needs to be accessed experimentally, the Fick approach can be applied directly. Taking advantage of both, the thermodynamic factor $\boldsymbol{\Gamma}$ is indispensable. Since it is a derivative of the chemical potential, $\boldsymbol{\Gamma}$ cannot be measured experimentally. Instead, excess Gibbs energy models or equations of state are usually employed to extract $\boldsymbol{\Gamma}$ from phase equilibrium data. As an alternative route, molecular simulation combined with KBI [8] allows for the sampling of Γ_{ij} [9, 10, 18].

KBI based on molecules' center of mass radial distribution functions (RDF) was implemented in the NVT ensemble both for MC or MD simulations. In $ms2$, RDF are sampled in the entire cubic simulation volume L^3 up to $\sqrt{3}L/2$, i.e. beyond the cutoff radius that is independently specified for explicitly evaluating the intermolecular interactions. Thus, extended schemes may be applied [19]. In the context of KBI, RDF corrections are required. The correction proposed by Ganguly and van der Vegt [20], referred to as vdV, was implemented into $ms2$ because it was found to be the most adequate [9]. It takes the excess or depletion of molecular species j around a given molecule i at the distance r into account such that the asymptotic behavior of the RDF should yield an improved convergence to unity. The vdV correction is given by Ganguly and van der Vegt [20]

$$g_{ij}^{\mathrm{vdV}}(r) = g_{ij}(r)\frac{N_j(1-V(r)/L^3)}{N_j(1-V(r)/L^3)-\Delta N_{ij}(r)-\delta_{ij}}\,, \tag{16}$$

where $g_{ij}(r)$ is the RDF between components i and j, N_j is the number of molecules j, δ_{ij} the Kronecker delta and $V(r) = 4\pi r^3/3$. Excess or depletion is determined by $\Delta N_{ij}(r) = \int_0^r 4\pi r'^2 \rho_j [g_{ij}(r') - 1]\mathrm{d}r'$ with the partial density ρ_j.

KBI are strictly defined in the μVT ensemble only, which is challenging to impose for dense liquid states. In order to apply KBI to NVT ensemble simulation data, an integral truncation and correction by Krüger *et al.* [21] was implemented, such that KBI are calculated by

$$G_{ij}(R) = \int_0^{2R} 4\pi r^2 (1 - 3x/2 + x^3/2)[g_{ij}(r) - 1]\mathrm{d}r\,, \tag{17}$$

with $x = r/2R$. Its success for finite system sizes was discussed recently [9]. However, the extrapolation to the thermodynamic limit $V \to \infty$, where all ensemble types converge, is essential [9, 19]. Thus, for $V \to \infty$ the following KBI approximation [19] was implemented

$$G_{ij}^{\infty}(R) \approx \int_0^{2R} 4\pi r^2 (1 - 23x^3/8 + 3x^4/4 + 9x^5/8)[g_{ij}(r) - 1]\mathrm{d}r\,. \tag{18}$$

Eqs. (17) and (18) were implemented for both standard and vdV corrected RDF in $ms2$. Fig. 2 exemplarily shows G_{ij} over inverse radius R^{-1} for a binary LJ mixture. An almost linear behavior is produced by Eq. (17) and extrapolations to the thermodynamic limit are well presented by Eq. (18), whereas standard KBI is of little use for extrapolation purposes.

Expressions of the $(n-1)\times(n-1)$ matrix $\mathbf{\Gamma}$ based on KBI G_{ij} can be found in the literature [8, 18, 22] for binary and ternary mixtures. Ben-Naim [22] outlined a general formalism to obtain chemical potential derivatives from KBI. Employing these, $\mathbf{\Gamma}$ expressions for quaternary mixtures were recently derived and evaluated

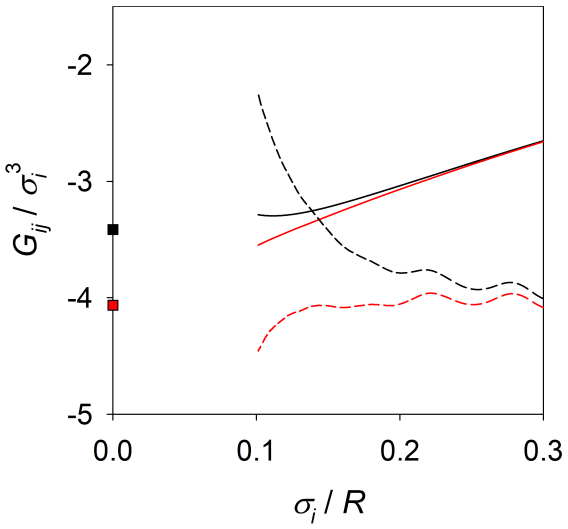

Figure 2: G_{ij} and G_{ij}^{∞} over inverse radius R^{-1} for a liquid binary LJ mixture ($\sigma_j/\sigma_i = 1.5$, $\varepsilon_j/\varepsilon_i = 0.75$). RDF were sampled by MD every time step over a production period of $1.5 \cdot 10^7$ time steps in the NVT ensemble containing $N = 4000$ molecules; dashed lines: standard $G_{ij} = 4\pi \int_0^R [g_{ij}(r) - 1] r^2 \mathrm{d}r$; solid lines: G_{ij} from Eq. (17); squares: G_{ij}^{∞} from Eq. (18); black: standard RDF; red: vdV corrected RDF.

by our group [10]. Fig. 3 shows results for $\mathbf{\Gamma}$ for a quaternary state point. Taking into consideration that there are no other standard simulation methods available for the direct sampling of $\mathbf{\Gamma}$ and due to the satisfactory performance shown in [11], $\mathbf{\Gamma}$ expressions for mixtures with up to four components were implemented into $ms2$ such that cumbersome post-processing is omitted. Invoking KBI leads to an increase of execution time by about 4 % only for realistic molecules because of efficient coding and parallelization [9].

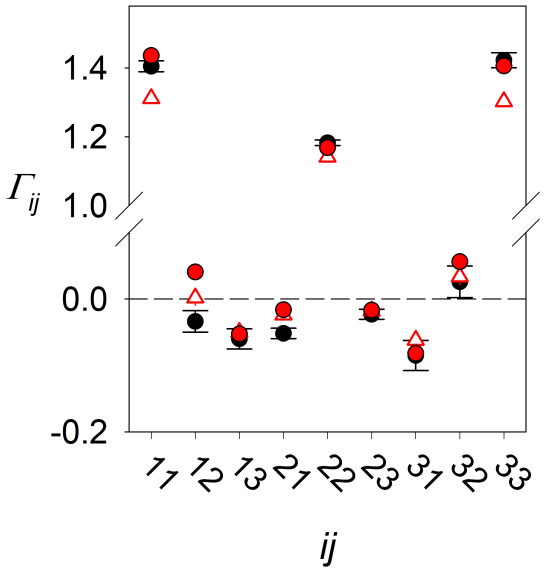

Figure 3: Thermodynamic factor $\mathbf{\Gamma}$ of a liquid-like supercritical quaternary LJ mixture. Black circles: $\mathbf{\Gamma}$ based on numerical chemical potential derivatives sampled with Widom's test particle insertion; red circles/triangles: $\mathbf{\Gamma}$ calculated with the expressions for quaternary mixtures based on vdV corrected RDF and Eq. (18)/Eq. (17) (statistical uncertainties are within symbol size).

5. Orientational distribution function

The sampling of the orientational distribution function (ODF) of dipolar fluids with MD simulation was implemented as a new feature in *ms*2. The ODF quantifies how neighboring molecules are mutually oriented. Such information on the fluid structure is useful for the parametrization of equations of state that give access to the relative permittivity [23, 24].

The ODF O_{ij} for molecules of species j around central molecules of species i can be defined implicitly by the two particle density

$$n_{ij}(r, \varphi_i, \varphi_j, \gamma) = \rho_i \rho_j g_{ij}(r) O_{ij}(r, \varphi_i, \varphi_j, \gamma), \tag{19}$$

which quantifies how many molecules of species i and j at a distance r have a mutual orientation given by the angles φ_i, φ_j and γ. n_{ij} can be separated into the bulk partial densities ρ_i and ρ_j, the RDF g_{ij} and the ODF O_{ij}, which considers mutual orientation. The distance and the three angles characterizing the ODF are depicted in Fig. 4.

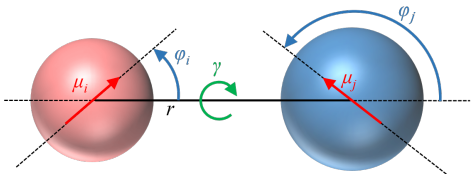

Figure 4: Distance r and angles of the coordinate system for ODF calculation. $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ denote the effective dipole vectors of molecules i and j, which may arise either from point dipoles, a distribution of charges or a combination of the two. φ_i and φ_j are their inclination angles with respect to the intermolecular distance vector $\boldsymbol{r}$. The torsion angle γ describes the angle difference between the molecules' orientation around the intermolecular distance vector.

These three angles fully describe the mutual orientation of two molecules with two rotational degrees of freedom, such as hydrogen chloride or Stockmayer models, which do not rotate around their main axis. The characterization of the mutual orientation of two molecules with three rotational degrees of freedom would necessitate the introduction of two additional angles to describe each molecule's orientation around their main axis, which is not yet supported by *ms*2. The ODF may nonetheless be sampled for molecules with three rotational degrees of freedom, but only the three angles depicted in Fig. 4 will be evaluated.

The ODF is sampled as an average quantity by discretizing the distance and orientation space with a classic binning scheme. The cosines of the three angles can be computed directly from the direction vectors of the dipoles through algebraic operations. When sampling the ODF, $\cos(\varphi_i)$, $\cos(\varphi_j)$, γ and r are discretized into bins of constant size. Sampling φ_i and φ_j in terms of their respective cosines avoids two numerically costly arccosine operations per molecular pair. It also homogenizes the quality of the data across the orientation space, as the volumes around the molecules spanned by the discrete angular elements are equally large when a constant angular increment $\Delta\cos(\varphi_i)$ is chosen instead of constant $\Delta\varphi_i$. The increment size may be specified by the user. The total number of sampling points is the product of the number of grid points in the four relevant dimensions.

Langenbach [23] used 40 grid points for $\cos(\varphi_i)$ and $\cos(\varphi_j)$ and 36 grid points for the torsion angle γ, while sampling the ODF within the first coordination shell without applying further discretization to the intermolecular distance, which resulted in a total of 57,600 sampling points. This led to a satisfactory resolution of the ODF, while keeping memory demand and data output within reasonable limits. The present implementation always samples the ODF within the cutoff radius. It is recommended to specify the element size Δr on the same order of magnitude as the size of the molecular models under investigation. For spherical molecular

models, it thus makes sense to choose Δr to be the molecular radius. In this case, the first radial element corresponds to the volume occupied by the central molecule, which carries little information. Beyond the first radial element, then each two consecutive radial elements cover one spherical shell that roughly corresponds to one of the central molecule's coordination shells. With post-processing, data for two radial elements can be merged to characterize the respective coordination shell. The ODF is normalized by $ms2$ so that the average of all sampling points within the same radial sampling element is unity. The normalization value of each shell is provided in the output.

The present implementation samples mutual orientations on the basis of the total dipole moment vectors of the molecules and works for both point dipoles and dipole moments arising from a distribution of partial point charges. The implementation is compatible with Ewald summation and the reaction field method for treating electrostatic long-range interactions. Any number of components may be chosen as long as at least one dipolar species is present (which may have a vanishing dipole moment). In this case, ODF are recorded for every like and unlike pair of dipolar species. In case of a binary mixture of the dipolar components i and j, this entails four ODF: the two like ones O_{ii} and O_{jj}, which describe mutual orientations between molecules of the same species, O_{ij}, which describes how molecules of species j orient themselves around central molecules of species i and O_{ji} describing the reverse case. The unlike ODF are not identical, but adhere to symmetry conditions. Thus, O_{ji} is not sampled directly by $ms2$. Instead, it is computed upon output from the data for O_{ij}.

The ODF can be visualized in terms of isosurfaces that represent equal probabilities within orientation space. Fig. 5 exemplarily shows the ODF of a pure Stockmayer fluid sampled with MD simulation in the NVT ensemble.

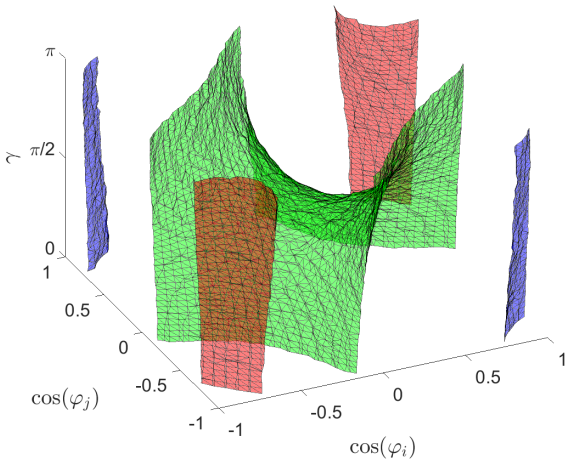

Figure 5: Isosurfaces of the ODF within the first coordination shell $0.5 < r/\sigma < 1.5$ of the Stockmayer fluid with a dipole moment $\mu/(4\pi\varepsilon_0\varepsilon\sigma^3)^{1/2} = 1.5$ at $k_\mathrm{B}T/\varepsilon = 3$ and $\rho\sigma^3 = 1$. Red surfaces represent states with a probability that is 40 % higher than random orientation, green surfaces indicate random orientation and blue surfaces depict a probability that is 40 % lower than random orientation.

6. Thermal diffusion in binary mixtures

The phenomenological coefficients for heat, mass as well as coupled heat and mass transport, as defined by the framework of irreversible thermodynamics [25], can be sampled with equilibrium MD simulation employing the Green-Kubo formalism. Considering the Soret and Dufour effects, the equations for heat flux J_Q and mass flux of component 1 J_1^m in a binary mixture are [25]

$$\mathbf{J}_Q = -L_{QQ}\frac{\nabla T}{T^2} - L_{Q1}\left(\frac{\partial \mu_1}{\partial w_1}\right)_{T,p}\frac{\nabla w_1}{(1-w_1)T}, \tag{20}$$

$$\mathbf{J}_1^m = -L_{1Q}\frac{\nabla T}{T^2} - L_{11}\left(\frac{\partial \mu_1}{\partial w_1}\right)_{T,p}\frac{\nabla w_1}{(1-w_1)T}, \tag{21}$$

where L_{ab} are the so-called phenomenological Onsager coefficients, describing the proportionality between the thermodynamic forces and the fluxes that they induce. μ_1 and w_1 stand for the chemical potential and mass fraction of component 1, respectively. The coupled phenomenological transport coefficients follow Onsager's reciprocity relations (ORR), i.e. $L_{1Q} = L_{Q1}$.

The phenomenological coefficient for mass transport L_{11} is

$$L_{11} = \frac{V}{3k_\mathrm{B}} \int_0^\infty \langle \mathbf{J}_1^m(0) \mathbf{J}_1^m(t) \rangle \mathrm{d}t, \tag{22}$$

with the mass flux $\mathbf{J}_i^m$

$$\mathbf{J}_i^m(t) = \frac{1}{V} \sum_{k=1}^{N_i} m_i (\mathbf{v}_i^k(t) - \langle \mathbf{v} \rangle). \tag{23}$$

Therein, V is the volume, m_i the molecular mass of component i, $\mathbf{v}_i^k(t)$ the center of mass velocity vector of molecule k of component i at some time t and N_i the number of molecules of component i. The brackets $\langle ... \rangle$ denote the NVT ensemble average. If initialized accordingly, $\langle \mathbf{v} \rangle$ deviates from zero during simulation only within machine error so that $\mathbf{J}_1^m = -\mathbf{J}_2^m$.

The phenomenological coefficient for heat transport L_{QQ} is

$$L_{QQ} = \frac{V}{3k_\mathrm{B}} \int_0^\infty \langle \mathbf{J}_Q(0) \mathbf{J}_Q(t) \rangle \mathrm{d}t. \tag{24}$$

However, in equilibrium MD simulation, only the internal energy flux $\mathbf{J}_E$ and not the heat flux $\mathbf{J}_Q$ can be accessed directly, but both quantities are related by Perronace *et al.* [26]

$$\mathbf{J}_Q = \mathbf{J}_E - \left(\frac{h_1}{m_1} - \frac{h_2}{m_2} \right) \mathbf{J}_1^m, \tag{25}$$

where h_i is the partial molar enthalpy of component i and

$$\frac{\mathbf{J}_E}{V} = \frac{1}{2} \sum_{i=1}^{2} \sum_{k=1}^{N_i} m_i^k \left(\mathbf{v}_i^k \right)^2 \cdot \mathbf{v}_i^k - \frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{N_i} \sum_{l \neq k}^{N_j} \left[\mathbf{r}_{ij}^{kl} \colon \frac{\partial u_{ij}^{kl}}{\partial \mathbf{r}_{ij}^{kl}} - \mathbf{I} \cdot u_{ij}^{kl} \right] \cdot \mathbf{v}_i^k. \tag{26}$$

Therein, u_{ij}^{kl} is the intermolecular potential energy, $\mathbf{r}_{ij}^{kl}$ the distance vector between molecules k and l, while the indices i and j denote the molecular species. The second term in the brackets is a dyadic product (denoted by a colon) with the unitary tensor $\mathbf{I}$.

The phenomenological coefficient for internal energy transport L_{EE} is

$$L_{EE} = \frac{V}{3k_\mathrm{B}} \int_0^\infty \langle \mathbf{J}_E(0) \mathbf{J}_E(t) \rangle \mathrm{d}t, \tag{27}$$

and is related to L_{QQ} by [27]

$$L_{QQ} = L_{EE} - 2L_{1E} \left(\frac{h_1}{m_1} - \frac{h_2}{m_2} \right) + L_{11} \left(\frac{h_1}{m_1} - \frac{h_2}{m_2} \right)^2. \tag{28}$$

The phenomenological coefficient for coupled heat and mass transport L_{1Q} is

$$L_{1Q} = \frac{V}{3k_\mathrm{B}} \int_0^\infty \langle \mathbf{J}_1^m(0)\mathbf{J}_Q(t)\rangle \mathrm{d}t, \tag{29}$$

and the phenomenological coefficient for coupled internal energy and mass transport L_{1E} is

$$L_{1E} = \frac{V}{3k_\mathrm{B}} \int_0^\infty \langle \mathbf{J}_1^m(0)\mathbf{J}_E(t)\rangle \mathrm{d}t. \tag{30}$$

Because of the ORR, phenomenological cross-coefficients are symmetric, i.e. $L_{1E} = L_{E1}$ and $L_{1Q} = L_{Q1}$, but statistically independent. Exemplarly, Fig. 6 shows the cross-correlation functions underlying to the phenomenological coefficients for coupled heat and mass transport L_{1Q} and L_{Q1} of the mixture of argon + krypton in its liquid state. As can be seen, both functions oscillate around the same values and can therefore be averaged to improve statistics. The resulting averaged cross-correlation function is then integrated to obtain the coupled phenomenological coefficient, cf. Fig. 6.

If the partial molar enthalpy h_i of all components is known, the according set of values should be specified in the *.par file and $ms2$ will calculate the phenomenological coefficients L_{QQ} and L_{1Q}. If this is not the case, the phenomenological coefficients L_{EE} and L_{1E} will be calculated instead. L_{1Q} and L_{1E} are related by [27]

$$L_{1Q} = L_{1E} - \left(\frac{h_1}{m_1} - \frac{h_2}{m_2}\right) L_{11}. \tag{31}$$

An example of how to obtain partial molar enthalpy values is given in the supplemental material. Finally, the thermal diffusion coefficient D_T can be accessed by comparing the phenomenological Eqs. (20) and (21) with the equations for the heat and mass fluxes according to Fourier and Fick considering the Soret and Dufour effects

$$J_Q = -\lambda \nabla T - \left(\frac{\partial \mu_1}{\partial w_1}\right)_{T,p} \rho w_1 T D_T^D \nabla w_1, \tag{32}$$

$$J_1 = -\rho w_1 w_2 D_T^S \nabla T - \rho D \nabla w_1, \tag{33}$$

$$\tag{34}$$

where ρ is the density, λ the thermal conductivity and D the Fick diffusion coefficient. D_T^S and D_T^D are the thermal diffusion coefficients of Soret- and Dufour-type, respectively [26]

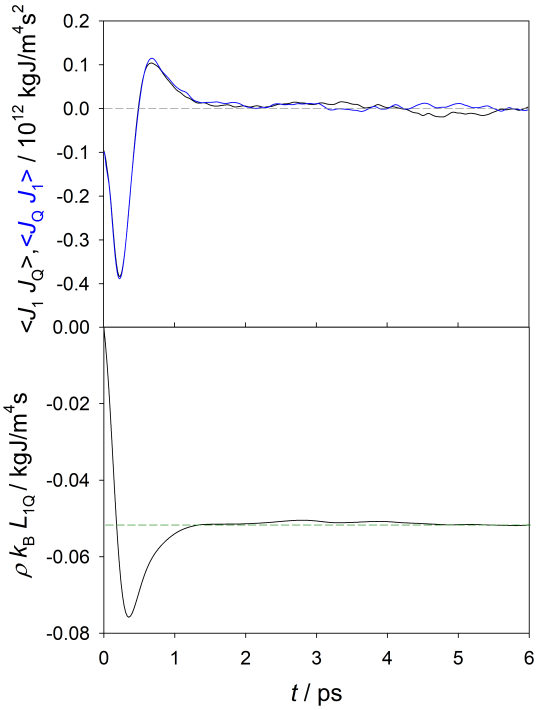

Figure 6: Cross-correlation functions (top) $\langle \mathbf{J}_1(0)\mathbf{J}_Q(t)\rangle$ (black line) and $\langle \mathbf{J}_Q(0)\mathbf{J}_1(t)\rangle$ (blue line) are shown together with the integral of their average (bottom) as a function of time for the liquid mixture argon + krypton at $T = 95.2$ K, $p = 0.1$ MPa and $x_1 = 0.6759$ mol·mol^{-1} sampled with $N = 1000$ molecules.

$$D_T^S = \frac{L_{1Q}}{\rho w_1 w_2 T^2}, \quad D_T^D = \frac{L_{Q1}}{\rho w_1 w_2 T^2}. \tag{35}$$

It thus also follows from the ORR that $D_T^S = D_T^D = D_T$. *ms*2 calculates the thermal diffusion coefficient D_T on the basis of the average of the sampled phenomenological cross-coefficients for coupled heat and mass transport L_{1Q} and L_{Q1}. The thermal diffusion coefficient D_T is strongly dependent on the enthalpic contribution to the heat flow so that it is only calculated if the partial molar enthalpy of all components is specified in the *.par file. Note that Eq. (35) is valid for binary mixtures only.

7. Einstein relations

The Green-Kubo formalism was adopted in previous versions of $ms2$ to sample transport properties, such as diffusion coefficients, viscosity or thermal conductivity. An alternative is offered by the Einstein relations, which can be understood as an integral form for determining these properties. For diffusion coefficients, the Einstein relations deal with molecular displacements, while the Green-Kubo formalism operates with correlation functions of velocities. Both approaches are equivalent, but they show different statistics in practice. For example, long time tails may be encountered with the Green-Kubo formalism, while the Einstein relations do not suffer from this problem.

The expressions for self-diffusion or intra-diffusion D_i and Onsager coefficients Λ_{ij} take the form [28]

$$D_i = \frac{1}{6N_i} \lim_{\Delta t \to \infty} \frac{1}{\Delta t} \left\langle \sum_{k=1}^{N_i} [\boldsymbol{r}_i^k(t+\Delta t) - \boldsymbol{r}_i^k(t)]^2 \right\rangle, \tag{36}$$

$$\Lambda_{ij} = \frac{1}{6N} \lim_{\Delta t \to \infty} \frac{1}{\Delta t} \left\langle \sum_{k=1}^{N_i} [\boldsymbol{r}_i^k(t+\Delta t) - \boldsymbol{r}_i^k(t)] \sum_{l=1}^{N_j} [\boldsymbol{r}_j^l(t+\Delta t) - \boldsymbol{r}_j^l(t)] \right\rangle. \tag{37}$$

Therein, N_i and N_j stand for the number of molecules of components i and j, N is the total number of molecules, $\boldsymbol{r}_i^k$ denotes the Cartesian coordinate vector of molecule k belonging to component i and the brackets $\langle ... \rangle$ indicate ensemble averaging. These relations are analogous to those of the Green-Kubo formalism, except that molecular displacements are considered instead of correlation functions of velocities. Onsager coefficients from both approaches are associated with MS diffusion coefficients in the same way [2, 3, 4].

The Einstein relation for the shear viscosity has the form

$$\eta = \frac{1}{2V k_\mathrm{B} T} \lim_{\Delta t \to \infty} \left\langle [G(t+\Delta t) - G(t)]^2 \right\rangle, \tag{38}$$

where

$$G(t) = \sum_{i=1}^{N} m_i \, r_i^\alpha(t) \, v_i^\beta(t). \tag{39}$$

Eq. (38) cannot be directly applied because $G(t)$ is not continuous and introduces unphysical behavior under periodic boundary conditions [29]. However, this problem can be avoided by substituting the difference in Eq. (38) with the integral [30]

$$\Delta G = G(t+\Delta t) - G(t) = \int_{t}^{t+\Delta t} J_p^{\alpha\beta}(\tau)\mathrm{d}\tau. \tag{40}$$

Consequently, the expression for shear viscosity reads

$$\eta = \frac{1}{2Vk_\mathrm{B}T} \lim_{\Delta t\to\infty} \left\langle \left(\int_{t}^{t+\Delta t} J_p^{\alpha\beta}(\tau)\mathrm{d}\tau \right)^2 \right\rangle, \tag{41}$$

where $J_p^{\alpha\beta}$ is a stress tensor element, which is exactly the same as that in the Green-Kubo formalism (see Eq. (11) in Ref. [2] for details)

$$J_p^{\alpha\beta} = \sum_{l=1}^{N} m_i v_i^\alpha v_i^\beta - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=1}^{n} \sum_{l=1}^{n} r_{ij}^\alpha \frac{\partial u_{ij}}{\partial r_{kl}^\beta}. \tag{42}$$

Therein, $\alpha, \beta = x, y, z$ are Cartesian coordinates, m_i and v_i^α are mass and velocity of molecule i, k and l are the indices of the n interaction sites constituting a molecular model, r_{ij}^α is the site-site distance and u_{ij} the potential energy of the site-site interaction. The present implementation averages over the three off-diagonal elements J_p^{xy}, J_p^{xz}, J_p^{yz} of the stress tensor (42).

Procedures for sampling the transport properties can be employed concurrently with both approaches. Invoking the Einstein formalism leads to little additional computational effort. On average, switching on the Einstein procedure increases the total execution time by less than 2%.

Fig. 7 shows a comparison between data sampled with the Green-Kubo formalism and the Einstein relations. For all considered transport properties, an excellent agreement between both approaches was reached.

Based on Eq. (36) it is straightforward to analyze system dynamics, e.g. solid-fluid phase transitions by the mean-squared displacement (MSD) $\langle \Delta r^2(t) \rangle$ [31]. In this context, the closely related Non-Gaussian parameter is given by [32]

$$\alpha_2(t) = \frac{3\langle \Delta r^4(t) \rangle}{5\langle \Delta r^2(t) \rangle^2} - 1\,. \tag{43}$$

This property can be applied to study dynamic heterogeneity in terms of mobile and immobile particles [31]. Both MSD and α_2 were implemented into $ms2$ and can be sampled independently of the other transport properties.

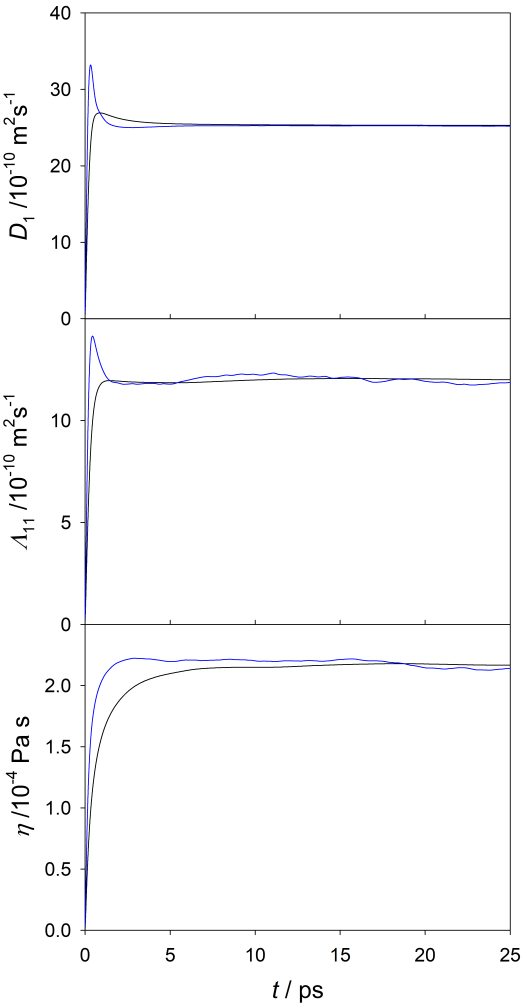

Figure 7: Intra-diffusion coefficient of argon (top), Onsager coefficient Λ_{11} (center) and shear viscosity (bottom) determined with the Green-Kubo formalism (blue) and the Einstein relations (black) for the liquid mixture argon (1) + krypton (2) at $T = 95.25$ K, $p = 0.1$ MPa and $x_1 = 0.6759$ mol·mol^{-1} sampled with $N = 1000$ molecules.

8. Dielectric constant

The sampling of the static dielectric constant ε of non-polarizable fluid models, also known as relative permittivity, was implemented in *ms*2. In the NVT ensemble, it is computed from Kirkwood's fluctuation formula [33]

$$\varepsilon - 1 = \frac{4\pi}{3k_\mathrm{B}TV}\left(\langle \boldsymbol{M}^2\rangle - \langle \boldsymbol{M}\rangle^2\right), \tag{44}$$

where all symbols have their usual meaning and $\boldsymbol{M}$ is the total dipole moment of the simulation volume

$$\boldsymbol{M} = \sum_{i=1}^{N} \boldsymbol{\mu}_i, \tag{45}$$

that is constituted by the sum of the dipole moment vectors $\boldsymbol{\mu}_i$ of all molecules i. In the NpT ensemble, the volume V in Eq. (44) has to be replaced by the ensemble average $\langle V\rangle$. In case of isotropic and non-ferroelectric fluids, the second term $\langle \boldsymbol{M}\rangle^2$ should vanish when sufficiently long sampling is carried out. Nevertheless, that term is preserved in the present implementation to allow for convergence checks.

In *ms*2, Eq. (44) can be sampled both with MC and MD simulations. However, MD simulations are recommended as long individual series of samples are needed for the term $\langle \boldsymbol{M}^2\rangle$ to converge (see below). The present implementation is compatible with both the reaction field method and Ewald summation for treating the long-range electrostatic interactions. For molecular models containing a distribution of partial charges, in case of the reaction field method, their dipole moment vectors $\boldsymbol{\mu}_i$ are readily available. In case of Ewald summation, the summation of partial charges to a molecular dipole moment $\boldsymbol{\mu}_i$ is carried out for each molecule. If a mixture contains ions, which carry a permanent charge, they affect the dielectric constant only through their interactions that alter the orientation of solvent molecules [34].

Fig. 8 shows the running averages of the two terms involved in Eq. (44) from MD simulations of SPC/E water. The reaction field method with conducting boundary conditions was used to treat long-range electrostatics. It can be seen that the second term $\langle \boldsymbol{M}\rangle^2$ indeed quickly vanishes, but a long simulation is needed for the first term $\langle \boldsymbol{M}^2\rangle$ to converge.

In addition to previous validations of the *ms*2 implementation [24, 34, 35] by comparison to literature data, the dielectric constant of two water models and two methanol models at ambient conditions was computed with *ms*2. Again, the reaction field method with conducting boundary conditions was used to treat long-range electrostatics. Good agreement with literature data was found, as Tab. 1 shows.

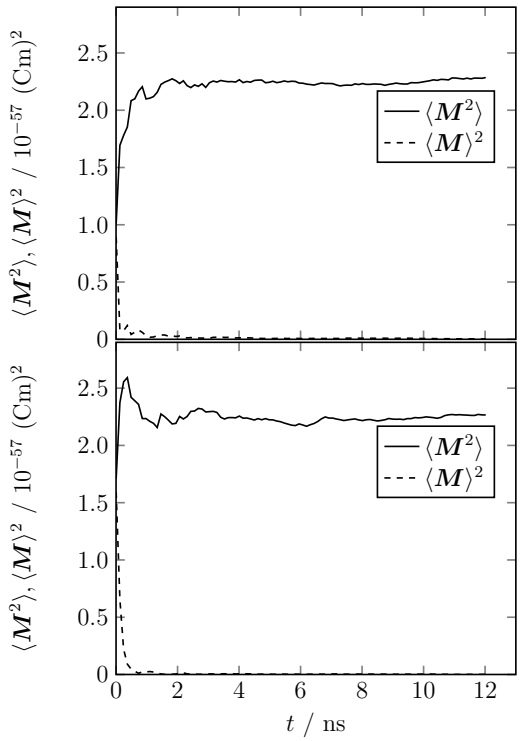

Figure 8: Convergence of running averages of the total dipole moment of SPC/E water at $T = 298.15$ K and $p = 0.1$ MPa. Top: NpT simulation, bottom: NVT simulation.

Table 1: Dielectric constant ε of four pure component models at $T = 298.15$ K and $p = 0.1$ MPa calculated with *ms*2 in comparison with benchmark values from the literature. Uncertainties of the last specified digit of the *ms*2 results are given in parentheses. Uncertainties of the literature values are not available.

Model	*ms*2	Literature	Ref.
Water SPC/E	71(1)	70	[36]
Water TIP4P/2005	57(1)	59	[36]
Methanol OPLS/2016	26.6(7)	26.4	[37]
Methanol Schnabel *et al.*	21.2(6)	21.2	[37]

9. Vapor-liquid equilibria with the *NpT* plus second virial coefficient method

VLE of fluids constituted by any number of components can be calculated by $ms2$ with the Grand equilibrium method [38] through two subsequent simulations of the coexisting phases. However, the vapor simulation can often be substituted with an equation of state, which shortens the process, saves computational effort and avoids the difficulty of sampling low density states. For instance, the Haar-Shenker-Kohler equation has been applied in concert with the NpT plus test particle method to pure fluids [39] and mixtures [40, 41].

An alternative route to VLE at low pressure is the NpT plus second virial coefficient method (NpT + SVC). This approach was used for VLE calculations of phenol, aniline and cyclohexylamine as well as their mixtures before [42] and was fully described elsewhere [39].

One liquid phase simulation run in the NpT ensemble at specified temperature T, composition $\boldsymbol{x}$ and some pressure p_0 gives the chemical potential as a function of pressure p of all components in the liquid as a first-order Taylor expansion around p_0

$$\mu_i^l(T, \boldsymbol{x}, p) = \mu_{i0}^l(T, \boldsymbol{x}, p_0) + (\partial \mu_i / \partial p)_{T,\mathbf{x}} \cdot (p - p_0)\,, \tag{46}$$

where $(\partial \mu_i / \partial p)_{T,\mathbf{x}} = v_i^l$ is the partial molar volume of component i. Sampling of the liquid phase with Widom's test particle method yields values for all coefficients of Eq. (46) [38, 39]. The chemical potential of the vapor can be expressed on the basis of the virial equation of state $p = \rho k_\mathrm{B} T(1 + B\rho)$ by

$$\mu_i^v(T, \mathbf{y}, p) = k_\mathrm{B} T \ln y_i + k_\mathrm{B} T \ln \rho + 2 k_\mathrm{B} T \rho \sum_{j=1}^{n} y_j B_{ij}, \tag{47}$$

with the second virial coefficient $B = \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j B_{ij}$ and the vapor density $\rho = (\sqrt{4Bp/k_\mathrm{B}T + 1} - 1)/(2B)$. The second virial coefficient is evaluated in $ms2$ by numerical integration of Mayer's f-function. The phase equilibrium conditions are then employed to identify the saturated vapor pressure p and the saturated vapor composition $\boldsymbol{y}$ through the nonlinear system of equations

$$\mu_i^l(T, \mathbf{x}, p) = \mu_i^v(T, \mathbf{y}, p) \quad \text{for} \quad i = 1, \ldots, n. \tag{48}$$

A modified Newton method was implemented into the present version of $ms2$ to solve Eq. (48).

To evaluate the performance of the NpT + SVC method, the saturated vapor density and compressibility factor of the pure LJ fluid are compared to the EOS by Thol *et al.* [6] in Fig. 9. The NpT + SVC method excellently reproduces both of these properties up to $k_\mathrm{B}T/\varepsilon = 1.05$, with maximum deviations of 0.6 %. In

fact, an evaluation of 45 different VLE data sets for the LJ fluid showed that the systematic simulation errors of the saturated vapor density and the compressibility factor are ± 1.0 % and ± 1.25 %, respectively [43]. Therefore, the NpT + SVC method should not be used for the LJ fluid outside this systematic error span, i.e. above $k_\mathrm{B}T/\varepsilon = 1.05$, cf. Fig. 9. Moreover, this method can be applied to all VLE state points of the binary mixture N_2 + O_2 between 80 and 120 K, as shown in Fig. 10. However, it fails for higher temperatures and thus higher vapor densities due to the limitations of the virial expansion, while the Grand equilibrium method also operates under such conditions.

The range of applicability and precision of the NpT + SVC method can be estimated before the start of a VLE calculation, if data for the compressibility factor, the SVC and the saturated vapor density are available. The vapor density can only be positive, if $\sqrt{(4Bp/k_\mathrm{B}T + 1)} < 1$ and $(4Bp/k_\mathrm{B}T + 1) \geq 0$, since the SVC is negative up to the Boyle temperature. The combination of these terms with $z = p/(\rho k_\mathrm{B}T)$ yields the limiting compressibility factor $z_\mathrm{lim} = -1/4B\rho$. Thus, Eq. (48) has real solutions only for $z < z_\mathrm{lim}$. The closer the ratio z/z_lim is to unity, the less accurate the NpT + SVC method becomes, cf. Tab. 2. However, at low saturated vapor densities, where $z/z_\mathrm{lim} < 0.7$, the deviations for z and ρ'' remain under 1.6 % for both the LJ fluid and the mixture N_2 + O_2.

The NpT + SVC method can replace simulations in the low density regime and leads to lower statistical uncertainties of the VLE properties compared to methods entirely based on simulations. However, it fails to yield VLE near the critical point because the SVC is insufficient in this region. Consequently, the Grand equilibrium method in its classic form with a vapor simulation run should be used under such conditions.

Table 2: Ratio of the calculated to the limiting compressibility factor for the LJ fluid. Results from the NpT + SVC method and the Grand equilibrium method are presented, where the former does not converge above $k_\mathrm{B}T/\varepsilon = 1.20$ as indicated by an asterisk.

$k_\mathrm{B}T/\varepsilon$	z_GE	z_SVC	$z_\mathrm{lim,GE}$	$z_\mathrm{lim,SVC}$	$\frac{z_\mathrm{GE}}{z_\mathrm{lim,GE}}$	$\frac{z_\mathrm{SVC}}{z_\mathrm{lim,SVC}}$
1.05	0.81	0.80	1.27	1.26	0.63	0.64
1.15	0.71	0.68	0.83	0.77	0.86	0.88
1.25	0.58	*	0.52	*	1.10	*

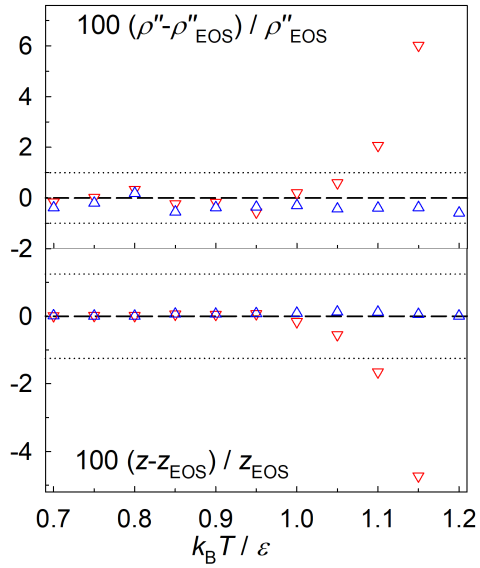

Figure 9: Relative error of saturated vapor density (top) and compressibility factor (bottom) for the LJ fluid determined with the NpT + SVC method (red triangles) and the Grand equilibrium method (blue triangles) in comparison to the EOS by Thol *et al.* [6]. The dotted lines represent the systematic error of the saturated vapor density (± 1%) and the compressibility factor (± 1.25%) evaluated from 45 different VLE data sets for the LJ fluid [43]. Statistical uncertainties are within symbol size.

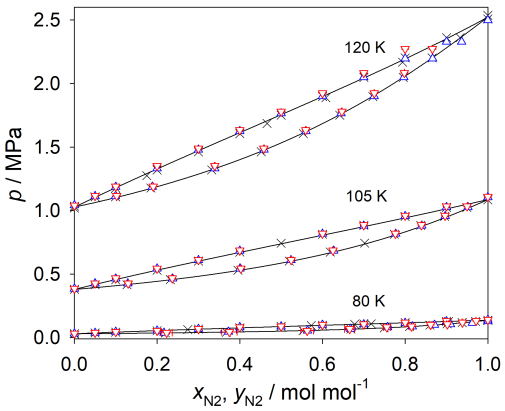

Figure 10: VLE phase diagram of the binary mixture N_2 + O_2 from the NpT + SVC method (red triangles) and the Grand equilibrium method (blue triangles) compared to the Peng-Robinson EOS (black line) and experimental data (cross symbols) [44].

10. Cluster criteria for nucleation

A method in the context of metastable states and homogeneous nucleation [45] was implemented in $ms2$. Its focus lies on the identification of an ongoing vapor-liquid transition, considering both droplet and bubble formation. The primary requirement of such methods is a definition of clusters and voids to identify emerging phases, which typically translates into the evaluation of intermolecular distances, as opposed to methods relying on the chemical potential or multiple metrics.

Instead of comparing distances between molecules and constructing logical structures that are similar to neighbor lists, the present method utilizes an independent grid. Distance checks between molecular positions and grid points of a regular cubic lattice are performed. This route is robust, can be parallelized efficiently and its feasibility has been tested for droplet and bubble nucleation [46].

Phase transitions are associated with spontaneous and significant changes of the local density [47]. However, when the interest lies in the sampling of thermodynamic properties of metastable states, trajectories with an ongoing phase change should be avoided [48]. A criterion has to be set up to consider microstates only that are still consistent with the initial phase and eventually terminates sampling if this is not wanted.

By introducing a regular cubic grid, the instantaneous local density is sampled with $ms2$ by assigning every particle to its surrounding grid points. This operation has a complexity of $O(N)$, where N is the number of particles and the spatial density distribution is evaluated on that grid at user-specified time instances. This evaluation should only be carried out when the molecular configuration had sufficient time to significantly change its structure.

Processing is done directly as a molecule contribution to the according grid points, avoiding expensive iterations over the entire grid. This contribution is calculated by division of the molecular position by the grid constant ΔL. Fig. 11 provides an illustration of the inner workings of the method. For each molecule a primary grid point is evaluated by $\text{index}_x = \text{INT}(r_x/\Delta L)$. All other grid points surrounding that molecule are subsequently assigned with that molecule as well. This simple design works straightforwardly for grids that fill the entire simulation volume.

However, this principle was extended to grids that do not fill the entire simulation volume. This is a mandatory requirement when the grid constant ΔL is an input parameter and $L/\Delta L$ is not an integer. The remaining tripod (in 3D) with a thickness below ΔL was treated in a periodic boundary fashion to preserve same surrounding volume of each grid point.

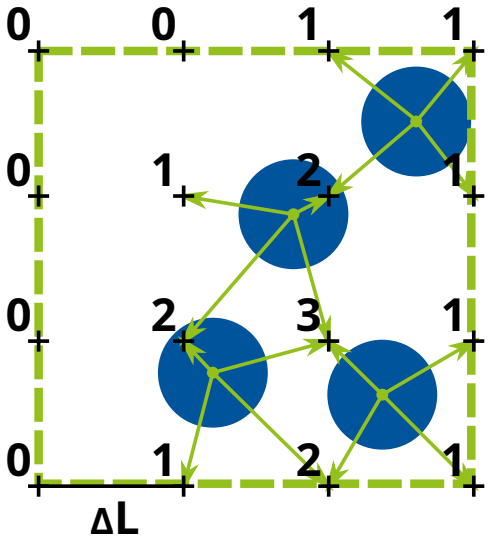

Figure 11: Illustration of the regular grid example (in 2D) with associated values at the grid points marked with +, the simulation volume is delimited by the dashed line, molecule contributions to the grid points are denoted as arrows and grid constant ΔL.

The present approach is rounded up with a decision procedure that evaluates the grid data and terminates sampling, if a specified percentage of grid points signals a local density below or above a specified density threshold. Here, priority was given to robustness across molecular species, investigated phase transition direction, number of molecules etc. The multi-ensemble feature of $ms2$ [4] requires two levels of termination. Once a given ensemble has reached its termination criterion, it is not further sampled. This approach easily extends into simultaneous simulation of multiple ensembles. The entire multi-ensemble simulation is ended only when all ensembles have either reached their termination criterion, the specified number of time steps or wall time.

The desired outcome is to sample the properties of metastable systems before the onset of a phase transition, which can be achieved by properly specifying the parameters of the present approach. The criteria can also be used for other purposes, such as explorations subsequent to nucleation processes and cluster/void precursor evolution. An example for such an observation is presented in Fig. 12 for a bubble nucleation situation, depicting a system before the liquid-to-vapor transition with emerging voids in the system volume. This simplified figure depicts only grid points reporting low molecule neighbor counts, corresponding to emerging voids at the beginning of the phase transition.

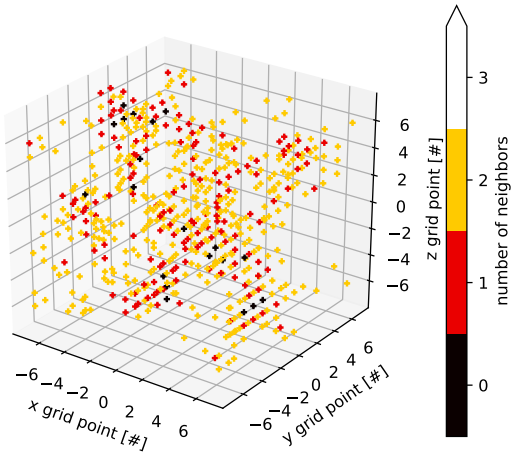

Figure 12: Visualization of grid points that report less than three molecule neighbors in a system with bubble formation. Grid points shown as + are color coded based on the reported number of molecule neighbors. Figure axes correspond to the spatial placement of the grid points within the simulation volume.

Acknowledgments

The authors gratefully acknowledge financial support by German Federal Ministry of Education and Research (BMBF) under the grant 01IH16008 "TaLPas: Task-basierte Lastverteilung und Auto-Tuning in der Partikelsimulation". We gratefully acknowledge the Paderborn Center for Parallel Computing (PC^2) for the generous allocation of computer time on the OCuLUS and Noctua clusters, the High Performance Computing Center Stuttgart (HLRS) under the grant MMHBF2 as well as the Boltzmann-Zuse Society (BZS).

Appendix A. Supplemental material

Supplemental material related to this article can be found online at ...

References

[1] S. Stephan, M.T. Horsch, J. Vrabec, H. Hasse, Mol. Sim. 45 (2019) 806–814.

[2] S. Deublein, B. Eckl, J. Stoll, S.V. Lishchuk, G. Guevara-Carrion, C.W. Glass, T. Merker, M. Bernreuther, H. Hasse, J. Vrabec, Comput. Phys. Commun. 182 (2011) 2350–2367.

[3] C.W. Glass, S. Reiser, G. Rutkai, S. Deublein, A. Köster, G. Guevara-Carrion, A. Wafai, M. Horsch, M. Bernreuther, T. Windmann, H. Hasse, J. Vrabec, Comput. Phys. Commun. 185 (2014) 3302–3306.

[4] G. Rutkai, A. Köster, G. Guevara-Carrion, T. Janzen, M. Schappals, C.W. Glass, M. Bernreuther, A. Wafai, S. Stephan, M. Kohns, S. Reiser, S. Deublein, M. Horsch, H. Hasse, J. Vrabec, Comput. Phys. Commun. 221 (2017) 343–351.

[5] M. Schappals, A. Mecklenfeld, L. Kröger, V. Botan, A. Köster, S. Stephan, E.J. García, G. Rutkai, G. Raabe, P. Klein, K. Leonhard, C.W. Glass, J. Lenhard, J. Vrabec, H. Hasse, J. Chem. Theory Comput. 13 (2017), 4270–4280.

[6] M. Thol, G. Rutkai, A. Köster, R. Lustig, R. Span, J. Vrabec, J. Phys. Chem. Ref. Data 45 (2016) 023101.

[7] K.T. Tang, J.P. Toennies, J. Chem. Phys. 80 (1984) 3726–3741.

[8] J.G. Kirkwood, F.P. Buff, J. Chem. Phys. 19 (1951) 774-778.

[9] R. Fingerhut, J. Vrabec, Fluid Phase Equilib. 485 (2019) 270–281.

[10] R. Fingerhut, G. Herres, J. Vrabec, Mol. Phys. 118 (2020) e1643046.

[11] G. Guevara-Carrion, R. Fingerhut, J. Vrabec, J. Phys. Chem. B 124 (2020) 4527–4535.

[12] G. Mie, Ann. Phys. 11 (1903) 657–697.

[13] T. Lafitte, A. Apostolakou, C. Avendaño, A. Galindo, C.S. Adjiman, E.A. Müller, G. Jackson, J. Chem. Phys. 139 (2013) 154504.

[14] R. Lustig, Mol. Phys. 65 (1988) 175–179.

[15] F.T. Smith, Phys. Rev. A 5 (1972) 1708–1713.

[16] H.-J. Böhm, R. Ahlrichs, J. Chem. Phys. 77 (1982) 2028–2034.

[17] T. Steinbrecher, D.L. Mobley, D.A. Case, J. Chem. Phys. 127 (2007) 214108.

[18] X. Liu, A. Martín-Calvo, E. McGarrity, S.K. Schnell, S. Calero, J.M. Simon, D. Bedeaux, S. Kjelstrup, A. Bardow, T.J.H. Vlugt, Ind. Eng. Chem. Res. 51 (2012) 10247–10258.

[19] P. Krüger, T.J.H. Vlugt, Phys. Rev. E 97 (2018) 051301.

[20] P. Ganguly, N.F.A. van der Vegt, J. Chem. Theor. Comput. 9 (2013) 1347–1355.

[21] P. Krüger, S.K. Schnell, D. Bedeaux, S. Kjelstrup, T.J.H. Vlugt, J.M. Simon, J. Phys. Chem. Lett. 4 (2013) 235–238.

[22] A. Ben-Naim, Molecular Theory of Solutions, Oxford University Press, Oxford, 2006.

[23] K. Langenbach, Chem. Eng. Sci. 174 (2017) 40–55.

[24] K. Langenbach, M. Kohns, J. Chem. Eng. Data 65 (2020) 980–986.

[25] S. R. de Groot, P. Mazur, Non-Equilibrium Thermodynamics, Dover Publications, New York, 1984.

[26] A. Perronace, G. Ciccotti, F. Leroy, A. H. Fuchs, B. Rousseau, Phys. Rev. E 66 (2002) 031201.

[27] J. Armstrong, F. Bresme, Phys. Chem. Chem. Phys. 16 (2014) 12307–12316.

[28] R. Krishna, J.M. van Baten, Ind. Eng. Chem. Res. 44 (2005) 6939–6947.

[29] J.J. Erpenbeck, Phys. Rev. E. 51 (1995) 4296–4308.

[30] K. Meier, A. Laesecke, S. Kabelac, Int. J. Thermophys. 22 (2001) 161–173.

[31] P. Mausbach, R. Fingerhut, J. Vrabec, "Structure and dynamics of the Lennard-Jones fcc-solid focusing on melting precursors", J. Chem. Phys. (2020), submitted.

[32] A. Rahman, Phys. Rev. 136 (1964) A405–A411.

[33] J.G. Kirkwood, J. Chem. Phys. 7 (1939) 911–919.

[34] D. Saric, M. Kohns, J. Vrabec, J. Chem. Phys. 152 (2020) 164502.

[35] M. Kohns, Fluid Phase Equilib. 506 (2020) 112393.

[36] J.L. Aragones, L.G. MacDowell, C. Vega, J. Phys. Chem. A 115 (2011) 5745–5758.

[37] D. Gonzalez-Salgado, C. Vega, J. Chem. Phys. 145 (2016) 034508.

[38] J. Vrabec, H. Hasse, Mol. Phys. 100 (2002) 3375–3383.

[39] A. Lotfi, J. Vrabec, J. Fischer, 76 (1992) 1319–1333.

[40] J. Vrabec, A. Lotfi, J. Fischer, Fluid Phase Equilib. 112 (1995) 173–197.

[41] J. Vrabec, J. Fischer, Mol. Phys. 85 (1995) 781–792.

[42] Y.M. Muñoz-Muñoz, C.-M. Hsieh, J. Vrabec, J. Phys. Chem. B 121 (2017) 5374–5384.

[43] S. Stephan, M. Thol, J. Vrabec, H. Hasse, J. Chem. Inf. Model. 59 (2019) 4248–4265.

[44] B.F. Dodge, Chem. Metall. Eng. 35 (1928) 622.

[45] D. Kashchiev, Nucleation, Elsevier, Oxford, 2000.

[46] K. Langenbach, M. Heilig, M. Horsch, H. Hasse, J. Chem. Phys. 148 (2018) 124702.

[47] S. Toxvaerd, J. Chem. Phys. 143 (2015) 154705.

[48] A. Linhart, C.-C. Chen, J. Vrabec, H. Hasse, J. Chem. Phys. 122 (2005) 144506.

Supplemental Material to:

*ms*2: A molecular simulation tool for thermodynamic properties, release 4.0

Robin Fingerhut[a], Gabriela Guevara-Carrion[a], Isabel Nitzke[a], Denis Saric[a], Joshua Marx[b], Kai Langenbach[b], Sergei Prokopev[c], David Celný[d], Martin Bernreuther[e], Simon Stephan[b], Maximilian Kohns[b], Hans Hasse[b], Jadran Vrabec[a,*]

[a] *Thermodynamics and Process Engineering, Technical University Berlin, 10587 Berlin, Germany*
[b] *Laboratory of Engineering Thermodynamics, University Kaiserslautern, 67653 Kaiserslautern, Germany*
[c] *Computational Fluid Dynamics Laboratory, Institute of Continuous Media Mechanics UB RAS, 614013 Perm, Russia*
[d] *Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 11519 Prague, Czech Republic*
[e] *High Performance Computing Center Stuttgart (HLRS), 70550 Stuttgart, Germany*

This supplemental material contains additional information on the new features of *ms*2 as well as example parameter and potential model files for execution.

*Corresponding author.
E-mail address: vrabec@tu-berlin.de

1. Mie potential

In *ms*2, the potential model type for repulsive and dispersive interactions is selected in the `*.pm` file that specifies the molecular model, cf. Listing 1. To select the Mie potential [1], the user has to set the keyword to `SiteType = MIE` which has to appear below the keyword `NSiteTypes`. For each repulsive/dispersive site of the molecular model the keywords `MIE_n = '#mie parameter n'` and `MIE_m = '#mie parameter m'` have to be set to real (or integer) values `#mie parameter n` and `#mie parameter m` according to the Mie potential parameters n and m. Both keywords have to appear above the coordinates `x, y, z` of each according site, whereas the repulsion parameter n appears above the dispersion parameter m.

Listing 1: Example file (`*.pm`) of a two site Mie potential model.

```
NSiteTypes      = 1

SiteType        = MIE
NSites          = 2

MIE_n           = 12.25
MIE_m           = 6.0
x               = 0.0
y               = 0.0
z               = 0.0
sigma           = 1.0
epsilon         = 1.0
mass            = 1.0

MIE_n           = 11.9
MIE_m           = 6.0
x               = 0.8
y               = 0.0
z               = 0.0
sigma           = 1.0
epsilon         = 1.0
mass            = 1.0

NRotAxes        = auto
```

2. Tang-Toennies potential

In *ms*2, the potential model type for repulsive and dispersive interactions is selected in the `*.pm` file that specifies the molecular model. In case of the Tang-Toennies potential [2], the user has to set the keyword to `SiteType = TT68` which has to appear below the keyword `NSiteTypes`, cf. Listing 2. For each repulsive/dispersive site of the molecular model, the keywords `A = '#tt68 parameter A'`, `b = '#tt68 parameter b'`, `alpha = '#tt68 parameter alpha'`, `C6 = '#tt68 parameter C6'` and `C8 = '#tt68 parameter C8'` have to appear in the given order below the coordinates `x, y, z` of the site. They have to be set to real values `#tt68 parameter A`, `#tt68 parameter b`, `#tt68 parameter alpha`, `#tt68 parameter C6` and `#tt68 parameter C8` according to the Tang-Toennies potential parameters A, b, α, C_6 and C_8. Further, the keyword `shielding = '#tt68 parameter shielding'` has to be set to the real value `#tt68 parameter shielding` to specify the minimal distance up to which the standard representation of the potential is used. The keyword has to appear below the parameter `mass` of each repulsion and dispersion site.

For the calculation of the chemical potential via thermodynamic integration [3], the user has to set `ChemPotMethod = ThermoInt` in the ensemble section in the `*.par` file. Next, the keyword `LambdaExponent = '#TI exponent of lambda'` has to be set to the real (or integer) value `#TI exponent of lambda`. This value specifies the parameter d for non-linear scaling, i.e. $u(\lambda) = \lambda^d u$ for $\lambda \in [0, 1]$.

Listing 2: Example file (`*.pm`) of a single site Tang-Toennies potential model.

```
NSiteTypes = 1

SiteType   = TT68
NSites     = 1

x          = 0.0
y          = 0.0
z          = 0.0
A          = 4466710
b          = 4.18
alpha      = 2.48
C6         = 756300
C8         = 10000000

mass       = 39.948
shielding  = 0.02

NRotAxes   = auto
```

3. Thermodynamic factor through Kirkwood-Buff integration

Kirkwood-Buff integration (KBI) [4] as well as the calculation of the thermodynamic factor matrix $\mathbf{\Gamma}$ [4, 5, 6, 7, 8] can be invoked by the keyword `KBIFreq = '#KBI frequency'` that has to appear under the simulation section in the `*.par` file, cf. Listing 3. The user has to set `#KBI frequency` to an integer value. This value specifies the sampling frequency for center of mass radial distribution functions (RDF) g_{ij} which are the essential input for KBI. The computational effort for KBI is minor [9], sampling each step is thus recommended for highly accurate RDF. Next, the keyword `KBIResetFreq = '#KBI reset frequency'` has to be set to the integer value `#KBI reset frequency` according to a block length in simulation steps. In these blocks, RDF are independently sampled and KBI G_{ij} are determined with the approach given by Krüger *et al.* [10]. In this procedure, KBI G_{ij} are accumulated block-wise over simulation and their statistical uncertainties are determined. For applying KBI G_{ij}^{∞} in the thermodynamic limit $V \to \infty$ according to Krüger *et al.* [11], averaged RDF are required and calculated block-wise. The keyword `KBINumShells = '#RDF number shells'` has to be set to the integer value `#RDF number shells` according to the number of shells for the RDF. Invoking KBI, RDF are sampled in the entire cubic simulation volume up to $\sqrt{3}L/2$, i.e. beyond the cutoff radius that is independently specified for explicitly evaluating the intermolecular interactions. Half of the edge length of the simulation volume $L/2$ is divided into the chosen number of RDF shells so that the number of shells is extended automatically by a factor of $\sqrt{3}$.

The `*.kbirdf` output file contains the block-wise averaged RDF over the entire cubic simulation volume. Therein, standard RDF as well as corrected RDF according to Ganguly and van der Vegt [12] are written with the `#KBI reset frequency`. The `*.kbirav` output file contains the block-wise running averages of KBI G_{ij}, G_{ij}^{∞} and their statistical uncertainties for standard KBI as well as the expressions developed by Krüger *et al.* [10, 11]. Moreover, each KBI is given for standard RDF and corrected RDF [12]. The simulation result file `*.res` contains the thermodynamic factor matrix $\mathbf{\Gamma}$ [4, 5, 6, 7, 8] for each RDF type and KBI type with statistical uncertainties according to the error propagation law.

Listing 3: Example simulation parameter file (`*.par`) of a binary Lennard-Jones mixture for executing KBI.

```
Units          =        Reduced
LengthUnit     =        1.0
EnergyUnit     =        1.0
MassUnit       =        1.0
Simulation     =        MD
Integrator     =        Gear
TimeStep       =        3.0E-4
Ensemble       =        NVT
MCORSteps      =        0
NVTSteps       =        800000
RunSteps       =        15000000
ResultFreq     =        1000
ErrorsFreq     =        5000
VisualFreq     =        0
KBIFreq        =        1
KBIResetFreq   =        10000
KBINumShells   =        500
CutoffMode     =        COM
NEnsembles     =        1

Temperature    = 0.85
Pressure       = 0.03
Density        = 0.199734749
NParticles     = 4000
NComponents    = 2

PotModel       = LJA.pm
MoleFract      = 0.05
ChemPotMethod= none

PotModel       = LJB.pm
MoleFract      = 0.95
ChemPotMethod= none

eta            = 1.0
xi             = 1.0

Cutoff         = 5.0
Epsilon        = 1.0E10
```

4. Orientational distribution function

Sampling of the orientational distribution function (ODF) [13] is enabled by entering the keyword `ODFRecordingFreq = '#ODF recording frequency'` in the ensemble section of the `*.par` file, cf. Listing 4. By choosing `#ODF recording frequency` as an integer value greater than zero, the user specifies the frequency with which the ODF is sampled. Due to the high dimensionality of the ODF, it is recommended to sample it every time step to achieve an adequate data quality within a reasonable time frame. The frequency of output generation is specified with the keyword `ODFOutputFreq = '#ODF output frequency'`. A single output file is written for the ODF, which is updated with a frequency specified by the value of `#ODF output frequency`. The output file does not list the data for all sampling blocks individually and the ODF is not reset. Instead, whenever output is generated, the existing output file is overwritten with the data covering the entire production run up until the last completed time step. The ODF is sampled with a classic binning scheme. By specifying the value `NShellsODF = '#number of shells for ODF sampling'` the user chooses into how many segments the sampling radius of the ODF is divided. The sampling radius always equals the cutoff radius. The number of segments for the cosines of the angles φ_i and φ_j is specified with the keywords `nPhiODF = '#number of segments of` $\cos(\varphi_i)$ `and` $\cos(\varphi_j)$ `for ODF sampling'` and the number of segments for the angle γ is specified with `nGammaODF = '#number of segments of` γ `for ODF sampling'`.
The output file first lists the normalization values for each radial shell of each component pair. Subsequently, the ODF values for each component pair and each bin are listed in a table. Each bin is referred to by its coordinate values at the center of the segment. I.e., if a cutoff radius of 4.5 σ is chosen and divided into three radial segments, the length of each segment is 1.5 σ. The values for the intermolecular radius r listed in the output file are then 0.75 σ, 2.25 σ and 3.75 σ, referring to the value of r in the center of the first, second and third radial segment, respectively.

Listing 4: Example simulation parameter file (`*.par`) of the pure fluid R32 for sampling ODF.

```
Units              =        SI
LengthUnit         =        3.5
EnergyUnit         =        100.0
MassUnit           =        40.0
Simulation         =        MD
TimeStep           =        5E-4
Ensemble           =        NVT
NVTSteps           =        50000
RunSteps           =        1000000
ResultFreq         =        1000
ErrorsFreq         =        5000
VisualFreq         =        0
CutoffMode         =        COM
NEnsembles         =        1

Temperature        = 300
Density            = 20
PistonMass         = 1.0E-4
NParticles         = 1000
NComponents        = 1

PotModel           = R32.pm
MolarFract         = 1.0

ODFRecordingFreq= 1
ODFOutputFreq      = 1000000
NShellsODF         = 3
nPhiODF            = 40
nGammaODF          = 36

Cutoff             = 4.5
Epsilon            = 1.0E10
```

5. Thermal diffusion in binary mixtures

In *ms*2, the thermal diffusion coefficient is calculated with the Green-Kubo formalism, which requires the same input parameters in the `*.par` file from the user as required for other transport properties in the previous implementation, e.g. diffusion coefficients, shear viscosity, thermal and electrical conductivity. Therefore, the keyword `CorrfunMode= YES` has to appear under the simulation section in the `*.par` file and the ensemble-specific keywords `StepsCorrfun, Corrlength, SpanCorrfun, ViewCorrfun` and `ResultFreqCF` as implemented in previous versions of *ms*2. Additionally, the user has to specify the partial molar enthalpy for each component in reduced units employing the keyword `PartMolEnt= '#partial molar enthalpy'` in the `*.par` file, cf. Listing 5. Note that if the keyword `PartMolEnt` is not given in the `*.par` file or its value is set to zero, *ms*2 will not calculate the thermal diffusion coefficient. However, the values of the phenomenological cross-coefficients L_{1E} and L_{E1} as well as the thermal conductivity will be calculated, neglecting the effect of the partial molar enthalpy, and the resulting values will be written into the `*.res` file with the corresponding remark for the users.

The `*.rtr` output file, containing the averaged correlation functions and their integrals, was extended with the corresponding values for the phenomenological cross-coefficients. It should be noted that, unlike other transport properties, these correlation functions are not normalized with their initial value, therefore they do not start at unity.

In the following, an example of how to obtain the partial molar enthalpy is given. In the case of binary mixtures, it is determined in two steps. First, the residual molar enthalpy of the mixture h^{res} is calculated in the isobaric-isothermal (NpT) ensemble over a wide composition range around the required state point. Thereupon, the total enthalpy h is calculated by adding the ideal part h^{id} to the residual enthalpy h^{res}. An appropriate function $h = f(x_i)$ is fitted by a least square optimization to the resulting data for the composition dependence of the total molar enthalpy. The partial molar enthalpy is then calculated by

$$h_i = h + x_j \left(\frac{\partial h}{\partial x_i} \right).$$

In case of that h^{id} is the same for both mixture components, the partial molar enthalpy can directly be determined from the values of the residual enthalpy. For mixtures consisting of more than two components, more simulations are required to obtain the appropriate function for the total enthalpy.

Listing 5: Example simulation parameter file (`*.par`) of the binary mixture argon + krypton for sampling the thermal diffusion coefficient.

```
Units          =        SI
LengthUnit     =        3.405
EnergyUnit     =        119.8
MassUnit       =        39.944
Simulation     =        MD
Integrator     =        Gear
TimeStep       =        6.95658E-4
Ensemble       =        NVT
MCORSteps      =        100
NVTSteps       =        6000000
NPTSteps       =        100000
RunSteps       =        12000000
ResultFreq     =        1000
ErrorsFreq     =        5000
VisualFreq     =        0
CutoffMode     =        COM
NEnsembles     =        1
CorrfunMode    =        yes

Temperature    = 95.2
Pressure       = 0.1
Density        = 31.069649165
NParticles     = 1000
NComponents    = 2

StepsCorrfun = 2
Corrlength   = 4000
SpanCorrfun  = 200
ViewCorrfun  = 10
ResultFreqCF = 1

PotModel       = Ar.pm
MoleFract      = 0.6759
PartMolEnt     = -3.823787458
ChemPotMethod= none

PotModel       = Kr.pm
MoleFract      = 0.3241
PartMolEnt     = -6.979858886
ChemPotMethod= none

eta            = 1.0
xi             = 1.0

Cutoff         = 4.0
Epsilon        = 1.0E10
```

6. Einstein relations

The procedure for sampling transport properties with the Einstein relations, as well as with the Green-Kubo formalism, requires the compiler flag `TRANS=1`. To select the employed calculation procedure, the keyword `TransMethod = '#Einstein/GKEinstein/GK'` was introduced with following options:
- `Einstein`: Einstein procedure is switched on (Green-Kubo off),
- `GKEinstein`: Both, Green-Kubo and Einstein procedure are switched on,
- `GK`: Green-Kubo procedure is switched on (Einstein off), which is the default setting,

cf. `*.par` file (Listing 6). No other options are needed because the Einstein procedure uses the same parameters as the Green-Kubo formalism, i.e. `Corrlength, SpanCorrfun and StepsCorrfun.`

The meaning of these parameters is outlined in the following. Simultaneously, averaging over several correlation function samples is performed, cf. Fig. S1. The first correlation function is sampled when the equilibration process has terminated. After `SpanCorrfun` time steps, a new correlation function is started. Each correlation function has the length `Corrlength` (in time steps) and after the end of the averaging length, a new correlation function is sampled. Averaging is made over all samples of correlation functions. The parameter `StepsCorrfun` stands for the frequency, which determines how often correlation functions are called. As an example, these parameters could be set as `Corrlength = 30000, SpanCorrfun = 1000 and StepsCorrfun = 2`. It is recommended to set them as multiples of each other.

The result of averaging transport properties over time is written to the file with the extension `*.ecoef` (in analogy to the `*.rtr` files of the Green-Kubo formalism). The final results are written to the `*.res` file.

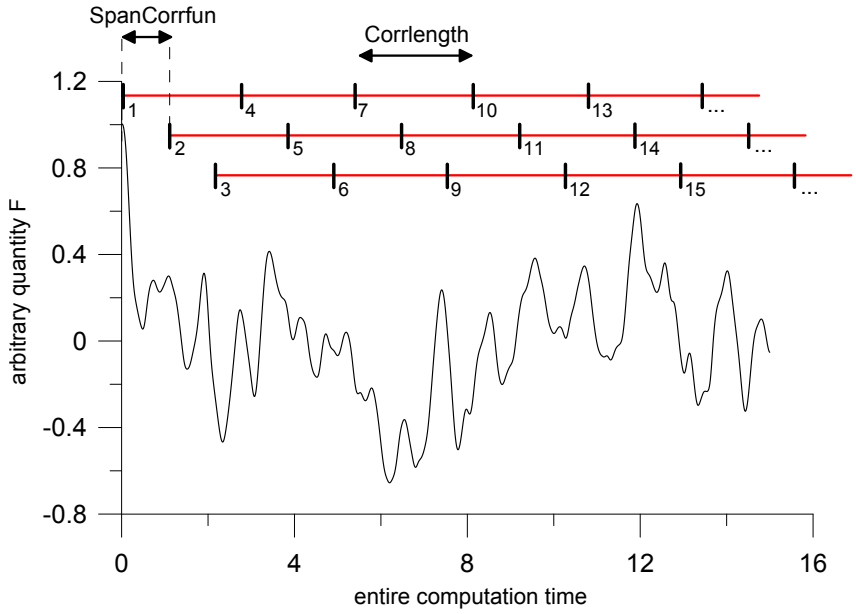

Figure S1: An arbitrary quantity F as a function of time.

Listing 6: Example simulation parameter file (`*.par`) of the binary mixture argon + krypton for applying the Einstein relations.

```
Units           =       SI
LengthUnit      =       3.405
EnergyUnit      =       119.8
MassUnit        =       39.944
Simulation      =       MD
Integrator      =       Gear
TimeStep        =       6.95658E-4
Ensemble        =       NVT
MCORSteps       =       200
NVTSteps        =       1000000
NPTSteps        =       100000
RunSteps        =       20000000
ResultFreq      =       1000
ErrorsFreq      =       5000
VisualFreq      =       0
CutoffMode      =       COM
NEnsembles      =       1
CorrfunMode     =       yes
TransMethod     =       Einstein

Temperature     = 95.2
Pressure        = 0.1
Density         = 31.069649165
NParticles      = 1000
NComponents     = 2

StepsCorrfun    = 1
Corrlength      = 20000
SpanCorrfun     = 200
ViewCorrfun     = 100
ResultFreqCF    = 1
```

```
PotModel        = Ar.pm
MoleFract       = 0.6759
PartMolEnt      = -3.823787458
ChemPotMethod   = none

PotModel        = Kr.pm
MoleFract       = 0.3241
PartMolEnt      = -6.979858886
ChemPotMethod   = none

eta             = 1.0
xi              = 1.0

Cutoff          = 4.0
Epsilon         = 1.0E10
```

The mean-squared displacement (MSD) $\langle \Delta r^2(t) \rangle$ and the Non-Gaussion parameter $\alpha_2(t)$ can be invoked in MD simulations with the keyword `ALPHA2Freq = '#Alpha2 frequency'` that has to appear under the simulation section in the `*.par` file, cf. Listing 7. The user has to set `#Alpha2 frequency` to an integer value which specifies the sampling frequency of both time correlation functions. Next, the keyword `ALPHA2Length = '#Alpha2 length'` has to be set to the integer value `#Alpha2 length` according to the chosen time correlation function length in simulation steps. Moreover, the keyword `ALPHA2Span = '#Alpha2 span'` has to be set to the integer value `#Alpha2 span` in simulation steps. Time correlation functions are subsequently started when `#Alpha2 span` is reached.

The `*.a2rav` output file contains ensemble averaged time correlation functions $\langle \Delta r^2(t) \rangle$ and $\alpha_2(t)$ over the chosen time correlation function length `#Alpha2 length`. Time is given in reduced units and SI units (in fs) according to the chosen time step Δt in the `*.par` and the sampling frequency `#Alpha2 frequency`. Moreover, the number of averaged functions is given in this output file and is written with the frequency of the result file `*.res`.

Listing 7: Example simulation parameter file (`*.par`) of a pure LJ fcc solid for sampling time correlation functions $\langle \Delta r^2(t) \rangle$ and $\alpha_2(t)$.

```
Units          =        Reduced
LengthUnit     =        1.0
EnergyUnit     =        1.0
MassUnit       =        1.0
Simulation     =        MD
Integrator     =        Gear
TimeStep       =        0.001
Ensemble       =        NVT
MCORSteps      =        0
NVTSteps       =        200000
RunSteps       =        1000000
ResultFreq     =        1000
ErrorsFreq     =        5000
VisualFreq     =        0
ALPHA2Freq     =        1
ALPHA2Length   =        10000
ALPHA2Span     =        100
CutoffMode     =        COM
NEnsembles     =        1

Temperature    = 1.0
Density        = 1.8
NParticles     = 10976
NComponents    = 1

PotModel       = LJ126.pm
MolarFract     = 1.0
ChemPotMethod= none
NTest          = 2000

Cutoff         = 9.1
Epsilon        = 1.0E10
```

7. Vapor-liquid equilibria with the NpT plus second virial coefficient method

The more rapid calculation of vapor-liquid equilibria taking advantage of the second virial coefficient (NpT + SVC method) is specified in the `*.par` file under the ensemble section, cf. Listing 8. Specifically, the user has to define `Ensemble = NPTSVC`. Note that this method only works if the chemical potential calculation is turned on in the `*.par` file, i.e. `ChemPotMethod` $\neq$ `None`.

The simulation result file `*.res` contains results of the liquid simulation run in the NpT ensemble, the second virial coefficient and all vapor-liquid equilibrium properties. Furthermore, the compressibility factor ratio for the assessment of the accuracy of the NpT + SVC method is also written to the `*.res` file.

Listing 8: Example simulation parameter file (`*.par`) of the binary mixture nitrogen + oxygen with the NpT + SVC method.

```
Units          =        SI
LengthUnit     =        3.0
EnergyUnit     =        100.0
MassUnit       =        50.0
Simulation     =        MC
Acceptance     =        0.5
Ensemble       =        NPTSVC
NVTSteps       =        20000
NPTSteps       =        20000
RunSteps       =        500000
ResultFreq     =        1000
ErrorsFreq     =        5000
VisualFreq     =        0
CutoffMode     =        COM
NEnsembles     =        1

Temperature    = 80.00
Pressure       = 0.033
Density        = 35.00
NParticles     = 864
NComponents    = 2

PotModel       = N2.pm
MolarFract     = 0.050
ChemPotMethod= Widom
NTest          = 3456

PotModel       = O2.pm
MolarFract     = 0.950
ChemPotMethod= Widom
NTest          = 3456

eta            = 1.0
xi             = 1.007

Cutoff         = 4.0
Epsilon        = 1.0E10
```

8. Cluster criteria for nucleation

The identification of clusters/voids is enabled with the option `ClusterIsCriteria=yes` in the simulation section of the `*.par` file, cf. Listing 9. Although it can be applied to MC and MD simulations, it is recommended to use MD only to sample the thermodynamic properties of metastable states. The identification should be carried out in sufficiently spaced time intervals that can be specified for each ensemble individually with `ClusterCriteriaFreq`. Reasonable values are multiples of 100 time steps. The identification operates in two directions, i.e. in vapor→liquid or in liquid→vapor, which are selected with the option `ClusterCriteriaType='#gridvap / gridliq'`. This determines how `ClusterMoleculeCount` is utilized. In the vapor→liquid case, where clusters may emerge, the grid points are checked for **greater than or equal to** `ClusterMoleculeCount`. Alternatively, in the liquid→vapor case, grid points are checked for voids with **less than or equal to** `ClusterMoleculeCount`.

The parameter `ClusterCriteriaDistance` specifies the grid constant ΔL (in σ_{ref}) and thus the volume attributed to each grid point $V_{\mathrm{gridpoint}} = (2\Delta L)^3$. Of course, ΔL implicitly places an upper limit on the maximum count of molecules that may be assigned to a grid point. It is not required that the edge length of the simulation volume has to be an integer multiple of the grid constant ΔL as the algorithm can deal with this case.

To account for small clusters/voids that can temporarily emerge even in stable systems, the parameter `ClusterMaximumAllowed` was included. It is a threshold percentage of grid points of the entire grid and determines how many grid points have to report a density fluctuation for the ensemble to be terminated. Experience shows that 5 to 10% is a good choice for smaller systems with < 2000 grid points.

The identification procedure generates two file types. A file with the extension `*.grid` contains the positions of the grid points and additional grid properties in human readable format. A second file with the extension `*.clust` contains neighbor counts for all grid points that were sampled, assigning one time instance to a line. Depending on the grid size, this file can become large. Files are created for all ensembles individually and are updated with the frequency `ClusterCriteriaFreq`.

Fig. S2 explains the parameters `ClusterMoleculeCount` and `ClusterMaximumAllowed`. A multi-ensemble simulation was carried out for many densities along one isotherm with non-terminating criteria settings, while returning grid point neighbor data. Fig. S2 shows the results after 10^6 time steps.

A percentage of grid points is specified with `ClusterMaximumAllowed` that terminates the sampling of the ensemble, when the corresponding number of grid points signals that the threshold `ClusterMoleculeCount` was reached. If the parameter pair `ClusterMaximumAllowed=2.0%` and `ClusterMoleculeCount=4` would have been specified for the vapor states depicted in Fig. S2, only the stable and first two metastable ensembles

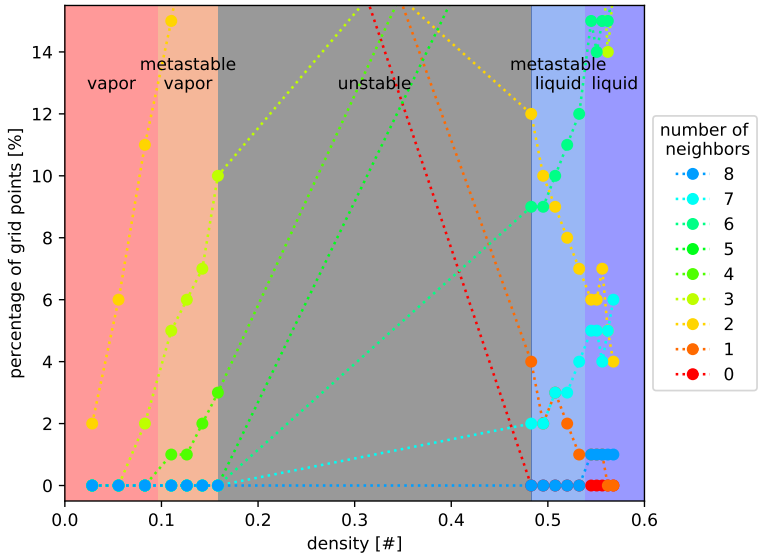

Figure S2: Percentage of grid points reporting a specified number of neighboring molecules for multiple densities along one isotherm, sampled over 10^6 time steps without termination. For orientation, the corresponding regions of the phase diagram are color-coded.

would have been sampled over 10^6 time steps, while the other metastable vapor ensembles would have been terminated earlier. Similar considerations apply to the liquid side. With the parameter pair `ClusterMaximumAllowed=2.0%` and `ClusterMoleculeCount=1`, the two most supersaturated ensembles in Fig. S2 would have been terminated before reaching 10^6 time steps.

The present approach was designed to be simple and provide insight into the metastable region, while ensuring that phase identity of the system is preserved. The current implementation considers only pure component systems, but can be extended to mixtures.

Listing 9: Example simulation parameter file (`*.par`) of the single ensemble simulation of the Lennard-Jones fluid with cluster criteria enabled. The simulation in this configuration is stopped after 100 time steps.

```
Units                  = Reduced
LengthUnit             = 1.0
EnergyUnit             = 1.0
MassUnit               = 1.0
Simulation             = MD
Integrator             = Gear
TimeStep               = 0.0025
Ensemble               = NVT
MCORSteps              = 10
NVTSteps               = 100
RunSteps               = 1000
ResultFreq             = 100
ErrorsFreq             = 100
CutoffMode             = COM
NEnsembles             = 1
ClusterIsCriteria      = Yes

Temperature            = 1.219677
Density                = 0.142332
NParticles             = 1372
NComponents            = 1
ClusterCriteriaFreq    = 100
ClusterCriteriaType    = gridvap
ClusterCriteriaDistance= 1.0
ClusterMoleculeCount   = 5
ClusterMaximumAllowed  = 0.0
ClusterIsCvim          = No

PotModel               = LJ126.pm
MolarFract             = 1.0
ChemPotMethod          = none
NTest                  = 10

Cutoff                 = 5.0
Epsilon                = 1.0E10
```

9. Minor changes with version release 4.0

Pressure calculation with Monte Carlo simulations

MC simulations require potential energy calculations for the translation and rotation acceptance criteria. Thus, force calculations are not necessary in contrast to MD. However, these intense force computations must be invoked when pressure is sampled with MC. Therefore, the keyword `OptPressure = '#yes/no'` under the ensemble section in the `*.par` file was set to logicals `#yes` or `#no` when pressure sampling was turned on or off. Due to structural changes and optimizations in the MC code, the force computations are done most efficiently now. Thus, such as with MD, pressure is throughout sampled with MC up from this version release and the keyword `OptPressure = '#yes/no'` was removed.

Helmholtz energy A_{00} calculation with NVT and NVE ensemble simulations

Residual Helmholtz energy derivatives A^r_{mn} are determined on the fly with the Lustig formalism [14] when applying NVT or NVE ensemble simulations in $ms2$ [15]. Additionally, for these ensemble types the residual Helmholtz energy A^r_{00} is determined by

$$A^r_{00} = -A^r_{01} + \sum_{i=1}^{n} x_i \mu^r_i \,, \tag{1}$$

if the chemical potential is set to `ChemPotMethod` $\neq$ `None` in the `*.par` file. It was implemented for pure fluids and mixtures. Results for A^r_{00} and its statistical uncertainty are given in the `*.res` file.

References

[1] G. Mie, Ann. Phys. 11 (1903) 657–697.

[2] K.T. Tang, J.P. Toennies, J. Chem. Phys. 80 (1984) 3726–3741.

[3] D. Frenkel, B. Smit, Understanding molecular simulation: From algorithms to applications, Academic Press, San Diego, 2002.

[4] J.G. Kirkwood, F.P. Buff, J. Chem. Phys. 19 (1951) 774–778.

[5] A. Ben-Naim, Molecular Theory of Solutions, Oxford University Press, Oxford, 2006.

[6] E. Ruckenstein, I. Shulgin, Fluid Phase Equilib. 180 (2001) 345–359.

[7] X. Liu, A. Martín-Calvo, E. McGarrity, S.K. Schnell, S. Calero, J.M. Simon, D. Bedeaux, S. Kjelstrup, A. Bardow, T.J.H. Vlugt, Ind. Eng. Chem. Res. 51 (2012) 10247–10258.

[8] R. Fingerhut, G. Herres, J. Vrabec, Mol. Phys. 118 (2020) e1643046.

[9] R. Fingerhut, J. Vrabec, Fluid Phase Equilib. 485 (2019) 270–281.

[10] P. Krüger, S.K. Schnell, D. Bedeaux, S. Kjelstrup, T.J.H. Vlugt, J.M. Simon, J. Phys. Chem. Lett. 4 (2013) 235–238.

[11] P. Krüger, T.J.H. Vlugt, Phys. Rev. E 97 (2018) 051301.

[12] P. Ganguly, N.F.A. van der Vegt, J. Chem. Theor. Comput. 9 (2013) 1347–1355.

[13] K. Langenbach, Chem. Eng. Sci. 174 (2017) 40–55.

[14] R. Lustig, Mol. Phys. 110 (2012) 3041–3052.

[15] C.W. Glass, S. Reiser, G. Rutkai, S. Deublein, A. Köster, G. Guevara-Carrion, A. Wafai, M. Horsch, M. Bernreuther, T. Windmann, H. Hasse, J. Vrabec, Comput. Phys. Commun. 185 (2014) 3302–3306.