NTNU
Norwegian University of
Science and Technology

Evaluation of Different Methods for Computation of Kirkwood-Buff Integrals in the Thermodynamic Limit from Molecular Dynamics Simulations

Vilde Bråten

Chemical Engineering and Biotechnology
Submission date: June 2018
Supervisor: Bjørn Hafskjold, IKJ

Norwegian University of Science and Technology
Department of Chemistry

Acknowledgements

I would like to thank my supervisors Bjørn Hafskjold and Signe Kjelstrup for accepting me as a part of the Thermodynamics research group and for their help, insight and motivation throughout the project period. I would also like to thank my co-supervisor Sondre Schnell for helping me develop the computational methods, and always inspiring me to work harder, and Olav Galteland for assisting me with programming in Python. In addition, I would like to give a special thanks to Jean-Marc Simon for welcoming me at the Université de Bourgogne, and for sharing his insight in fluctuation based calculations. I am grateful for all their support throughout the project, providing me with knowledge I will benefit from in the future.

Lastly, I would like to thank the rest of the students in the Thermodynamics research group and others that have been present at our weekly meetings, listening to my presentations, taking part in discussions and helping us carry our projects forward. A very special thanks goes to my accumulated particles: Laura Edvardsen and Kristine Veie, for showing me how to do that. I hope our friendship lasts forever. Computer resources have been provided by Faculty of Natural Sciences at NTNU and NOTUR.

Abstract

The total correlation integrals can provide useful information about the local structure of fluids, both for mixtures and for pure materials. These integrals are commonly referred to as Kirkwood-Buff integrals, due to their central role in the solution theory derived by Kirkwood and Buff. The solution theory was originally derived from molecular distribution functions in the grand canonical ensemble, but several methods exist for the purpose of the computation of Kirkwood-Buff integrals using other ensembles. The purpose of this master's thesis is therefore to evaluate the efficiency of different methods available for computation of Kirkwood-Buff integrals from finite size computer simulations. To determine the method with the best overall performance, four factors are considered: accuracy, precision, availability and computational cost. All methods analyzed here are based on extracting the Kirkwood-Buff integrals in the thermodynamic limit from the ones computed for different-sized finite volumes. The finite-size Kirkwood-Buff integrals are either calculated from a running integral over the radial distribution function, or from fluctuations of number of particles in small subsystems inside a larger reservoir. Both methods give finite-size Kirkwood-Buff integrals that scale linearly with the inverse system size, making the values in the thermodynamic limit readily available. Recently, a scaling equation taking the finite size of the system into account was formulated (J Chem Phys. 2016;145(14):141103), and two ways of defining the subvolumes are available. The first is by placing the subvolumes at random locations inside the simulation box, while the second defines the subvolumes as lattice cells of a superimposed grid on the total simulation box. In this study, all different alternatives are investigated in detail in order to determine their effect on the Kirkwood-Buff integrals in the thermodynamic limit.

It was found that the superimposed lattice does not provide enough data points to precisely extract the Kirkwood-Buff integral in the thermodynamic limit, since the number available subvolume sizes is constrained by the total simulation box. In addition, the fluctuations are directly correlated for subvolumes placed right next to each other. The different techniques to extract the thermodynamic limit value of the Kirkwood-Buff integrals mainly resulted in similar values, but the ones involving fewer rounds of curve fitting were more precise. The randomly positioned spheres is the most flexible method. Combined with linear scaling it also provided the best overall accuracy, provided that the system is sufficiently far from the critical point. The radial distribution function is less sensitive to the critical point and the ensemble of the reservoir. The fluctuating volume in the isobaric-isothermal ensemble results in unwanted contributions to the fluctuations in the largest subsystems, and should therefore be avoided.

Sammendrag

Totalkorrelasjonsintegralene kan gi nyttig informasjon om den lokale strukturen til både blandinger og rene materialer. Disse integralene er vanligvis kalt Kirkwood-Buff integraler, grunnet deres sentrale rolle i teorien for løsninger utviklet av Kirkwood og Buff. Kirkwood og Buffs teori for løsninger ble opprinnelig utviklet fra molekylærdistribusjonsfunksjoner i det store kanoniske ensemble, men flere metoder er utviklet med hensikt å beregne Kirkwood-Buff integral ved bruk av andre ensembler. Hensikten med denne masteroppgaven er derfor å vurdere effektiviteten til et utvalg metoder som beregner Kirkwood-Buff integral fra simuleringer. For å avgjøre hvilken metode som er mest effektiv, er fire punkter unsersøkt: nøyaktighet, presisjon, tilgjengelighet og beregningskostnader. Alle metodene undersøkt her er basert på beregning av Kirkwood-Buff integralene i den termodynamiske grensen fra de beregnet for endelige volum av varierende størrelse. Kirkwood-Buff integralene med endelige størrelser er enten beregnet fra et integral over radialfordelingsfunksjonen, eller fra fluktuasjoner i antall partikler i subsystem plassert inne i et større reservoar. Begge metodene gir Kirkwood-Buff integral som skalerer lineært med den inverse størrelsen til systemet, slik at verdiene i den termodynamiske grensen er lett tilgjengelige. Nylig ble det presentert en skaleringslikning som også tar den endelige størrelsen av reservoaret med i beregningen (J Chem Phys. 2016;145(14):141103). To metoder for å definere subvolumene er også tilgjengelige. Den første plasserer sfæriske volum på tilfeldige posisjoner inne i simuleringsboksen, mens den andre definerer subvolumene som gitterceller i et innkopiert nett på den totale simuleringsboksen. Disse forskjellige alternativene er undersøkt i detalj for å bestemme hvilken effekt de har på verdien av Kirkwood-Buff integralene i den termodynamiske grensen.

Det ble funnet at gittercellene ikke ga tilstrekkelig antall datapunkter for å gi et presist estimat av Kirkwood-Buff integralene i den termodynamiske grensen, fordi antall tilgjengelige størrelser av subvolum er begrenset av den totale simuleringsboksen. I tillegg er fluktuasjonene korrelerte i subvolum plassert rett ved siden av hverandre. De forskjellige teknikkene brukt for ekstrapolering av Kirkwood-Buff integralene til den termodynamiske grensen ga i hovedsak de samme verdiene, men de som involverte færre kurvetilpasninger resulterte i høyere presisjon. De tilfeldig plasserte sfæriske subvolumene er mest fleksible, kombinert med lineær skalering gir denne metoden også best nøyaktighet, gitt at systemet er langt nok unna det kritiske punktet. Radialfordelingsfunksjonen er mindre sensitiv til det kritiske punktet, og til ensemblet til reservoaret. Det isobarisk-isotermiske ensemble ga uønskede bidrag til fluktuasjonene i de største subvolumene, og burde derfor unngås.

Contents

1 Introduction

The connection between macroscopic thermodynamic properties and microscopic behaviour is the main subject of statistical mechanics. The microscopic behaviour is described by the molecular distributions, which contain local information about the correlation between densities at different points in the system [1]. For homogeneous systems, allowed to exchange particles with the surroundings, these molecular distribution functions are interpreted as the fluctuations of number of particles. Two systems at equilibrium, that are allowed to exchange particles, will have constant chemical potential [2]. If also the temperature is constant, the system can be regarded as grand canonical, meaning that the fluctuations in number of particles are described by the grand canonical partition function. Through the fundamental equations of statistical mechanics, the partition function is in turn connected to a number of macroscopic properties, like the compressibility, derivative of chemical potential, partial molar enthalpies and partial molar volumes [1]. One important application of this coupling is the solution theory formulated by Kirkwood and Buff in 1951 [3]. Central to this theory are the Kirkwood-Buff integrals calculated from the integral of the radial distribution function over volume

$$G_{ik} \equiv \int_0^\infty \left[g_{ik}^\infty(r) - 1\right] 4\pi r^2 \mathrm{d}r \tag{1.1}$$

These are especially important when simulating molecular systems since they give access to a variety of thermodynamic properties, that normally are not available by simply analyzing the simulation trajectories [4]. Obtaining these properties is typically even more complicated for mixtures, when they are functions of composition as well. The solution theory by Kirkwood and Buff [3] has therefore been applied in a number of studies of simulations of aqueous solutions [5] [6] [7], and in parameterization of force fields [8] [9] [10].

The theory was originally derived for open systems of infinite size. This presents some challenges since these conditions are normally not available during computer simulations. In molecular dynamics simulations, the systems are closed and finite, and while Monte Carlo is able to create open systems, these are also constrained to a finite, and often small number of particles. It has also been shown to be generally difficult to use Monte Carlo to simulate high density systems, due to inefficient particle insertion procedure [11] [12] [13] [14]. Since infinite particle distances are not normally available, several studies have simply truncated the integral in equation (1.1) at finite distances [10] [15] [16]. However, this approximation does not provide accurate estimates of the Kirkwood-

Buff integrals [4]. When the system is very far from thermodynamic limit, meaning that it not large enough to reproduce macroscopic properties, the truncation method can lead to significant errors. Several methods have been proposed to account for this, and normally, a correction for the radial distribution function $g_{ik}^{\infty}(r)$ is suggested [17] [18]. Some of these methods are rather numerically involved and difficult to implement, and the results it that the use of Kirkwood-Buff integrals no longer is a simple way to obtain the macroscopic properties.

Another part of the Kirkwood-Buff theory, that has been receiving increased attention, is the expression for Kirkwood-Buff integrals for finite volumes. Through the interpretation of the molecular distribution functions, this expression is directly coupled to the fluctuations of number of particles in open and finite systems

$$\begin{aligned} G_{ik}^{V} &= \frac{1}{V}\int_{V}\int_{V}\left[g_{ik}^{\infty}(r_{12})-1\right]\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_1 \\ &= V\left(\frac{\langle N_i N_k\rangle - \langle N_i\rangle\langle N_k\rangle}{\langle N_i\rangle\langle N_k\rangle} - \frac{\delta_{ik}}{\langle N_i\rangle}\right) \end{aligned} \tag{1.2}$$

Many recently published papers have calculated estimates of the Kirkwood-Buff integrals in the thermodynamic limit ($V \to \infty$) from the ones calculated for various sizes of these finite volume integrals [8] [19] [20] [21]. Methods calculating G_{ik}^{V} from particle fluctuations, and by integration of $g_{ik}^{\infty}(r)$ exist, and they are both normally followed by extrapolation in order to obtain the thermodynamic limit value, G_{ik}. Some comparisons of the efficiency of these methods exists, but they mostly focus on the differences between integration of $g_{ik}^{\infty}(r)$, and fluctuation based methods [8] [22] [23] [24]. Since several different methods for calculation of fluctuations also are available [25] [20] [23], a comparison between these is also of interest. What still remains is therefore a thorough evaluation of the efficiency of the different methods, answering the question: which method shows the best overall accuracy, precision, availability and computational effort?

This master's thesis is organized as follows. Section 2 presents the theoretical background of the calculated properties. Section 3 presents the computational background of molecular dynamics simulations, and the different procedures used to obtain the Kirkwood-Buff integrals in the thermodynamic limit. Section 4 describes how the simulations and calculations were performed, while results and discussion are presented in section 5, followed by conclusion in section 6.

2 Theoretical Background

The following section presents a description of properties often reported for mixtures, together with models and methods available to obtain these. This includes partial molar properties and isothermal compressibility. A section describing different ensembles often applied in simulations and experiments, and their connection to local structure and fluctuations of thermodynamic quantities, is also included. Thereafter follows a section describing molecular distribution functions, and one of its most important applications, the Kirkwood-Buff theory of solutions.

2.1 Properties of Mixtures

Mixtures act differently than pure materials, giving rise to some additional, mixture specific properties. To be able to accurately describe and understand the behaviour of mixtures, it is therefore necessary to define some quantities and models that characterize these properties [26]. In most engineering applications, the temperature, T, and pressure, p, are constant. Consequently, a general extensive property, B, of a multicomponent system is conveniently defined by

$$B = f(T, p, n_1, ..., n_M) \tag{2.1}$$

where n_i is the number of moles of component i in a mixture of M different components.

2.1.1 Partial Molar Properties

One central central derivative of B is the partial molar property, defined by

$$B_i \equiv \left(\frac{\partial B}{\partial n_i}\right)_{T,p,n_j[i]} \tag{2.2}$$

These can be described as how one property of a mixture changes with the number of moles of component i in the mixture while T, p and the number of all components except i are held constant (indicated by subscript $n_j[i]$) [2]. The partial molar properties will be zero order in number of moles, making them intensive properties. This means that they do not depend on the size of the system, but that they can depend on the temperature, pressure and composition. Naturally, the partial molar properties are also properties of the specific mixture considered, and not simply properties of the component *i*.

The partial molar properties can be evaluated by a number of different methods [26]. One of the most effective is to express the property B as a function of composition, and evaluate B_i by differentiation. This can be done by starting off with expressing the total differential of the intensive property $\widetilde{B} = B/n$ as

$$\mathrm{d}\widetilde{B} = \left(\frac{\partial \widetilde{B}}{\partial T}\right)_{p,x} \mathrm{d}T + \left(\frac{\partial \widetilde{B}}{\partial p}\right)_{T,x} \mathrm{d}p + \sum_{j\neq i} \left(\frac{\partial \widetilde{B}}{\partial x_j}\right)_{T,p,x[j,i]} \mathrm{d}x_j \tag{2.3}$$

where x_j is the mole fraction of component *j*, and the subscript $x[j,i]$ indicates that all mole fractions in the set $(x_1,...,x_M)$ are held constant, except x_j and x_i. After some algebra, explained in detail in reference [26], B_i yields

$$B_i = \frac{B}{n} - \sum_{j\neq i} x_j \left(\frac{\partial\left(\frac{B}{n}\right)}{\partial x_j}\right)_{T,p,x[j,i]} \tag{2.4}$$

which is an expression that can be applied to all multicomponent systems for any property of interest [27].

One central partial molar property that provides useful information about the role of the different components in a mixture is the partial molar Gibbs energy, defined as the chemical potential, $\mu_i \equiv G_i$ [26]. This property is particularly useful because it can describe both phase behaviour and driving forces of different systems [28]. Because of its importance in both thermodynamics and solution chemistry, several methods exist for the purpose of its calculation. Currently, the best known method for calculation of chemical potential is the Widom particle insertion method [29]. An ideal mixture is characterized by equal forces acting between all molecules [2]. For such mixtures, there exist theoretical predictions for most state functions and thermodynamic properties, and chemical potential is no exception. It is expressed as

$$\mu_i^{\mathrm{ID}} = RT\ln x_i + \mu_i^* \tag{2.5}$$

where μ_i^* is the chemical potential of pure component *i* at current pressure and temperature of the mixture. The partial derivative of the chemical potential with mole fraction in an ideal mixture is therefore given by

$$\left(\frac{\partial \mu_i^{\mathrm{ID}}}{\partial x_i}\right)_{T,p} = \frac{RT}{x_i} \tag{2.6}$$

2.1.2 Isothermal Compressibility

Another property that is often reported both for mixtures and pure materials, principally because it can be measured by simple experiments, is the isothermal compressibility [30]. It can be expressed as follows

$$\kappa_T \equiv -\frac{1}{V}\left(\frac{\partial V}{\partial p}\right)_T \tag{2.7}$$

where V is volume. This property is easily accessible since the derivative can be can be determined numerically by constant temperature experiments that provide data for volume as a function of pressure. For simple mixtures, an approximate value of the isothermal compressibility is available through the expression

$$\kappa_T = \phi_i \kappa_{T,i}^0 + \phi_k \kappa_{T,j}^0 \tag{2.8}$$

where ϕ_i is the volume fraction of component i in the solution, calculated from the product of the bulk density of this component and its partial molar volume, $\rho_i V_i$, at the given composition [9]. $\kappa_{T,i}^0$ is the isothermal compressibility of a pure system of component i. For complex mixtures, equation (2.8) will not give very accurate estimates of the isothermal compressibility, but it can provide an indication of its order of magnitude.

2.2 Ensembles and Fluctuations

An ensemble can be defined in two ways. Either it describes the set of variables that are controlled throughout the experiment or simulation, or it refers to a collection of all possible microstates of the system, called ensemble members [31]. Some of the most common ensembles are the micro canonical ensemble, grand canonical ensemble, canonical ensemble and the isobaric-isothermal ensemble [32]. The micro canonical ensemble is represented by the fixed variables: number of particles N, V, and energy, E. The grand canonical ensemble represents an open system in a heat bath, which means that μ, V and

T are held constant. The canonical ensemble represents a closed system in a heat bath, meaning that *N*, *V* and *T* are constant. The isobaric-isothermal ensemble, with the fixed variables *N*, *p* and *T*, is the one most used in experiments and engineering applications. All properties that are not controlled by the ensemble will therefore fluctuate around a mean value. The magnitude of such fluctuations for a general property *B*, can be measured by

$$\sigma_B^2 = \langle (B - \langle B \rangle)^2 \rangle = \langle B^2 \rangle - \langle B \rangle^2 \tag{2.9}$$

Each ensemble is also represented by a particular partition function, which is a sum of Boltzmann factors specifying how particles are partitioned throughout the accessible states [31]. These partition functions provide a connection between macroscopic thermodynamic properties and microscopic models, since a number of thermodynamic properties can be described by the ensemble's partition function, or its derivatives. Take as an example, the grand canonical partition function Ξ, which is coupled to the product of pressure and volume through the fundamental connection

$$p(T,V,\mu)V = k_\mathrm{B} T \ln \Xi(T,V,\mu) \tag{2.10}$$

where k_B is the Boltzmann constant [33]. The second derivative of the partition function is in turn related to fluctuations of particle number through

$$(k_\mathrm{B} T)^2 \left(\frac{\partial^2 \ln \Xi(T,V,\mu)}{\partial \mu^2} \right)_{T,V} = \langle N^2 \rangle - \langle N \rangle^2 \tag{2.11}$$

where the brackets $\langle \cdots \rangle$ denote ensemble averages. Thus, by measuring something as simple as average number of particles, a large variety of thermodynamic properties are available [33]. The isothermal compressibility is actually available through fluctuations both in the μVT and in the NpT ensemble. By fluctuation of number of particles in the μVT ensemble, it is expressed as

$$\kappa_T = \frac{V}{T k_\mathrm{B}} \frac{\langle N^2 \rangle - \langle N \rangle^2}{\langle N \rangle^2} \tag{2.12}$$

while it also is available from volume fluctuations in the NpT ensemble by

$$\kappa_T = \frac{1}{Tk_\mathrm{B}} \frac{\langle V^2 \rangle - \langle V \rangle^2}{\langle V \rangle} \tag{2.13}$$

This connection means that the relative fluctuations in volume in the NpT ensemble are equal to the relative fluctuations in number of particles in the μVT ensemble, provided that the average volume of the NpT system, $\langle V \rangle$, is equal to the constant volume of the μVT system, V.

In real life, these fluctuations arise naturally, as a result of which other properties are held constant. When performing simulations, the goal is to mimic this behaviour properly, and to make sure that the fluctuations follow the distributions determined by the partition functions. Having the correct fluctuation distributions becomes especially important when measuring equilibrium averages that are sensitive to, or dependent on fluctuations [34]. A region in which fluctuations should be treated with particular care, is close to the critical point of the system. For fluids, this is at the given temperature, pressure and volume, where the phase boundary between liquid and gas ceases to exist. The region is characterized by large fluctuations in local thermodynamic variables, and in the point itself, the amplitude of the fluctuations will diverge [35]. One practical example of this is the isothermal compressibility, which becomes infinite in the critical point. Since it is directly coupled to fluctuations through equations (2.12)-(2.13), these clearly also need to reflect this behaviour [36].

2.3 Molecular Distribution Functions

The above introduced connection between macroscopic thermodynamic properties and microscopic behavior is the main subject of molecular theories of liquids and liquid mixtures [1]. Central to both these theories are the molecular distribution functions, since these convey local information about the densities, and the correlation between densities at different points in the system.

A molecule's location in space can be described by the vector $\mathbf{r} = (x, y, z)$, where x, y and z are Cartesian coordinates [33]. However, this description of the configuration is only complete if the molecule is spherical. If the molecule is non-spherical, an additional three angles are needed to describe the orientation of the molecule, specified by the vector $\boldsymbol{\omega} = (\phi, \theta, \psi)$, where ϕ, θ and ψ is the set of Euler angles. For some cases, the molecule's internal degrees of freedom, which consists of rotation and vibration, also need to be included in the description. However, for the purpose of this text, the description of the spatial coordinates of rigid molecules is sufficient. The molecular distribution functions are here presented for a two-component mixture, consisting of the species i and k, in a grand canonical system.

Two central functions in this description are the singlet distribution function, $\rho_i^{(1)}(\mathbf{r}_1)$, and the pair distribution function, $\rho_{ik}^{(2)}(\mathbf{r}_1, \mathbf{r}_2)$. These both provide useful information when integrated over the entire volume V. The first can be regarded as the local density at the point $\mathbf{r}_1$. For an infinitesimal small volume $\mathrm{d}\mathbf{r}_1$, located at the point $\mathbf{r}_1$, the product $\rho_i^{(1)}(\mathbf{r}_1)\mathrm{d}\mathbf{r}_1$ can therefore be viewed as the average number of particles in that volume element. Consequently, the integration of the product over the entire volume V becomes the average number of particles in the entire system

$$\int_V \rho_i^{(1)}(\mathbf{r}_1)\mathrm{d}\mathbf{r}_1 = \langle N_i \rangle \tag{2.14}$$

The product $\rho_{ik}^{(2)}(\mathbf{r}_1, \mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2$ is interpreted as the average number of pairs of species i and k occupying the volume elements $\mathrm{d}\mathbf{r}_1$ and $\mathrm{d}\mathbf{r}_2$. The double integral of this product over V therefore becomes the average number of pairs occupying the entire system

$$\int_V \int_V \rho_{ik}^{(2)}(\mathbf{r}_1, \mathbf{r}_2)\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 = \langle N_i N_k \rangle - \delta_{ik}\langle N_i \rangle \tag{2.15}$$

where δ_{ik} is the Kronecker-delta, which is equal to one when $i = k$ and zero otherwise. See reference [1] for full derivation of above equations. The fluctuations of number of particles of a two-component mixture in an open volume are therefore expressed as

$$\int_V \mathrm{d}\mathbf{r}_1 \int_V \mathrm{d}\mathbf{r}_2 \left[\rho_{ik}^{(2)}(\mathbf{r}_1, \mathbf{r}_2) - \rho_i^{(1)}(\mathbf{r}_1)\rho_k^{(1)}(\mathbf{r}_2)\right] = \langle N_i N_k \rangle - \langle N_i \rangle\langle N_i \rangle - \delta_{ik}\langle N_i \rangle \tag{2.16}$$

For homogeneous and isotropic mixtures, the molecules are translational and rotational invariant. This means that the singlet molecular distribution function is expected to have the same value at any point $\mathbf{r}_1$ in the fluid. This value is the number density, given by $n_i = \langle N_i \rangle / V$, and will be equal to the macroscopic number density ρ_i. The pair distribution function can also be written in terms of the bulk molecular concentration, by the combination

$$\rho_{ik}^{(2)}(\mathbf{r}_1,\mathbf{r}_2) = \rho_i \rho_k g_{ik}(\mathbf{r}_1,\mathbf{r}_2) \tag{2.17}$$

but this expression includes the additional term $g_{ik}(\mathbf{r}_1,\mathbf{r}_2)$, which is identified as the locational pair correlation function.

2.3.1 Radial Distribution Function

The assumption of a homogeneous mixture also has an effect on the above introduced pair correlation function. For instance, if $\mathbf{r}_1$ is chosen at the origin, the relative orientation of the second particle is of no importance. By introducing the variable $r_{12} =| \mathbf{r}_1 - \mathbf{r}_2 |$, the pair correlation function can be expressed explicitly as a function of the distance r_{12}. This function is commonly referred to as the radial distribution function (RDF), and is central to theories of the liquid state because it characterizes local structure of the system [37].

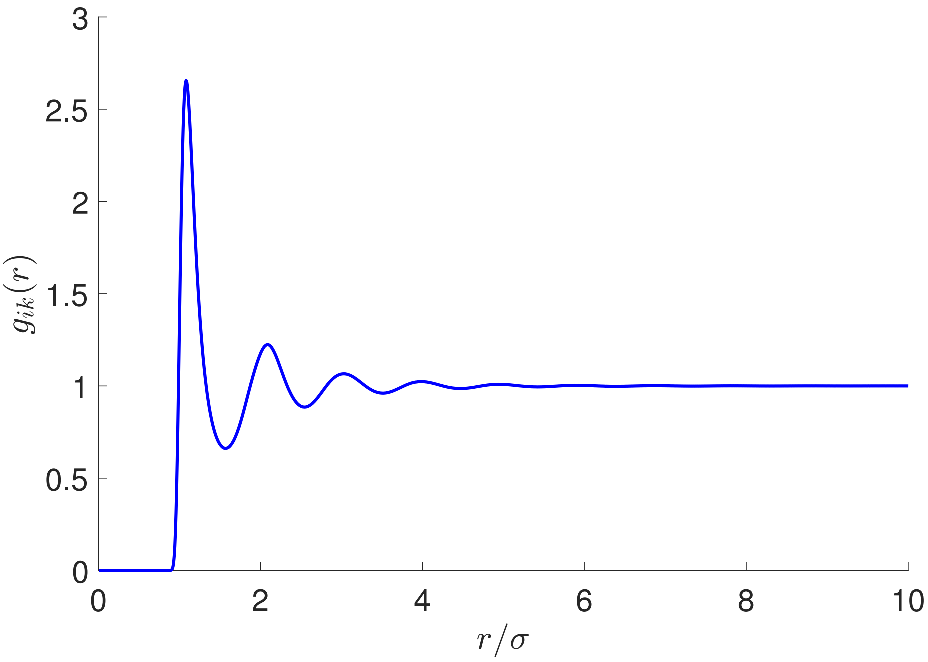

Figure 2.1: Radial distribution function of a pure Lennard-Jones system with parameters $\varepsilon = 1$ and $\sigma = 1$ and reduced density $\rho\sigma^3 = 0.7$.

The physical meaning of this function is the probability of finding a molecule of type k at a distance r_{12} from a molecule of type i. This probability is expressed as relative to the ideal gas distribution [38]. Subsequently, the RDF of an ideal gas in an open system is $g_{\text{ideal}}^{\infty}(r_{12}) = 1$ [39]. Since the correlation between the local densities at two points become smaller with the separation distance, the RDF should also for non-ideal, open systems reach unity for large values of r [1]. For closed and finite systems, the RDF of an ideal gas is $g_{\text{ideal}}(r_{12}) = 1 - 1/N$. Consequently, non-ideal closed systems also will approach this value for large values of r. This difference, and where it originates from is further explained in the applications of RDF in section 3.3.1.

The shape of the RDF is highly dependent on the density. Take for example, a fluid consisting of identical particles described by the Lennard-Jones potential (described in section 3.1.1). For all densities, the first peak of the RDF will appear close to the minimum of the Lennard-Jones potential, which is located at $r_{\text{min}} = 2^{1/6}\sigma$. For dilute systems this will be the only peak, but at larger densities, new peaks will develop and occur nearly at multiples of σ, $r \approx \sigma, 2\sigma, 3\sigma, \ldots$, which arise due to behaviour of packing of spherical molecules about another molecule [1]. The point where the RDF approaches unity is regarded as the structural correlation length, ξ, of the fluid. It represents the extent of the correlation between the local densities at two points in the mixture [1]. Figure 2.1 shows the RDF for a system with reduced density $\rho\sigma^3 = 0.7$ and structural correlation length $\xi \approx 7\sigma$.

2.4 Kirwood-Buff Theory of Solutions

One central application of the RDF is the powerful and general solution theory presented by Kirkwood and Buff [3]. This theory provides direct relations between various thermodynamic properties and the molecular distribution functions in the grand canonical ensemble. It can be used for mixtures of any number of components [4], and is widely applied in physical chemistry due to its ability to investigate complex, liquid mixtures. The theory defines the so-called Kirkwood-Buff (KB) integral

$$G_{ik} \equiv \int_0^{\infty} \left[g_{ik}^{\infty}(r) - 1\right] 4\pi r^2 \mathrm{d}r \tag{2.18}$$

which is the integral of $g_{ik}^{\infty}(r_{12})$ over of an infinite, open system. The physical meaning of these integrals is the measure of mutual affinity between component i and k in the mixture. The connection to the local structure becomes evident in the expression for the KB integrals for finite volumes, which is obtained by combining equations (2.16)-(2.17) and dividing by $\rho_i \rho_k V$

$$\begin{aligned} G_{ik}^V &= \frac{1}{V} \int_V \int_V \left[g_{ik}^{\infty}(r_{12}) - 1 \right] \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_1 \\ &= V \left(\frac{\langle N_i N_k \rangle - \langle N_i \rangle \langle N_k \rangle}{\langle N_i \rangle \langle N_k \rangle} - \frac{\delta_{ik}}{\langle N_i \rangle} \right) \end{aligned} \tag{2.19}$$

Full derivation of above expression can be found in reference [4].

While the KB integrals themselves can provide insight to local structure, they are most useful in their relation to the thermodynamic quantities, such as partial derivative of chemical potential, isothermal compressibility, and partial molar volumes of the components in the mixture.

For pure systems, some of these properties can be calculated directly from fluctuations of number of particles in the system (see section 2.2). This is not as straight forward for multicomponent systems, since the intermolecular interactions can be very different for the different components in the mixture. This is where the KB integrals become particularly useful. Through the connection provided by the grand canonical partition function (equation (2.10)), the thermodynamic properties are instead expressed as combinations of the KB integrals of the mixtures. Presented here, are the partial derivative of the chemical potential with mole fraction, isothermal compressibility and the partial molar volumes

$$\left(\frac{\partial \mu_i}{\partial x_i} \right)_{T,p} = \frac{k_\mathrm{B} T}{x_i (1 + x_i c_k (G_{ii} + G_{kk} - 2G_{ik})} \tag{2.20}$$

$$\kappa_T = \frac{T}{k_\mathrm{B}} \frac{1 + c_i G_{ii} + c_k G_{kk} + c_i c_k (G_{ii} G_{kk} - G_{ik}^2)}{c_i + c_k + c_i c_k (G_{ii} + G_{kk} - 2G_{ik})} \tag{2.21}$$

$$V_i = \frac{1 + (G_{kk} - G_{ik}) c_k}{c_i + c_k + c_i c_k (G_{ii} + G_{kk} - 2G_{ik})} \tag{2.22}$$

All these expressions require the KB integrals from open systems of infinite volume. The KB integrals calculated from equation (2.19) of finite sized volumes can not be used directly in equations (2.20)-(2.22), since they are not equal to the ones defined by equation (2.18). For homogeneous fluids in the limit $V \to \infty$, the system is translationally invariant, which means that the effects of orientation can be integrated out of the left hand side of equation (2.19) using the variable transformation $\mathbf{r}_2 \rightarrow \mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$. The double integral is then reduced to a single integral which yields the original expression for the KB integral in equation (2.18). However, for finite-size systems, the integration over $\mathbf{r}$ depends on $\mathbf{r}_1$ which means that the variable transformation is not possible. If a finite volume still was to be used in the integral equation (2.18), it would give $-\delta_{ik}/c_i$, which does not represent the true values of neither G_{ik}, nor G_{ik}^V [39]. Therefore, to obtain the exact KB integrals of finite volumes, and approximate values of the KB integral in the thermodynamic limit, alternative methods have to be used. Some of these are introduced in the following section.

Another problem that arises when working with simulations of finite volumes is that the RDFs are calculated for closed and finite systems, while RDFs for open and infinite systems are required for the KB integrals. The main difference is the asymptotic behaviour of the RDF, which is different for open and closed systems (see section 2.3.1). This means that, not only does the integration provide incorrect answers, but the integrand also contains errors. However, several methods are available for the estimation of $g_{ik}^{\infty}(r)$ from the RDF calculated for the closed and finite system $g_{ik}(r)$ [8] [19] [20].

3 Computational Background

The main tool used in this work is the simulation method molecular dynamics. The most important aspects of this method is therefore presented in the following section. Then follows a description of different approaches used for computation of KB integrals. All methods include computation of KB integrals for finite volumes followed by different techniques to obtain their values in the thermodynamic limit.

3.1 Molecular Dynamics

A system of N particles can be described as a classical many-body system if the motions of the particles obey the laws of classical mechanics. Such a system can, in each point in time, be described by $3N$ coordinates $\mathbf{r}$ and $3N$ momenta $\mathbf{p}$, which together spans the $6N$-dimensional phase space [37]. The system's energy is expressed by the Hamiltonian, $\mathscr{H}$, which under the classical description can be written as the sum of kinetic- and potential energy functions

$$\mathscr{H}(\mathbf{r},\mathbf{p}) = \mathscr{K}(\mathbf{p}) + \mathscr{V}(\mathbf{r}) \tag{3.1}$$

The kinetic energy is usually on the form

$$\mathscr{K}(\mathbf{p}) = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} \tag{3.2}$$

where m_i is the mass of particle i. The potential energy function, however, contains information about the particles' interactions, and will largely depend on the types of particles in the system. Molecular dynamics (MD) is a simulation method that utilizes the interactions arising from the potential energy function to describe the forces in an equation of motion. In that way, is attempting to mimic the motion of particles by computing trajectories consisting of each particle's positions and velocities in the entire time-evolution of the system [40]. Its governing equation of motion is Newton's second law

$$m_i \frac{\mathrm{d}^2\mathbf{r}_i}{\mathrm{d}t^2} = \mathbf{F}_i \tag{3.3}$$

where $\mathbf{r}_i$ is the coordinates of particle i and $\mathbf{F}_i$ is the sum of forces acting on particle i.

The forces are computed from the potential energy function, $\mathcal{V}(\mathbf{r})$, which is modelled by the interaction potential (described in section 3.1.1) and depends on the set of positions of the N particles. Since the masses of the electrons are much smaller than the masses of the nuclei, the electrons can more rapidly adjust to changes in nuclear position and relax to their ground state much faster than the nuclei. As a result, the electronic and nuclear motions can be separated. Therefore, the Born-Oppenheimer approximation can be assumed to operate, meaning that the nuclear coordinates are the only dynamical variables included in the calculation of the interactions [41]. In order to properly describe the time evolution of the motion of the particles, and achieve accurate results, a good description of these interactions is needed.

3.1.1 Interaction Potential

For a system of N particles, the potential energy as a function of the position of all the particles in the system $\mathbf{r} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N]$ can be expressed as a sum of different terms

$$\mathcal{V}(\mathbf{r}) = \sum_{i=1} v_1(\mathbf{r}_i) + \sum_{i=1} \sum_{j=i+1} v_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i=1} \sum_{j=i+1} \sum_{k=j+1} v_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + ... \tag{3.4}$$

where the first term represents the effect of an external field while the remaining are due to interactions between particles [42]. The third term, namely the triplet interactions, is rarely included in MD simulations because its computation is very time consuming. The second term is therefore the most important, and gives rise to the so-called pair potential approximation. The summation of this term indicates the summation over all distinct pairs of i and j without computing any pair twice. The pair potential approximation expresses this by

$$\mathcal{V}_{\text{pair}}(\mathbf{r}) = \frac{1}{2} \sum_{i \neq j} v_2(|\mathbf{r}_i - \mathbf{r}_j|) \tag{3.5}$$

where the sum is over all the pairs, but a factor of one half is included to make sure that no pair is counted twice, and $i \neq j$ indicates that one particle's interaction with itself should not be counted either. The pair potential is also assumed to be isotropic, which is why it can be expressed as a function of the magnitude of the pair separation $|\mathbf{r}_i - \mathbf{r}_j|$ (or simply r_{ij}) [42].

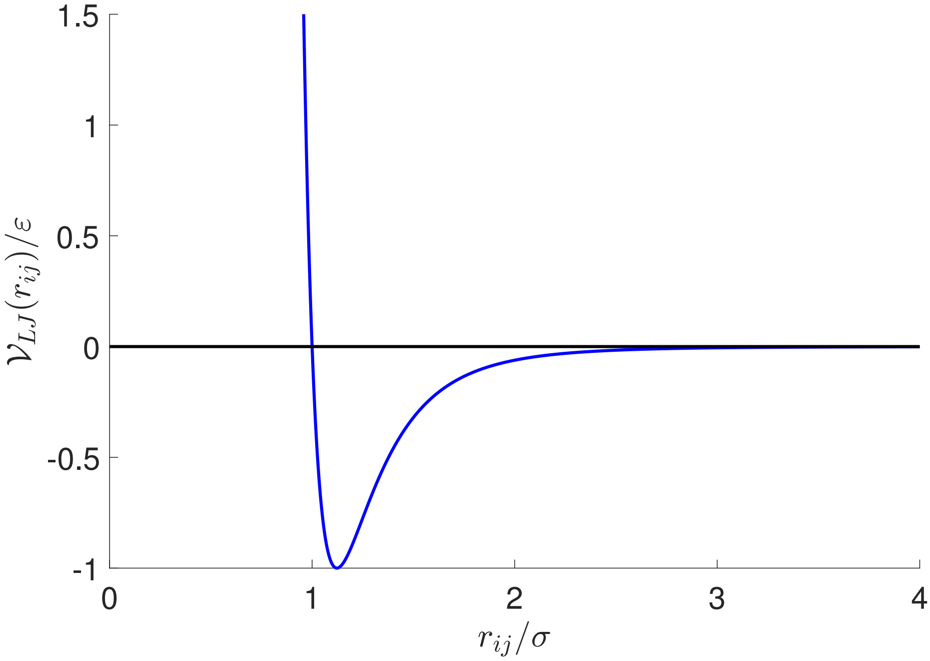

Figure 3.1: Lennard-Jones potential curve for $\varepsilon = 1$ and $\sigma = 1$.

For spherically symmetric particles with no charge, the pair potential can be visualized as an energy curve as a function of the separation distance, r_{ij}. The typical features of such a potential curve are as follows. There is an attractive tail at large separations, which is approaching zero at infinite separation. This is due to the correlations of the electron clouds of the atoms, namely 'London' and 'van der Waals' dispersions. At shorter separations, the energy is reduced until it reaches a minimum, before it rapidly increases. This represents the repulsive wall at small separations, due to overlap of electron clouds, namely the 'Pauli' repulsion [42] [43]. One often used model, that represents this empirical behavior, is the Lennard-Jones (LJ) potential

$$\mathcal{V}_{\mathrm{LJ}}(r_{ij}) = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \tag{3.6}$$

where the term r_{ij}^{-6} represents the long-range attractive tail, and the r_{ij}^{-12} is the repulsive term. The parameters σ and ε are the collision diameter, and the potential well depth respectively (see figure 3.1 for illustration). When these are chosen appropriately for the components of interest, the LJ potential provides a reasonable description of the pair potential [42].

For charged particles, an additional term is needed to describe the interactions arising from charges. The pairwise Coulombic interactions describe these as

$$\mathcal{V}_{\text{Coulomb}}(r_{ij}) = \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \tag{3.7}$$

where q_i is the charge of particle i and ε_0 is the dielectric constant of the interparticle medium [38]. The total pair potential then becomes

$$\begin{aligned}\mathcal{V}_{\text{pair}}(\mathbf{r}) &= \frac{1}{2}\sum_{i\neq j}\left(\mathcal{V}_{\text{LJ}}(r_{ij}) + \mathcal{V}_{\text{Coulomb}}(r_{ij})\right)\\ &= \frac{1}{2}\sum_{i\neq j} 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] + \frac{1}{2}\sum_{i\neq j}\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}\end{aligned} \tag{3.8}$$

3.1.2 Boundary Conditions

Treatment of boundaries is of great importance in molecular simulations. Normally, the aim is to compute macroscopic properties, but still keep the number of particles at a minimum. The result is that surface effects will influence a large number of the particles, and making these experience different forces than the particles in the bulk [38]. Since we often are interested in the bulk phase, these surface effects are unwanted. Introducing periodic boundary conditions (PBC) solves this problem. The simulation cell is then repeated through space, representing an infinite lattice. This means that when a particle exits the simulation box on one side, it reappears on the opposite side [42].

3.1.3 Truncation of Potential

The most time consuming part of computer simulations of molecular systems is the computation of non-bonded interactions [37]. Even when using the pair potential approximation, the interactions are still computed for every pair of particles in the system, making the computational effort scale as $\mathcal{O}(N^2)$. For many systems, this is an unnecessary effort, since the forces often become unimportant at large separation distances. One example is the LJ potential, which falls off very rapidly with the distance r_{ij}, and already at 2.5σ has just 1% of its value at σ. This can also be observed in figure 3.1 where the potential curve is close to zero at this point. Simplifications in computations of the interactions are therefore often applied, in order to lower the computational cost. One simplification often used for LJ systems is the truncated potential, which employs a cut-off distance for the potential. This means that particles separated by a distance longer than the cut-off distance experience zero interaction from each other.

Another principle that normally is applied together with the cutoff, is the minimum image convention, which becomes necessary due to the PBCs [38]. The idea behind this principle is that each particle should at most "see" one image of every other particle in the system. The cut-off can therefore not be larger than half the size of the total simulation box, to make sure that no particle sees its own image, or one of the other particles twice. For LJ systems, the cut-off is often set to 2.5σ. Since the forces in this potential are short-ranged, this mostly gives sufficiently small errors in the dynamics of the system, meaning that they still are representative of the macroscopic system the model is designed to simulate [44]. However, if long-range forces are present, a cut-off this short will lead to much larger errors. For electrostatic interactions, the range of the forces is often greater than half the box length, meaning that using a cut-off at all would lead to errors. A method often used to solve this problem is the Ewald summation [45], where each charge in the system is allowed to interact with an infinite array of periodic images of the simulation cell, making the cut-off for these interactions effectively infinite [38]. The Coulombic term, from equation (3.8), for interactions between the particles in the central box of length L and the particles in all surrounding image boxes is therefore written as as

$$\mathcal{V}_{\text{pair}}(\mathbf{r}) = \frac{1}{2} \sum_{|\mathbf{n}|=0}' \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{q_i q_j}{4\pi\varepsilon_0 \mid \mathbf{r}_{ij} + \mathbf{n} \mid} \tag{3.9}$$

where $\mathbf{n} = [n_x L, nyL, n_z L]$ is the position of the image box, and $\mathbf{r}_{ij}$ is the distance between the charges i and j in Cartesian coordinates. The prime on the first sum indicates that the series does not include the interaction $i = j$ for $\mathbf{n} = 0$, since this would represent the same particle within the central box [38]. In the current version of equation (3.9), the charges are represented as point charges. The Ewald method converts this to a sum over the interactions between the charges, plus a neutralizing charge distribution of equal magnitude, but of opposite sign. These charge distributions are represented by Gaussian distributions, usually on the form

$$\frac{q_i \alpha^3}{\pi^{(3/2)}} \exp(-\alpha^2 r^2) \tag{3.10}$$

The motivation behind this, is that the slowly converging sum in equation (3.9) is split into two series, each of which converge much more rapidly. The final equation therefore contains these two sums, and two additional correctional terms, written as

$$
\begin{aligned}
\mathcal{V}_{\text{Ewald}}(\mathbf{r}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} & \Bigg(\sum_{|\mathbf{n}|=0}^{\infty}{}' \frac{q_i q_j}{4\pi\varepsilon_0} \frac{\operatorname{erfc}(\alpha \mid \mathbf{r}_{ij} + \mathbf{n} \mid)}{\mid \mathbf{r}_{ij} + \mathbf{n} \mid} \\
& + \sum_{k\neq 0} \frac{1}{\pi L^3} \frac{q_i q_j}{4\pi\varepsilon_0} \frac{4\pi^2}{k^2} \exp\left(-\frac{k^2}{4\alpha^2}\right) \cos(\mathbf{k}\cdot\mathbf{r}_{ij}) \\
& - \frac{\alpha}{\sqrt{\pi}} \sum_{k=1}^{N} \frac{q_k^2}{4\pi\varepsilon_0} + \frac{2\pi}{3L^3} \left| \sum_{k=1}^{N} \frac{q_k}{4\pi\varepsilon_0} r_k \right|^2 \Bigg)
\end{aligned}
\tag{3.11}
$$

The first term represents the real space summation, which includes the error function

$$
\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2)\mathrm{d}t \tag{3.12}
$$

where α is the width of the Gaussian curves. A higher value of α leads to faster converge for the real space summation. The second sum is performed in the reciprocal space, and therefore consists of the reciprocal vectors $\mathbf{k} = 2\pi\mathbf{n}/L$. This summation also contains the α parameter, and converges more rapidly for lower α, in contrast to the first summation, which converges more rapidly for higher α. A point of balance therefore needs to be determined in order to optimize the convergence of the combined summation. In addition, two correctional terms are needed. The first subtracts the sum of Gaussian functions that interact with themselves in real space, while the second is only included if the surrounding medium is vacuum.

3.1.4 Velocity Verlet

After computing the forces acting on all the particles in the system, the equation of motion (equation (3.3)) needs to be integrated. The usual approach is a finite difference method, namely the velocity Verlet integrator scheme [37]. This method computes the positions and velocities by the following equations

$$
\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t\mathbf{v}(t) + \tfrac{1}{2}\Delta t^2 \mathbf{a}(t) \tag{3.13}
$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \tfrac{1}{2}\Delta t[\mathbf{a}(t) + \mathbf{a}(t+\Delta t)] \tag{3.14}$$

where $\mathbf{v}$ are the velocities, $\mathbf{a}$ are the accelerations and Δt is the time step. Since the computation of the velocity by equation (3.14) requires the acceleration at both t and $t+\Delta t$, this method also involves a predictor step. This is achieved by using

$$\mathbf{v}(t+\tfrac{1}{2}\Delta t) = \mathbf{v}(t) + \tfrac{1}{2}\Delta t\mathbf{a}(t) \tag{3.15}$$

before computing forces and accelerations at time $t+\Delta t$, and then completing the velocity calculation with

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\tfrac{1}{2}\Delta t) + \tfrac{1}{2}\Delta t\mathbf{a}(t+\Delta t) \tag{3.16}$$

where equation (3.15) is inserted for $\mathbf{v}(t+\frac{1}{2}\Delta t)$. The uses of the velocity Verlet algorithm are widespread, and it is probably the most attractive method for solving the equations of motion [42]. It is time reversible and area preserving (which leads to low long-term energy drift) in addition to being efficient, numerically stable and simple, which all are desirable traits [37].

3.1.5 Time Averages and Ensemble Averages

The aim of a MD simulation is usually to compute thermodynamic properties of a system. To measure these properties, they need to be expressed as a function of the position and momenta of the particles in the system. When equilibrium or steady state is reached, the properties are independent of time, but will fluctuate around a mean value. Only then can properties like pressure, temperature and internal energy be expressed as averages over time [37]. For the general property, the time average over the dynamical history of the system can be expressed as [32]

$$\langle B \rangle_t = \lim_{\tau\to\infty} \frac{1}{\tau} \int_0^{\tau} B[\mathbf{r}^N(t), \mathbf{p}^N(t)]\mathrm{d}t \tag{3.17}$$

Another alternative is to express the property as an average over all the members of an ensemble at one snapshot in time, namely an ensemble average [42]. The ensemble members are regarded as replicas of the physical system of interest, and each member's distribution of phase points are described by the partition function of the given ensemble.

3.2 Small System Method

In thermodynamics, the main difference between small systems and large systems is the surface area to volume ratio. Since this ratio is much larger for small systems, the surface effects become much more significant, and the properties can no longer be directly compared to thermodynamic properties of macroscopic systems [46]. This becomes clear by studying the system's extensive properties, which for small systems no longer are proportional to volume, but higher order functions of size and shape. As a result, classical thermodynamics can not be applied for systems far from the thermodynamic limit, without first doing some modifications [47]. The formalism developed by Hill [46] provides such modified equations through generalizing Gibbs equation to also apply for small systems. Hill's thermodynamics of small systems in the grand canonical ensemble can also be linked to the probability distributions of statistical mechanics. When the system is in the grand canonical ensemble, the particles are free to move in and out of the small system, which means that fluctuations in particle- and energy-density will arise [25]. These fluctuations are related to the second moments of the probability distribution and therefore also to fundamental thermodynamic quantities of small systems.

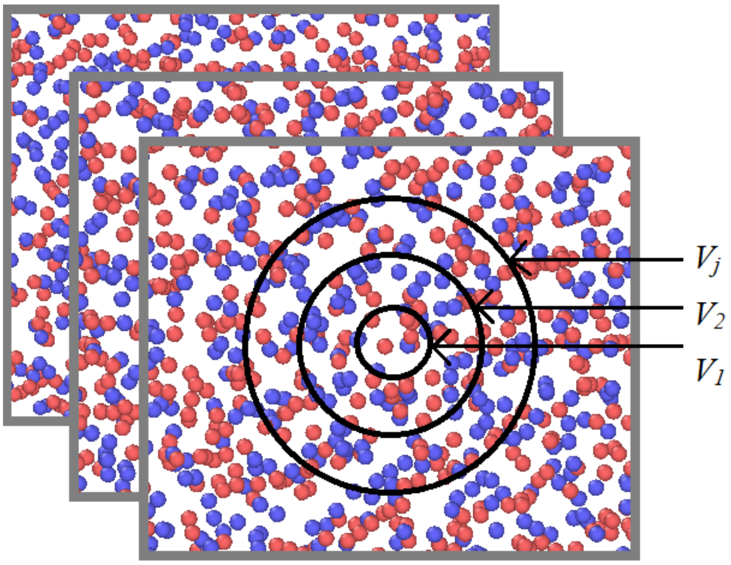

Figure 3.2: Illustration of the sampling of three configurations from the simulation trajectory and the procedure of sampling small spherical systems by varying the size. Simulation snapshots generated by Ovito version 2.9.0 [48].

The Small System Method (SSM), developed by Schnell *et al.* [25], exploits this connection to obtain macroscopic properties from energy- and density-fluctuations of small systems. The calculations are performed as MD simulations that place small non-periodic subsystems at random positions inside a larger periodic reservoir. By systematically increasing the volume of the subsystem, the wanted properties can be calculated as functions of size and shape of the small system. These properties scale linearly with the inverse cube root of the volume they were calculated in, which means that the macroscopic value can be found by extrapolation to the thermodynamic limit. The small system can only be regarded as in the grand canonical ensemble if the reservoir also is in the grand canonical ensemble, or if the small system is much smaller than the reservoir. For a closed system, a density change in the subvolume can not occur without a corresponding density change in the reservoir. This will affect the fluctuations of the largest of the embedded small systems, since the simulation box is not functioning properly as a grand canonical reservoir at this point, and deviations from linearity will occur. Strøm *et al.* [49] also concluded that the surface of the small system satisfies Gibbs thermodynamics for flat surfaces [50], which means that also the properties of the surface can be studied by Hill's thermodynamics for small systems [46].

There are two important things to consider when using SSM to calculate properties in the thermodynamic limit. The first condition is that the system is sufficiently far from its critical point, where fluctuations are very long ranged [44]. The second is that the total simulation box (reservoir) is large enough, such that the calculated properties for the different sized subvolumes display a definitive linear region as a function of inverse system size.

3.2.1 Calculating Properties

In the thermodynamics of small systems presented by Hill [46], $\mathcal{N}$ replicas of the small system were considered. Together, these replicas created an ensemble large enough to follow the laws of classical thermodynamics. The Gibbs equation for this ensemble is

$$\mathrm{d}U_\mathrm{t}^\mathrm{GC} = T\mathrm{d}S_\mathrm{t}^\mathrm{GC} - p^\mathrm{GC}\mathrm{d}V_\mathrm{t} + \sum_{i=1}^{n} \mu_i \mathrm{d}N_{i,\mathrm{t}}^\mathrm{GC} + X^\mathrm{GC}\mathrm{d}\mathcal{N} \tag{3.18}$$

where U_t is internal energy of the total system of $\mathcal{N}$ replicas, S_t is entropy of the total system of $\mathcal{N}$ replicas, V_t is volume of the total system, μ_i is chemical potential of component

i, and X^{GC} is given by

$$X^{\mathrm{GC}}(T,V,\mu_j) = \left(\frac{\partial U_{\mathrm{t}}^{\mathrm{GC}}}{\partial \mathcal{N}}\right)_{S_{\mathrm{t}},V,N_i} \equiv -\widehat{p}(T,V,\mu_j)V \tag{3.19}$$

which can be interpreted as the reversible work needed to add one replica of the small system at constant S_{t}, V and N_{t}. This means that S_{t} and N_{t} need to be redistributed over one more replica and that $V_{\mathrm{t}} = \mathcal{N}V$ increases. The superscript GC means grand canonical, and specifies that the variables should be calculated as a function of the total system's controlled variables T, V, μ_j and $\mathcal{N}$. The property $\widehat{p}$ is called the integral pressure and is only equal to the differential or normal pressure p in the thermodynamic limit [49]. When $\widehat{p}$ differs from p, the system can be considered as small [51]. By introducing ensemble averages of each property ($B_{\mathrm{t}}^{\mathrm{GC}}(T,V,\mu_j,\mathcal{N}) \equiv \mathcal{N}B^{\mathrm{GC}}(T,V,\mu_j)$) the average enthalpy of one replica can be written as $H^{\mathrm{GC}} \equiv U^{\mathrm{GC}} + p^{\mathrm{GC}}V$. However, the enthalpy of a *small system* with the controlled variables T, V and μ_j is defined by

$$\widehat{H}^{\mathrm{GC}} \equiv U^{\mathrm{GC}} + \widehat{p}V \tag{3.20}$$

By inserting the expressions for the ensemble averages of the variables into equation (3.18) and using the expression for the average internal energy of a single small system, the Gibbs-Duhem-like equation of a small system replica becomes

$$\mathrm{d}(\widehat{p}V) = S^{\mathrm{GC}}\mathrm{d}T + p^{\mathrm{GC}}\mathrm{d}V + \sum_{i=1}^{n} N_i^{\mathrm{GC}}\mathrm{d}\mu_i \tag{3.21}$$

and we get

$$\begin{aligned} S^{\mathrm{GC}} &= \left(\frac{\partial \widehat{p}V}{\partial T}\right)_{V,\mu_j} \\ p^{\mathrm{GC}} &= \left(\frac{\partial \widehat{p}V}{\partial V}\right)_{T,\mu_j} \\ N_i^{\mathrm{GC}} &= \left(\frac{\partial \widehat{p}V}{\partial \mu_i}\right)_{T,V,\mu_{j\neq i}} \end{aligned} \tag{3.22}$$

See references [49] or [51] for full derivation of the above equations. The superscript GC is from now on omitted. The equations (3.22) are particular for the small system, but are still linked to the probability distributions of statistical mechanics through Ξ, known as the grand canonical partition function [46]

$$\widehat{p}(T,V,\mu_j)V = k_\mathrm{B}T\ln\Xi(T,V,\mu_j) \tag{3.23}$$

As already mentioned, the second derivatives of the partition function are connected to fluctuations in particle- and energy-density. It is therefore possible to obtain fluctuation expressions for some central quantities, based on the derivatives of $\widehat{p}V$ (equations (3.22)).

For a two-component system, the fundamental properties of interest are the particle density $n_i \equiv \langle N_i\rangle/V$, the derivative of the internal energy density $u \equiv \langle U\rangle/V$

$$k_\mathrm{B}T\left(\frac{\partial u}{\partial\mu_i}\right)_{T,V,\mu_{j\neq i}} = \frac{\langle UN_i\rangle - \langle U\rangle\langle N_i\rangle}{V} \tag{3.24}$$

and a property defined by particle density fluctuations given by

$$\nu_{ik} \equiv \frac{k_\mathrm{B}T}{V}\left(\frac{\partial N_i}{\partial\mu_k}\right)_{T,V,\mu_{j\neq k}} = \frac{\langle N_iN_k\rangle - \langle N_i\rangle\langle N_k\rangle}{V} \tag{3.25}$$

These properties are central because they are linear combinations of N_i, U, $(\partial u/\partial\mu_i)_{T,V,\mu_{j\neq i}}$ and $(\partial N_i/\partial\mu_k)_{T,V,\mu_{j\neq k}}$ which are all extensive properties in sense of Hadwiger's theorem (explained in Klain [52]). This makes n_i, u and the properties defined in equations (3.24)-(3.25) extensive as well. According to Hadwiger [52], an extensive property can be expressed as a sum of four terms, each depending on the shape or size of the system. It was also shown by Blokhuis [53] that one extra term, proportional to the integral of the square of the total curvature, is needed. When only the value in the thermodynamic limit is of interest, higher order terms can be discarded, and only two terms remain. Consequently, extensive properties are expressed as a sum of those two contributions, one proportional to the volume and one to the surface area. This means that as a first approximation, the density of a general extensive property can be written as

$$b(T,V,\mu_j) = \frac{B(T,V,\mu_j)}{V} = b^\infty(T,\mu_j) + \frac{\Omega}{V}b^\mathrm{s}(T,\mu_j) \tag{3.26}$$

where b^∞ is the volume contribution and b^s is the surface contribution of b, while Ω is the surface area. Hence, we can write the extensive properties as

$$k_\mathrm{B}T\left(\frac{\partial u}{\partial \mu_i}\right)_{T,V,\mu_{j\neq i}} = k_\mathrm{B}T\left(\frac{\partial u}{\partial \mu_i}\right)^{\infty}_{T,V,\mu_{j\neq i}} + k_\mathrm{B}T\frac{\Omega}{V}\left(\frac{\partial u}{\partial \mu_i}\right)^{\mathrm{s}}_{T,V,\mu_{j\neq i}} \tag{3.27}$$

$$\nu_{ik} = \nu^{\infty}_{ik} + \frac{\Omega}{V}\nu^{\mathrm{s}}_{ik} \quad \text{and} \quad n_i = n^{\infty}_i + \frac{\Omega}{V}n^{\mathrm{s}}_i \tag{3.28}$$

Consequently, these are the properties that can be extrapolated to the thermodynamic limit using scaling laws. However, most properties of interest are not extensive in the sense of Hadwiger's theorem (explained in Klain [52]), and can therefore not be directly split into surface and volume terms. The solution is therefore to write these properties as combinations of the proper extensive properties (given in equations (3.27) and (3.28)) and find the thermodynamic limit values of the combinations. Using the equation for the enthalpy of a small system (3.20) and expressing the partial enthalpy as a combination of the extensive properties in equations (3.27)-(3.28) gives

$$\begin{aligned}\left(\frac{\partial\langle\widehat{H}\rangle}{\partial\langle N_i\rangle}\right)_{T,V,\mu_{j\neq i}} &= \frac{\langle UN_i\rangle - \langle U\rangle\langle N_i\rangle + \langle N\rangle k_\mathrm{B}T}{\langle N_i^2\rangle - \langle N_i\rangle^2} \\ &= k_\mathrm{B}T\frac{\left(\frac{\partial u}{\partial \mu_i}\right)^{\infty}_{T,V,\mu_{j\neq i}} + \frac{\Omega}{V}\left(\frac{\partial u}{\partial \mu_i}\right)^{\mathrm{s}}_{T,V,\mu_{j\neq i}} + n^{\infty} + \frac{\Omega}{V}n^{\mathrm{s}}}{\nu^{\infty} + \frac{\Omega}{V}\nu^{\mathrm{s}}}\end{aligned} \tag{3.29}$$

The same procedure can be applied for the KB integrals in the grand canonical ensemble

$$\begin{aligned}G^V_{ik} &= V\left(\frac{\langle N_iN_k\rangle - \langle N_i\rangle\langle N_k\rangle}{\langle N_i\rangle\langle N_k\rangle} - \frac{\delta_{ik}}{\langle N_i\rangle}\right) \\ &= \frac{\nu^{\infty}_{ik} + \frac{\Omega}{V}\nu^{s}_{ik}}{(n^{\infty}_i + \frac{\Omega}{V}n^{\mathrm{s}}_i)(n^{\infty}_k + \frac{\Omega}{V}n^{\mathrm{s}}_k)} - \frac{\delta_{ik}}{n^{\infty}_i + \frac{\Omega}{V}n^{\mathrm{s}}_i}\end{aligned} \tag{3.30}$$

In their work, Strøm *et al.* [49] investigated the effect of shape of the small systems on the thermodynamic properties of water. It was discovered that the deviations from linear behaviour become significant for the same volume of each shape (when the size of the small system approaches the size of the reservoir). It was therefore determined that these deviations were due to the finite size of the reservoir. To correct for this effect, a method for describing the finite reservoir contributions to the volume and surface terms was suggested. This method assumes that the effect only is a function of the ratio of subvolume to total simulation box. Hence, a general extensive property can be expressed as

$$b = b^{\infty} + \frac{\Omega}{V} b^{\mathrm{s}} + \frac{V}{V_0} b^{\mathrm{r}} \tag{3.31}$$

where b^{r} is the contribution from the reservoir effect, and V_0 is the reservoir volume, which is assumed to be of the same shape as the small systems. When the subvolume fraction is unity (the small system is of the same size as the reservoir), the fluctuations in equations (3.24)-(3.25) become zero. Hence, the general extensive property can be written as

$$b = \left[1 - \frac{V}{V_0}\right] b^{\infty} + \frac{\Omega}{V}\left[1 - \left(\frac{V}{V_0}\right)^{4/3}\right] b^{\mathrm{s}} \tag{3.32}$$

This equation is valid for all extensive properties in sense of Hadwiger's theorem (explained in Klain [52]) if they can be expressed by fluctuations that approach zero when $V/V_0 \to 1$. It should therefore provide volume and surface contributions of such properties, independent of the size of the reservoir. In other words, the underestimation of extensive properties, that can arise when the reservoir that is not large enough, can be corrected for.

At the other end of this scale, where the volumes are smaller than the the size of the particle, the behaviour depends on the instantaneous values of N_i. Since N_i only can be 0 or 1 for these volumes, their expected value for ν_{ii} in the limit is

$$\lim_{V\to 0} \nu_{ii} = \lim_{V\to 0} \frac{\langle N_i^2\rangle - \langle N_i\rangle^2}{V} = \frac{\langle N_i\rangle - \langle N_i\rangle^2}{V} \approx \frac{\langle N_i\rangle}{V} = n_i \tag{3.33}$$

The first equality is true because $N_i^2 = N_i$ when N_i either is equal to 1 or 0, and the second is true because $\langle N_i\rangle \gg \langle N_i\rangle^2$ for very small average values $\langle N_i\rangle$. The interaction between unlike particles should approach zero. This is because at least one of N_i and N_k always is zero when the volume is so small that it only can contain one particle. The result is that the product $N_i N_k$ always becomes zero, while the product $\langle N_i\rangle\langle N_k\rangle$ becomes so close to zero that it is negligible.

3.3 Integration of Radial Distribution Function

Recall that, in addition to using the fluctuation equations (2.19), the KB integrals can be obtained from integration of the RDF, using

$$G_{ik}^{V} = \frac{1}{V}\int_{V}\int_{V}\left[g_{ik}^{\infty}(r_{12}) - 1\right]\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_1 \tag{3.34}$$

where the double integral only can be reduced to a single integral in the limit $V \to \infty$. As mentioned in section 2.3.1, this results in a problem when attempting to calculate KB integrals for finite volumes. A proposed solution to this problem was presented by Krüger *et al.* [19] (explained in further detail in [54]) in their derivation of an exact expression for finite-volume KB integrals. This expression includes the purely geometrical function

$$w(r) \equiv \frac{1}{V}\int_{V}\int_{V}\delta(r - r_{12})\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 \tag{3.35}$$

where $\delta(r - r_{12})$ is the Dirac delta function. By combining $w(r)$ with equation (3.34), the double volume integrals are reduced to a single radial integral on the form

$$G_{ik}^{V} = \int\left[g_{ik}^{\infty}(r) - 1\right]w(r)\mathrm{d}r \tag{3.36}$$

The function $w(r)$ is characteristic for the volume V, and for hyperspheres of radius R in one, two and three dimensions, it can be calculated analytically. For a three-dimensional hypersphere the function is

$$w(r) = 4\pi r^2\left(1 - \frac{3r}{4R} + \frac{r^3}{16R^3}\right) \tag{3.37}$$

which gives the final expression for the KB integrals in a finite, three-dimensional system on the form

$$G_{ik}^{V} = 4\pi\int_{0}^{2R}\left[g_{ik}^{\infty}(r) - 1\right]r^2\left(1 - \frac{3r}{4R} + \frac{r^3}{16R^3}\right)\mathrm{d}r \tag{3.38}$$

It was also shown by Krüger *et al.* [19] that G_{ik}^{V} scales linearly with the inverse system size for systems of large R. The argument was based on considering the property

$$\begin{aligned} F &\equiv \int_V \int_{V_0 - V} \left[g_{ik}^{\infty}(r) - 1 \right] \mathbf{dr_1 dr_2} \\ &= \int_V \int_{V_0} \left[g_{ik}^{\infty}(r) - 1 \right] \mathbf{dr_1 dr_2} - \int_V \int_V \left[g_{ik}^{\infty}(r) - 1 \right] \mathbf{dr_1 dr_2} \end{aligned} \tag{3.39}$$

for a closed volume V, where V_0, as before, is the volume of the reservoir. For sufficiently large V, only particles in a layer of thickness equal to the correlation length of the system, on either side of the surface, Ω, contribute to F. This means that F scale as the size of the surface for increasing volumes. Krüger *et al.* [19] then assumed that the variable transform $\mathbf{r}_2 \rightarrow \mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$ could be applied for the first integral on the right-hand side of equation (3.39). By also recognizing that the second integral on the right-hand side of equation (3.39) is equal to the one in equation (3.34), we get $\Omega \propto F = VG_{ik} - VG_{ik}^V$. Since the surface to volume ratio of spheres is proportional to the inverse radius, the expression becomes $G_{ik}^V \propto 1/R + G_{ik}$.

Similarly to the SSM approach, the KB integrals can therefore be calculated for different sized volumes, plotted as a function of the inverse system size, and extrapolated to the thermodynamic limit. In other words, estimates of G_{ik} can be obtained by linear extrapolation of G_{ik}^V to $1/R \rightarrow 0$. The ability to calculate KB integrals for open systems while simulating closed ones makes the method very useful when use of open systems are prohibited, as it is in MD simulations. The procedure of the linear extrapolation, and effects of the finite size of the system is further discussed in [54]. Dawass *et al.* [55] has also recently shown that the method is not constrained to hyperspheres, but that the same procedure can be applied to calculate KB integrals from subvolumes of complex, arbitrary shapes.

3.3.1 Finite-Size Correction of Radial Distribution Function

When using the RDF to calculate KB integrals in finite and closed systems, there are two finite-size effects to consider. The first arises from the finite, and often too small, size of the domain used to compute the KB integrals, and can be overcome by the method proposed by Krüger *et al.* [19]. The second effect is due to the finite number of particles inside a closed system, which will affect the calculation of the RDF.

As mentioned in section 2.3.1, the RDFs calculated in closed and finite systems differ from those calculated in open and infinite systems. The difference is in a term of order N^{-1}, that arises from the finite number of particles in a closed system [56]. For large systems, consisting of many particles, this difference is small, but it does become important when integrating over the entire volume, as in equation (3.38). These discrepancies arise from the density calculations performed around each particle in the system during the RDF calculations. Fixing one particle at a specific point affects the density at any other point in the system, and reduces it to $\rho = (N-1)/V$ [4]. For a two-component system of species i and k, the effect will depend on which component is fixed. Placing a particle of type i at a fixed position changes the average number of i particles in the total system by exactly -1, but it does not change the total number of particles of type k. For both pure fluids and mixtures, the symmetry is not respected, and the average density will differ from the bulk density, except in the limit $V \rightarrow \infty$ [19].

Since both the original expression for the KB integrals (2.18), and the Krüger equation (3.38) require RDFs of open and infinite systems, this often becomes a problem when performing MD simulations. Several methods are available to account for this, and a comparison of the efficiency of three of these was performed by Dawass *et al.* [54]. They concluded that a method proposed by Ganguly *et al.* [8] yields the most accurate results. A similar analysis was performed by Milzetti *et al.* [22], which resulted in the same conclusion. This method, from now on referred to as the Ganguly correction, takes into account the effect on the density that arises from fixing one particle at a specific position. The idea is that the excess or depletion of particle type k around a particle type i at local scale is compensated by the excess or depletion at long distances, having that the total number of particles is constant. The correction includes calculating the excess or depletion of the number of molecules of type k around a central molecule of type i, as a function of the distance r. The excess or depletion is calculated inside spherical volumes $V(r)$ where the radius r is drawn from the central molecule of type i [54], and can be expressed as

$$\Delta N_{ik}(r) = \int_0^r \mathrm{d}r' 4\pi r'^2 \rho_k [g_{ik}(r) - 1] \tag{3.40}$$

where ρ_k is the bulk density of molecule type k. Assuming that the total simulation box is cubic, with box length L, makes the expression valid for $r < L/2$. Longer distances are not necessary, since it was concluded by Dawass *et al.* [54] that r should not be extended beyond half the length of the simulation box. The estimated RDF, according to the Ganguly correction is presented as

$$g_{ik}^{\text{Ganguly}}(r) = g_{ik}(r)\frac{N_k\left(1-\frac{V(r)}{V_0}\right)}{N_k\left(1-\frac{V(r)}{V_0}\right)-\Delta N_{ik}(r)-\delta_{ik}} \tag{3.41}$$

It is clear from the above equation that for infinitely large systems, $g_{ik}^{\text{Ganguly}}(r)$ is equal to $g_{ik}(r)$.

3.4 Cortes-Huerto *et al.* [20] Scaling Equation

Cortes-Huerto *et al.* [20] proposed yet another method of obtaining the KB integrals in the thermodynamic limit, that also utilizes small subsystems embedded in a MD simulation box. The fluctuations are calculated in the same manner as for SSM, but it applies a different technique to obtain the KB integrals in the thermodynamic limit. SSM is solely based on the thermodynamics of small systems presented by Hill [46], and can be used to obtain macroscopic values of all properties available from fluctuations. Cortes-Huerto *et al.* [20] instead modified equation (2.19) for the purpose of calculating KB integrals in the thermodynamic limit from the ones calculated for finite volumes of varying size. Since the scaling equation is derived from the integral on left-hand side of (2.19), it is still connected to the fluctuations of number of particles.

Cortes-Huerto *et al.* [20] stated that the motivation behind the derivation of this expression is that it explicitly takes the finite size of the simulation box into account. The reasoning is that the fluctuations are dependent on both the volume of the simulation box, and the subvolume they are calculated in. The derivation starts off by defining an expression, analogous to equation (2.19), that is valid for closed and finite reservoirs

$$\begin{aligned} G_{ik}^{V,V_0} &= \frac{1}{V}\int_V\int_V \left[g_{ik}(r_{12})-1\right]\mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_1 \\ &= V\left(\frac{\langle N_i N_k\rangle' - \langle N_i\rangle'\langle N_k\rangle'}{\langle N_i\rangle'\langle N_k\rangle'} - \frac{\delta_{ik}}{\langle N_i\rangle'}\right) \end{aligned} \tag{3.42}$$

where V_0 is the volume of the total simulation box. The average number of particles of type i is defined as $\langle N_i\rangle' \equiv \langle N_i\rangle_{V,V_0}$. The RDF is also explicitly expressed as the one that is obtained from a closed system. To account for this, a finite size correction is suggested on the form

$$g_{ik}(r) = g_{ik}^{\infty}(r) - \frac{1}{V_0}\left(\frac{\delta_{ik}}{\rho_i} + G_{ik}\right) \tag{3.43}$$

This expression is based on the work by Ben-Naim [39], which describes the differences between the RDFs obtained in open, and closed systems. In his work, it is also stated that the effect of a closed system only has an influence on the RDF for distances $r > \xi$, and that this becomes important when the integration extends to $r \to \infty$. It is named the closure correlation, which in the asymptotic limit takes the form

$$g_{ik}(r \gg \xi) = 1 - \frac{1}{V_0}\left(\frac{\delta_{ik}}{\rho_i} + G_{ik}\right) \tag{3.44}$$

Combining this with the asymptotic behaviour of the RDF of an open system ($g_{ik}^{\infty}(r \gg \xi) = 1$), gives the proposed expression in equation (3.43). Since this correction is derived based on the asymptotic behaviour of the RDFs, it strictly valid for $r \to \infty$. Hence, applying it for the entire range of distances means that the difference between $g_{ik}(r)$ and $g_{ik}^{\infty}(r)$ is independent of r. On that point, this method differs from the Ganguly correction [8] where the excess or depletion is considered for all distances r (see section 3.3.1).

Similarly to the work of Krüger *et al.* [19], the effect of the finite size of the subvolume is also considered. As explained in section 3.3 the transformation $\mathbf{r}_2 \to \mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$ can not be applied to equation (3.42) when V is finite. In order to solve the integral, it is therefore split into the domains $\int_V \int_{V_0}$ and $\int_V \int_{V_0 - V}$. The latter is solved by considering the same arguments as used by Krüger *et al.* [57]. That is, the particles in the region $V_0 - V$ will have a contribution that is proportional the surface area of the subvolume [58]. This integral can therefore be written as $\Omega C'_{ik}$, where C'_{ik} is a proportionality constant that depends on the thermodynamic state of the system. This, together with the expression for the correction of the RDF in equation (3.43), turns equation (3.42) into

$$G_{ik}^{V,V_0} = \frac{1}{V}\int_V \int_{V_0} \left[g_{ik}^{\infty}(r_{12}) - 1\right] \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 - \frac{V}{V_0}\left(\frac{\delta_{ik}}{\rho_i} + G_{ik}\right) + \frac{C_{ik}}{V^{1/3}} \tag{3.45}$$

where $\Omega / V \propto 1/V^{1/3}$ has been used for convenience, turning the proportionality constant C'_{ik} into C_{ik}. To solve the remaining integral, Cortes-Huerto *et al.* [20] applied the restriction $V_\xi < V < V_0$ where $V_\xi = 4\pi\xi^3/3$. As a result, finite size effects due to oscillations in RDF beyond V are neglected. Additionally, the PBCs are assumed to make the entire system translationally invariant, which means that the variable transform $\mathbf{r}_2 \rightarrow \mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$, can now be applied. Hence, the integral becomes the original expression in equation (2.18), yielding the KB integral in the thermodynamic limit G_{ik}. The final expression therefore becomes

$$G_{ik}^{V,V_0} = G_{ik}\left(1 - \frac{V}{V_0}\right) - \frac{V}{V_0}\frac{\delta_{ik}}{\rho_i} + \frac{C_{ik}}{V^{1/3}} \tag{3.46}$$

A more convenient version is obtained by introducing $\lambda \equiv (V/V_0)^{1/3}$

$$\lambda G_{ik}^{V,V_0}(\lambda) = \lambda G_{ik}(1 - \lambda^3) - \lambda^4 \frac{\delta_{ik}}{\rho_i} + \frac{C_{ik}}{V_0^{1/3}} \tag{3.47}$$

When $\lambda G_{ik}^{V,V_0}(\lambda)$ is plotted as a function if λ, a linear region will appear for $\lambda^3 \approx 0$, where the slope can be identified as the KB integral in the thermodynamic limit. Cortes-Huerto *et al.* [20] identify this region as $\lambda \leq 0.3$. As earlier mentioned, another constraint has already been applied in order to solve the integrals. That is, the volume of the subdomain has to be larger than V_ξ, resulting in a total constraint of

$$\left(\frac{4\pi}{3V_0}\right)^{1/3} \xi \leq \lambda \leq 0.3 \tag{3.48}$$

This should be taken into consideration when choosing size of subvolumes and when identifying the linear region.

3.4.1 Lattice Approach by Galata *et al.* [23]

An even newer contribution to the field of computation of KB integrals is the work by Galata *et al.* [23], which applied the scaling equation (3.47) by Cortes-Huerto *et al.* [20] to compute the KB integrals in the thermodynamic limit. However, a different definition of subvolumes was suggested. Instead of the randomly positioned subvolumes used by Cortes-Huerto *et al.* [20], which also is the standard method used in SSM, a three dimensional grid was superimposed on the cubic simulation box. In that way, the entire simulation box was partitioned into N_{cells} smaller cell that were considered to be in the grand canonical ensemble. The computation of the finite volume KB integrals from the fluctuations also differed from previous methods [49] [8] [20] [21]. Normally, the values of $\langle N_i \rangle$ and $\langle N_i N_k \rangle$ are computed as averages over subvolumes and snapshots, and later combined by the right-hand side of equation (2.19) to compute the KB integrals. Instead, Galata *et al.* [23] computed the fluctuations inside each cell, before computing the KB integrals directly from these. The average KB integrals for the current partitioning, λ, were then obtained by computing the average over all cells

$$G_{ik}^{V}(\lambda) = \frac{1}{N_{\text{cells}}} \sum_{j=1}^{N_{\text{cells}}} G_{ik,j}^{V}(\lambda) \tag{3.49}$$

By varying the size of the partitioning λ, the KB integrals for different sized subvolumes $G_{ik}^{V}(\lambda)$ could be calculated, and the KB integral in the thermodynamic limit were be obtained using equation (3.47).

There are two potential errors using this approach. The first is the use of a superimposed grid to define the subvolumes. For subsystems of macroscopic size, the fluctuations will in general not be correlated. However, if the subvolumes are of infinitesimal size, their fluctuations will be highly correlated if the separation of the subsystems is of the same size as the range of the interparticle forces [32]. The fact that the lattice cells at all times are placed right next to each other makes their fluctuations completely correlated, meaning that if a particle leaves one of the sub-boxes it immediately enters another. The result is that the lattice cells do not provide independent samples. The second problem is the computation of the KB integrals using equation (3.49). The brackets in equation (3.30) denote ensemble averages of the system. Computing $G_{ik,j}^{V}(\lambda)$ directly for each subvolumes means that the fluctuations are not proper ensemble averages, but rather time averages for each cell. Computing the KB integrals by averaging over all cells later will not correct for this, since the conditions of the right-hand side of equation (2.19) are not followed.

4 Computational Procedure

The following section presents the details of the simulations performed for the different mixtures. The procedures used for computation of the KB integrals are also described. This includes the integral of the RDF, and fluctuations calculated from randomly positioned subvolumes, as used by SSM, and lattice partitioning, as suggested by Galata *et al.* [23]. The programming language Python version 3.5.2 [59] was used for postprocessing of results, and for calculation of fluctuations by random positioned subvolumes for the LJ mixtures. For the methanol-water mixture, both subvolume methods were implemented using Python. LAMMPS input scripts used for simulations are found in appendix D, while Python scripts are found in appendix C.

4.1 Simulation Details

The MD simulations were performed using the open source code LAMMPS (version 11 August 2017) [60]-[61], which applies the Verlet algorithm and the rRESPA integrator derived by Tuckerman *et al.* [62] for integration of the equations of motion (see section 3.1.4). The simulation box was in all cases cubic and had periodic boundary conditions in all directions, to make sure that the system was studied in the bulk. Thermostatting and barostatting were performed using the Nosé-Hoover style equations of motion by Shinoda *et al.* [63]. These equations are based on the scheme of Martyna *et al.* [64], which are known to produce the desired distributions of position and velocity [65]. In other words, they generate positions and velocities sampled from the NVT or NpT ensemble.

Three different types of binary mixtures were studied. The first two were an ideal LJ mixture and a real LJ mixture. The parameters describing these mixtures and all settings used during the simulations were chosen to reproduce the work by Galata *et al.* [23] in order to make sure that their method was implemented correctly. The simulation details for the two LJ mixtures are identical, and are therefore presented together in section 4.1.1. The third mixture was of more complex character, consisting of methanol and water molecules, and is presented in section 4.1.2. For all mixtures, the tail correction implemented in LAMMPS was applied. This correction is based on the work by Sun [66], and makes it possible to analytically compute the contributions of a long-range van der Waals interaction to the total energy and pressure. The long-range tail corrections (LTC) to the total energy and pressure for a system of n different non-bonded atom types are given by

$$E_{\mathrm{LTC}} = \frac{1}{2}\sum_{i=1}^{n} N_i \sum_{k=1}^{n} \rho_k 4\pi \int_{r_c}^{\infty} g_{ik}(r) v_{2,ik}(r) r^2 \mathrm{d}r \tag{4.1}$$

$$p_{\mathrm{LTC}} = \frac{1}{6}\sum_{i=1}^{n} \rho_i \sum_{k=1}^{n} \rho_k 4\pi \int_{r_c}^{\infty} g_{ik}(r) \left(r \frac{\mathrm{d}v_{2,ik}(r)}{\mathrm{d}r} \right) r^2 \mathrm{d}r \tag{4.2}$$

where i and k represent the different components in a mixture, ρ_i is the density of component i, and $v_{2,ik}(r)$ is the pair potential between particles of type i and k. When this is applied during the simulations, the atoms do not feel any LJ pair interactions beyond the cutoff, but the calculated energy and pressure include an estimated contribution from those interactions [67]. The thermophysical properties are therefore not thermodynamically consistent with the truncated LJ force field. There are two conditions that need to be fulfilled in order to apply this tail correction. The first is that $g_{ik}(r)$ must be unity beyond cutoff, r_c, such that the integrals in equations 4.1-4.2 can be evaluated analytically. The second is that the system is a homogeneous liquid. For the methanol-water mixture, long-range tail correction is only applied to the energy and pressure for the Lennard-Jones portion of the pair interactions.

4.1.1 Ideal and Real Lennard Jones Mixtures

Reduced units were used for both LJ mixtures since this made it possible to study the problems at a more general level, in addition to making it more easily comparable to other studies of LJ mixtures. Definitions are found in list of reduced units. The interactions of like and unlike particles were modelled by the LJ potential with a cut-off set to $r_c = 3\sigma_{11}$. The LJ parameters for both mixtures are found in table 4.1, where the interaction between unlike particles were obtained using Lorentz-Berthelot combining rules [68]-[69]

$$\sigma_{12} = \frac{\sigma_{11} + \sigma_{22}}{2} \quad , \quad \varepsilon_{12} = \sqrt{\varepsilon_{11}\varepsilon_{22}} \tag{4.3}$$

Table 4.1: LJ parameters used for simulations of ideal and real LJ mixtures. Values were chosen in order to reproduce the work by Galata *et al.* [23].

Mixture	ε_{11}	ε_{22}	ε_{12}	σ_{11}	σ_{22}	σ_{12}
Ideal	1	1	1	1	1	1
Real	1	0.5968	0.7726	1	0.7677	0.8838

Both systems consisted of 20,000 particles. The production runs were performed in the canonical ensemble (NVT) with a reduced temperature of $T = 0.93$, and in the isobaric-isothermal ensemble (NpT), with the same temperature and a reduced pressure of $p = 0.47$. All initial configurations were created by starting off with placing 20,000 particles of type 1 at random positions inside a large simulation box ($V = 500^3$), in order to avoid overlap between particles. Applying the thermostat and barostat with the required temperature and pressure, eventually compressed the box to its equilibrium volume. In order to make sure that the simulation remained stable, the time step was set to $\Delta t = 0.002$, and the relaxation times of the temperature and pressure were set to $100 \cdot \Delta t = 0.2$ and $1000 \cdot \Delta t = 2$ respectively. The total number of time steps for the initialization was $10 \cdot 10^6$. By manually changing the identity of the particles from type 1 to type 2 in the end configuration file, the five desired compositions were obtained. The different compositions investigated were $x_1 = 0.05, 0.25, 0.50, 0.75, 0.95$. These were then equilibrated for an additional $10 \cdot 10^6$ time steps before starting the production runs. During the equilibriation and production runs the time step was set to $\Delta t = 0.0036$, and the relaxation times of the temperature and pressure were set to $100 \cdot \Delta t = 0.36$ and $1000 \cdot \Delta t = 3.6$ respectively. Different values were tested for the relaxation times, in order to determine which gave the least fluctuations around the mean value. All production runs consisted of a total of $50 \cdot 10^6$ time steps, divided into five parallels for computation of standard deviations.

4.1.2 Methanol-Water Mixture

The force field developed by Weerasinghe and Smith [9] was chosen to represent methanol in the mixture. This model was specifically designed to reproduce the experimental KB integrals in aqueous methanol mixtures, as a function of the methanol mole fraction. The total molecule is nonpolarizable, and the methyl group (CH_3) is modelled as one, spherical particle, with parameters taken from reference [70]. The methanol molecules are therefore rigid triangles with fixed bond lengths and angles, where the charges are situated on each particle. This methanol force field was designed to be used with the SPC/E water model [71] [72], which therefore was chosen to represent water in this work. The water molecules modelled by the SPC/E water model are also rigid triangles with charges situated on each atom, while the LJ sites are situated on the oxygen atoms. Bond lengths and angles are presented in table 4.2 for both molecule types.

Table 4.2: Angles and bond lengths for the rigid methanol molecules [9] and SPC/E water molecules [72], modelled by the KB force field.

Force field	Bond length [Å]	Angle [degree]
Water		
SPC/E	O-H = 1.000	H-O-H = 109.47
Methanol		
KBFF	CH_3-O = 1.430 O-H = 0.945	CH_3-O-H = 108.49

The interactions between the molecules in the mixture were described by a pair potential combining the LJ potential and the electrostatic interactions. The cut-off was set to 15 Å, meaning that the particles separated by a distance longer than 15 Å experience zero interaction from the LJ potential. The electrostatic forces were calculated using Ewald summation [45], where the pairwise interactions within the cut-off distance were computed directly, while interactions outside the cut-off distance were computed in reciprocal space (see section 3.1.3 for further explanation). The force field parameters for like particles of both molecules are given in table 4.3, while parameters for unlike particles were calculated using geometric mean given by

$$\sigma_{ik} = \sqrt{\sigma_{ii}\sigma_{kk}} \quad , \quad \varepsilon_{ik} = \sqrt{\varepsilon_{ii}\varepsilon_{kk}} \tag{4.4}$$

Table 4.3: Force field parameters for the methanol molecules [9] and SPC/E water molecules [72] modelled by the KB force field

Force field	Atom	ε [kcal/mol]	σ [Å]	q [C]	m [g]
Water					
SPC/E	O	0.1555	3.166	-0.8476	15.9994
	H	0	0	0.4238	1.008
Methanol					
KBFF	CH_3	0.2073	3.748	0.30	15.0347
	O	0.1555	3.192	-0.82	15.9994
	H	0.0210	1.580	0.52	1.008

The production runs were performed in the canonical ensemble (NVT) with the temperature set to T = 300 K. Three different compositions were investigated, x_{MeOH} = 0.30, 0.50, 0.70. All initial configurations were created by placing the a total of 500 molecules at random positions inside a large simulation box ($V = 1000^3$ Å^3), in order to avoid overlap between particles. The simulation box was then compressed to its equilibrium size by applying the NpT barostat with the required pressure of 1 atm. The simulation box was then replicated once in each direction in space, giving a total of 8,000 molecules. This was followed by 10^6 equilibriation steps before starting the production runs. All production runs consisted of a total of $50 \cdot 10^6$ time steps, divided into five parallels for computation of standard deviations. The time step was for all simulations set to $\Delta t = 1$ fs, while the relaxation time for temperature was set to $100 \cdot \Delta t = 100$ fs. The relaxation times for pressure during initialization was set to $1000 \cdot \Delta t = 1000$ fs.

4.2 Radial Distribution Function

The RDF is calculated by determining the total number of particles within a range of distances r from one central particle [38]. For the LJ mixtures this calculation was performed for all particles in the system, each time centered on the current particle. The range of distances represented spherical shells around the central particle, or rather bins in a histogram, in which all the neighbouring particles were sorted. This was performed for the three different types of interactions present in the mixture, that is, the 1-1, 2-2 and 1-2 interactions. For the methanol-water mixture, the central atom in both molecule types was considered to be oxygen, and the RDF was therefore computed between the oxygen atoms in each molecule, representing the three different interactions MeOH-MeOH, water-water and MeOH-water. The bins were of uniform size in radial distance, and the values of $g_{ik}(r)$ for each bin were then calculated by scaling it by the number of particles it would contain if the particles of type k were uniformly distributed [73]. This was performed using the built in compute `rdf` function in LAMMPS with the `rerun` command, which reads the configuration files from the original simulation, and performs the desired computations. Equation (3.41) was used to correct for finite size effects, and the KB integrals were calculated using equation (3.38) both from the original RDF and after applying Ganguly finite size correction.

For the LJ mixtures, two different sampling rates were used for the computation of the RDF. The system's configuration was sampled every 100 and every 10,000 time step, creating a total of 500,000 and 5,000 snapshots respectively. The radial distance from the central particle ranged from 0 to $L_0/2$, where L_0 is the minimum size of the simulation box over the course of the NpT simulation. For NVT simulations the volume was constant, and $L_0/2$ was therefore be equal to half the size of the simulation box at all times. To investigate how the computation of the RDF depend on the bin size, or rather the number of bins, three different total number of bins were investigated, 150, 1,000 and 2,000.

For the methanol-water mixture, the system's configuration was sampled every 100 step, creating a total of 500,000 snapshots. The radial distance from the central particle ranged from 0 to half the size of the simulation box, divided into 3,000 bins.

4.3 Small System Method

For the SSM calculations of all mixtures, the system's configuration was stored from the trajectory every 100 time step. For each configuration, 10 random positioned points inside the simulation box were used to position the center of the small systems, giving a total of 5×10^6 samples for each small system volume V. The small systems were spherical and centered at randomly chosen points, $p_\mathrm{c} = (x_\mathrm{c}, y_\mathrm{c}, z_\mathrm{c})$. All particles with position $p_\mathrm{p} = (x_\mathrm{p}, y_\mathrm{p}, z_\mathrm{p})$ satisfying $(x_\mathrm{p} - x_\mathrm{c})^2 + (y_\mathrm{p} - y_\mathrm{c})^2 + (z_\mathrm{p} - z_\mathrm{c})^2 \leq R^2$ were placed inside the sphere of radius R. For the LJ systems, 100 different sized small systems were investigated. These had a radius R increasing linearly from the size of a particle to little under half the minimum size of the simulation box over the course of the NpT simulations ($R_\mathrm{max} = 14$ for the ideal LJ mixture and $R_\mathrm{max} = 13$ for the real LJ mixture). For the methanol-water mixture, each molecule's center of mass was regarded as its position, and 200 different sized small systems were investigated, with a radius increasing linearly from 1 Å to 20 Å.

4.4 Lattice Partitioning

For every sampled configuration, eight different lattices of eight different grid sizes were superimposed on the minimum volume of the simulation box over the course of the NpT simulation (for the NVT simulations, the total simulation box was used). This gave eight different values of the scaling parameter $\lambda \approx 1$, 0.5, 0.333, 0.25, 0.2, 0.1667, 0.143, 0.125, which produced unequal amount of samples for subvolumes of different size. From one simulation snapshot, the number of samples are 1, 8, 27, 64, 125, 216, 343 and 512 for the different sized subvolumes respectively. The values of $\langle N_i \rangle$, $\langle N_i N_k \rangle$ where calculated as averages over subvolumes of the same size. The averaging method proposed by Galata *et al.* [23] was also tested. This method included calculating the fluctuations in each lattice cell, before calculating the values of $G^V_{ik,j}(\lambda)$ in each cell, followed by averaging over all cells to obtain $G^V_{ik}(\lambda)$ (see section 3.4.1 for further explanation). For the LJ mixtures, the instantaneous number of particles in each lattice cell was calculated by LAMMPS during the simulations. For the methanol-water mixture, each molecule's position was given by its center of mass.

5 Results and Discussion

The following sections present the KB integrals obtained by using the different methods presented in section 3. These methods can be sorted into two main categories: 1) integration of the RDF and 2) calculation of fluctuations in number of particles. Different variations within these methods are thoroughly analyzed, in order to determine how they affected the final value of the KB integral in the thermodynamic limit. The results of the LJ mixtures are first presented, followed by a comparison to theoretical values when these are available. Lastly, the results of the methanol-water mixture are presented and analyzed.

5.1 Lennard-Jones Mixtures - Radial Distribution Function

The method proposed by Krüger *et al.* [19] was used to calculate the KB integrals for finite volumes by integration of RDF. The finite-sized KB integrals were calculated from equation (3.38) for volumes of radius $2R$, where the integral was solved numerically using the trapezoidal method. These volumes represented the shells used for computation of the RDFs, and ranged up to half the size of the simulation box. Equation (3.38) states that, for an integration limit of $2R$, the KB integrals are calculated for volumes of radius R. Consequently, the largest volume the KB integrals were calculated for, was one with diameter equal to half the length of the total simulation box. Plotting the values of G_{ik}^{V} as functions of the inverse systems size $1/R$ display the expected linear behaviour in the region $1/R \rightarrow 0$, as illustrated in figure 5.1 for the ideal LJ mixture at $x_1 = 0.75$. The values of G_{ik} were then determined by extrapolation to the thermodynamic limit, using linear curve fitting with the `polyfit` function in MATLAB [74].

In order to apply this method, the KB integrals of finite volumes first need to be converged. The condition for convergence is that values calculated for the larger volumes should approach one value as a function of system size. After $4 \cdot 10^6$ time steps this was observed for both components at equimolar composition ($x_1 = 0.50$), while only the excess components reached convergence for other compositions. After $10 \cdot 10^6$ time steps, convergence was reached for all compositions of both LJ mixtures.

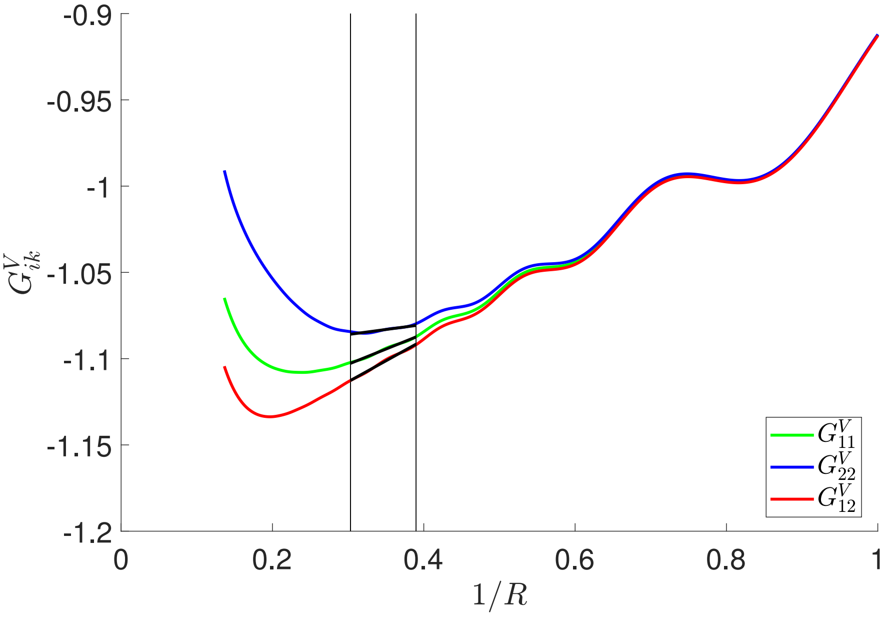

Figure 5.1: Finite size KB integrals for the ideal mixture at $x_1 = 0.75$ as a function of inverse system size, calculated using equation (3.38). The value of the KB integrals in the thermodynamic limit, G_{ik}, was found by linear curve fitting to the region marked by vertical lines.

Besides from the effect of simulation time, four different factors, and their influence on the value of the KB integral in the thermodynamic limit were investigated. These were:

1. Which ensemble was used during the simulation run.
   - NpT ensemble.
   - NVT ensemble.
2. The total number of bins (shells) used for computation of RDF.
   - 150 bins.
   - 1,000 bins.
   - 2,000 bins.
3. How often the simulation trajectory was sampled.
   - Every 10,000 time step.
   - Every 100 time step.
4. Applying the Ganguly finite size correction to the RDF.

The results of the KB integrals in the thermodynamic limit obtained from simulation in the NpT ensemble are presented in this section, while the equivalent figures of the simulation in the NVT ensemble are found in appendix B. First, the behaviour of the RDF is discussed.

The RDF showed unexpected behaviour at large distances. As stated in section 2.3.1, and explained in section 3.3.1, the RDF of both ideal and non-ideal, open systems should approach unity, while it should approach $(1-1/N)$ for closed systems. For the systems studied here, some of the RDFs instead approached a value a little higher than unity, stabilizing at ~ 1.00004. It was first believed to be caused by density changes arising from the fluctuations of the total volume of the simulation box in the NpT ensemble, but the same behaviour was also observed in the NVT ensemble. The fluctuating volume could therefore not be causing this behaviour.

Technically, $g_{ik}(r)$ displaying a value little higher than unity is interpreted as a system with a little higher probability of finding a particle of type k a distance r from a particle of type i, compared to an uniform distribution of component i. This is not believed to be the case for the systems studied here, since the densities of both components were observed to be uniform. Also, since this was observed for both LJ mixtures at all compositions, and in both ensembles, it is more likely that something methodical caused this behaviour. It is therefore believed to be a result of how the RDF was calculated by LAMMPS. One possible explanation could be long-range tail corrections that were applied during the simulation. Applying this correction assumes that the RDF is unity beyond cut-off. Since the cut-off was $r_c = 3\sigma$, and the RDF curves were observed to still fluctuate for r larger than 3σ, this condition did not hold for the LJ systems (see figures 5.4-5.7 in the following sections). This could have changed the actual dynamics of the system for the NpT simulation, since it is designed to correct the values of pressure and energy [67]. However, since the same behaviour also was observed for the NVT simulations, it is not likely that the tail corrections caused the discrepancies. To further validate this, the simulations for ideal LJ mixture at $x_1 = 0.50$ was repeated without tail corrections. This resulted in the same behaviour as explained above, thereby confirming that it was not caused by the tail corrections. The actual cause could not be further determined, but it is believed to be connected to the normalization or counting of particles performed by the RDF calculation, or other settings applied during the simulations.

The results obtained using different total number of bins is presented for the ideal and real LJ mixtures in figures 5.2-5.3 respectively, and further discussed in section 5.1.1. The effect of sampling rate is discussed in section 5.1.2, while the effect of applying the Ganguly finite size correction is discussed in section 5.1.3. First, the trends as functions of mole fraction are analyzed.

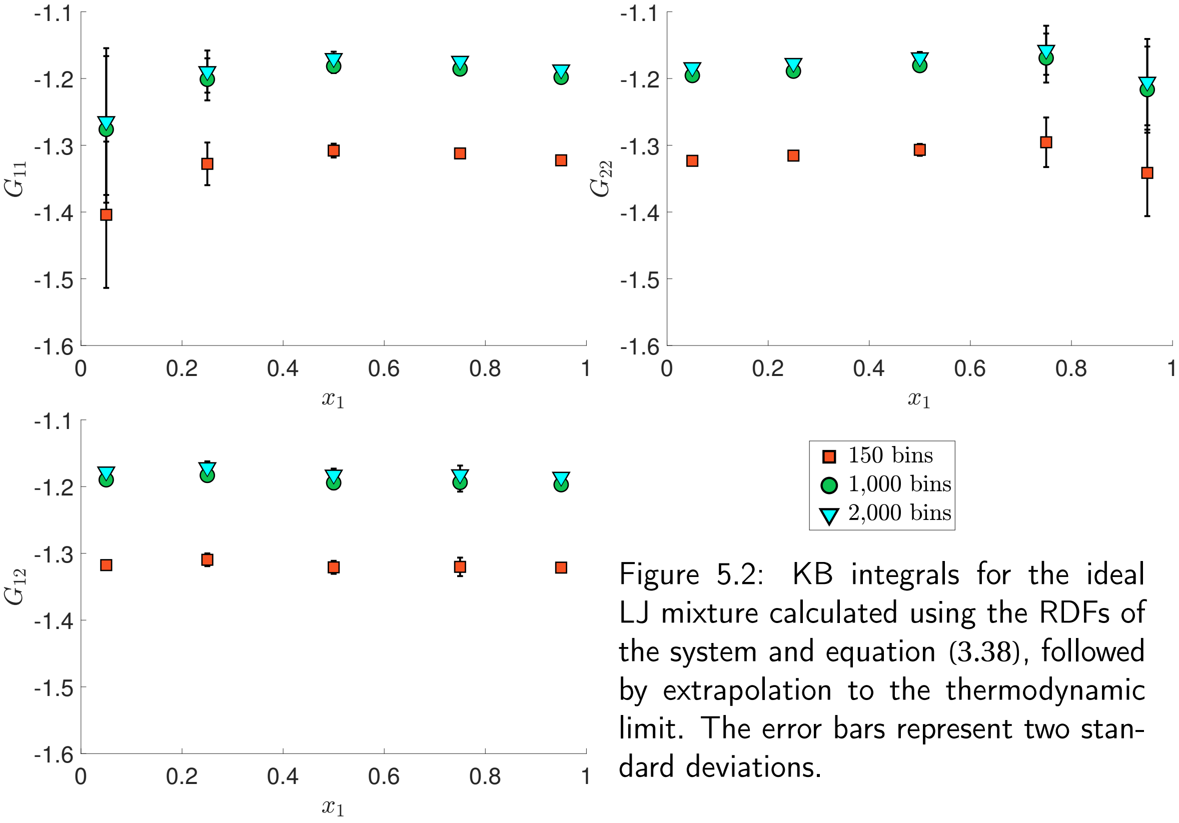

Figure 5.2: KB integrals for the ideal LJ mixture calculated using the RDFs of the system and equation (3.38), followed by extrapolation to the thermodynamic limit. The error bars represent two standard deviations.

As expected for an ideal mixture [75], the KB integrals for like components (G_{11} and G_{22}) show symmetric trends as functions of mole fraction. They both also show better statistics for compositions with higher mole fractions of their respective component. In contrast, the KB integrals for the 1-2 interaction vary less with mole fraction, and retain approximately the same precision at all compositions. Both of these effects were most likely caused by the precision of the RDF. For system with low mole fractions of component i ($x_i = 0.25$ or $x_i = 0.05$), the RDF of like particles of this type, $g_{ii}(r)$, was computed for fewer pairs than the RDF for the excess component, $g_{kk}(r)$. This resulted in a poorer average for component i, which in turn gave larger standard deviations for the KB integrals. As stated above, the finite-volume KB integrals of these compositions also required longer simulation time to reach convergence for the the component of lowest mole fraction. The trend is different for the 1-2 interactions because the number of pairs of unlike particles remained large enough to provide good average values, with low standard deviations, also in the outer ranges of the mole fraction scale.

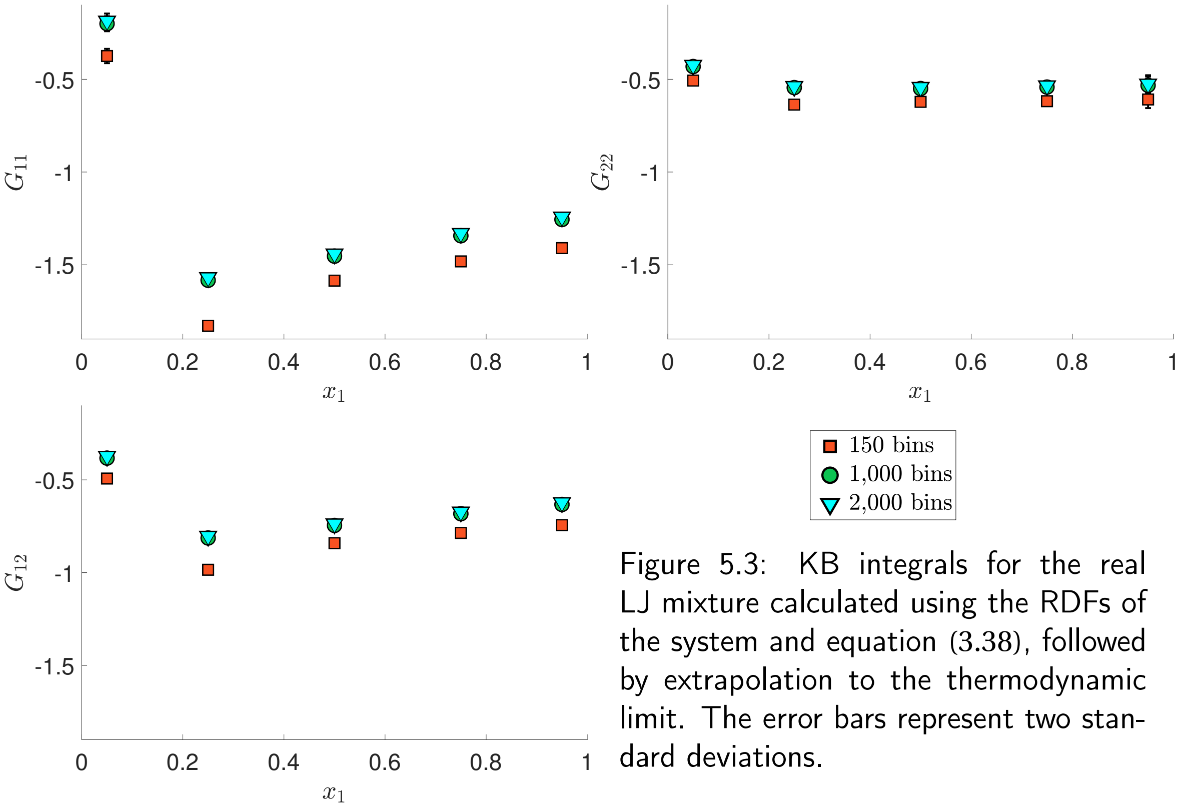

Figure 5.3: KB integrals for the real LJ mixture calculated using the RDFs of the system and equation (3.38), followed by extrapolation to the thermodynamic limit. The error bars represent two standard deviations.

The KB integrals for real LJ mixture do not show symmetric behaviour, and neither were they expected to. However, they did show the same trend as the ideal mixture in magnitude of standard deviations. Note that the ranges are different for the two figures 5.2-5.3, and that the standard deviations actually are of approximately the same magnitude for the two mixtures. This was expected, since the principles for average values described previously, also apply for the real LJ system. The linear region for both LJ mixtures was somewhere between $1/R \approx 0.3$ and $1/R \approx 0.5$ for all compositions.

5.1.1 Effect of Total Number of Bins

The RDFs were calculated using three different total number of bins: 150, 1,000 and 2,000. The same region was used for extrapolation for all three, in order to eliminate this as a source of discrepancies in the comparison. Figures 5.2-5.3 show that the KB integrals in the thermodynamic limit increase with the number of bins, and that all different bin sizes give the same trends as functions of mole fraction, both in mean values and in magnitude of standard deviations. This difference between the values also seem to be constant for all mole fractions, and equal for both the ideal and the real LJ mixture. This was validated by checking the value of the integral of the largest volume. The difference between this integral calculated from 150 and 2,000 bins was found to be constant equal to 0.04, while for 1,000 and 2,000 it was constant equal to 0.003. The fact that it was constant for both mixtures and all mole fractions suggests that there was something methodical causing the differences.

Figure 5.4 shows the finite size KB integrals and the values of $(g_{11}(r)-1)$ for the equimolar composition of the ideal mixture, calculated using 150 and 2,000 bins. All data points calculated by the RDF consisting of 150 bins fall on the curve of the RDF consisting of 2,000 bins, but 150 bins does not give good enough resolution provide accurate integration. This is most clearly observed in the first peak, which is higher for 2,000 bins than it is for 150 bins. There is also a small difference arising close to $R=1$, but the area between the two curves at this point is smaller than the difference under the highest peak. This suggests that the integration over the highest peak has the largest effect on the final value when calculating KB integrals in this manner.

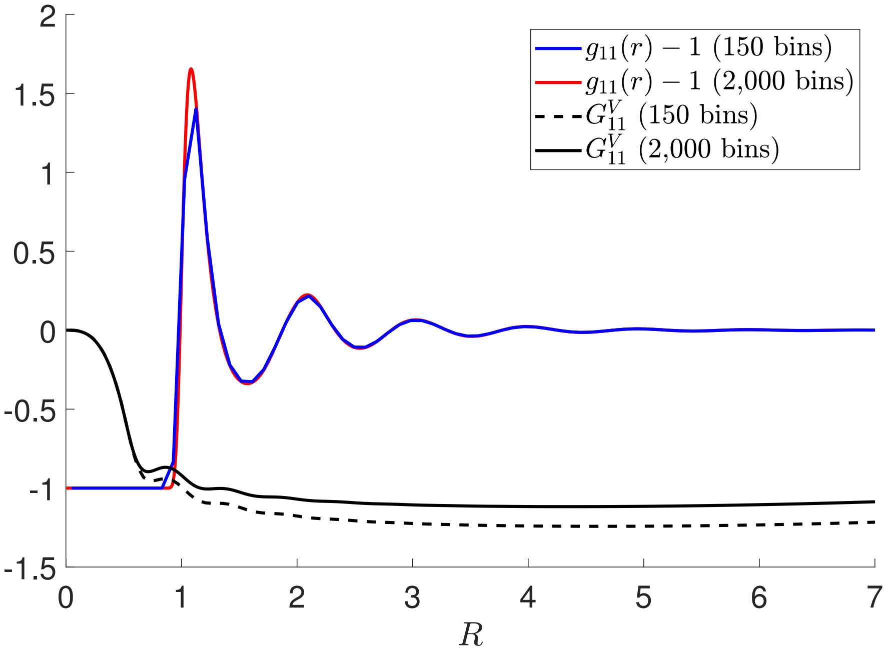

Figure 5.4: RDF of the ideal mixture at $x_1 = 0.50$ computed using 150 bins and 2,000 bins. The KB integrals were calculated using equation (3.38), and are shown as a function of the radius R of the volume they were calculated.The simulation trajectory was sampled every 100 time step, and finite size correction have not been applied.

As an attempt to increase the accuracy of the integration over the highest peak, Simpson's method [76] was tested, but the results did not change significantly. This means that the error arising from using too few data points can not be made up for by using a more accurate integration method. It should also be noted that the differences between the final results become smaller as the number of bins become higher. This suggests that a sufficiently large number of bins is reached when the final value of the KB integral does no longer change upon increasing the number of bins.

5.1.2 Effect of Sampling Rate

The simulation trajectory was sampled both every 100 time step and every 10,000 time step. For most systems, the sampling rate did not change the final value of the KB integral, but for the outer ranges of the mole fraction scale, some differences occurred. For these compositions ($x_1 = 0.05$ and $x_1 = 0.95$), sampling every 10,000 time step resulted in a lot of noise in the curve of the component with the lowest mole fraction, which is a clear indication of poor average values [77]. This is illustrated for RDFs of the ideal mixture at $x_1 = 0.05$, calculated using the two different sampling rates and 2,000 bins in figure 5.5.

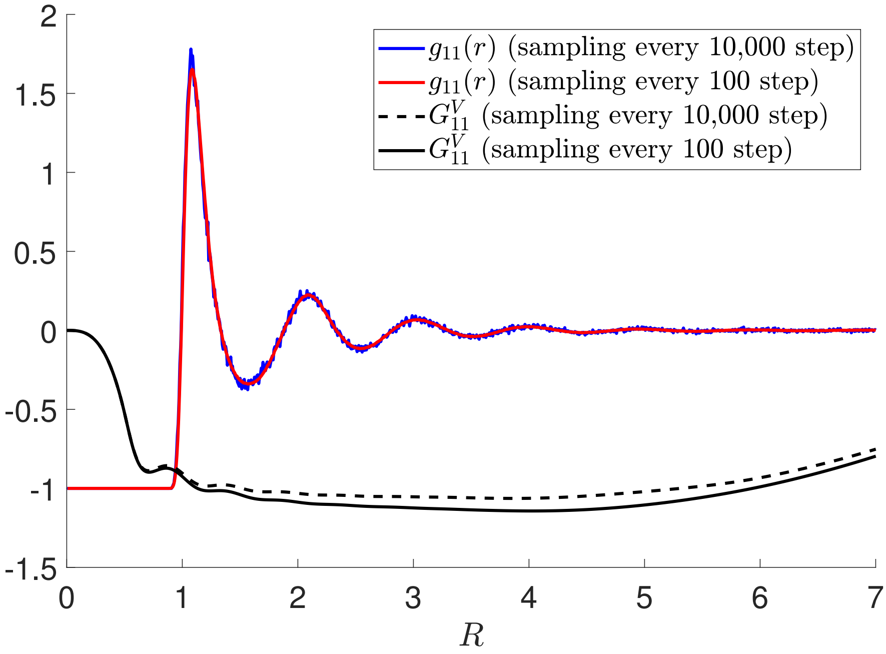

Figure 5.5: RDF of the ideal mixture at $x_1 = 0.05$ computed by sampling the simulation trajectory every 100 time step and every 10,000 time step. The KB integrals were calculated using equation (3.38), and are shown as a function of the radius R of the volume they were calculated. RDF was calculated using 2,000 bins and finite size correction have not been applied.

For the example shown here, the noise lead to an increased value in the highest peak, thereby also increasing the value of the integral. However, it did not affect the final value as much as the difference in bin size, and no systematical trend, governing for all systems was observed. Still, it is another indicator that the integration over the first peak is of great importance when attempting to achieve accurate KB integrals in the thermodynamic limit.

5.1.3 Effect of Finite Size Correction

The correction proposed by Ganguly *et al.* [8] was applied to the RDF of all systems in order to correct for finite size effects. However, it did not give the expected effect, but resulted instead in poorer convergence of the KB integrals. One of these cases is illustrated in figure 5.6 where the original RDF, the corrected RDF and their associated KB integrals are presented for the ideal mixture at $x_1 = 0.50$.

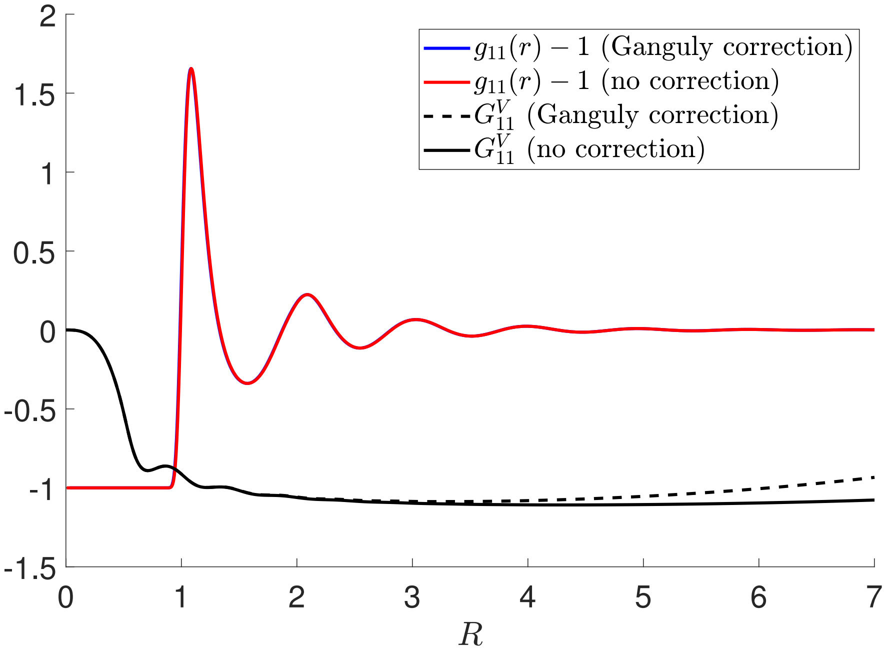

Figure 5.6: RDF and KB integrals of the ideal mixture at $x_1 = 0.50$ with and without the Ganguly correction [8]. The KB integrals were calculated using equation (3.38), and are shown as a function of the radius R of the volume they were calculated. The RDF was calculated using 2,000 bins and sampling every 100 time step.

This was observed for both LJ mixtures, at all compositions, but the difference was larger in the outer ranges of the mole fraction scale. The explanation is probably the unexpected behaviour of the RDF, as discussed initially. The Ganguly correction was designed to correct for the depletion of particle type k around a particle of type i arising from the finite size of the simulation box. As explained previously, some of the RDFs did not reflect this behavior, but indicated instead excess of particles in the cases where depletion was expected (for example figure 5.5 which has a asymptote indicating positive excess). This is even more clearly illustrated by figure 5.7 for the ideal LJ system at $x_1 = 0.50$, where the original RDF has a value a little above unity in the limit, and the corrected RDF has an even larger value. The correction equation therefore probably overestimated the depletion, which resulted in too high values for the RDF, and more drifting asymptotes for the KB integrals. The Ganguly correction itself, is believed to work exactly as intended, but it assumes a different behaviour of the RDF. The KB integrals in the thermodynamic limit calculated from the original RDF are therefore believed to be more accurate than the ones resulting from the corrected RDF.

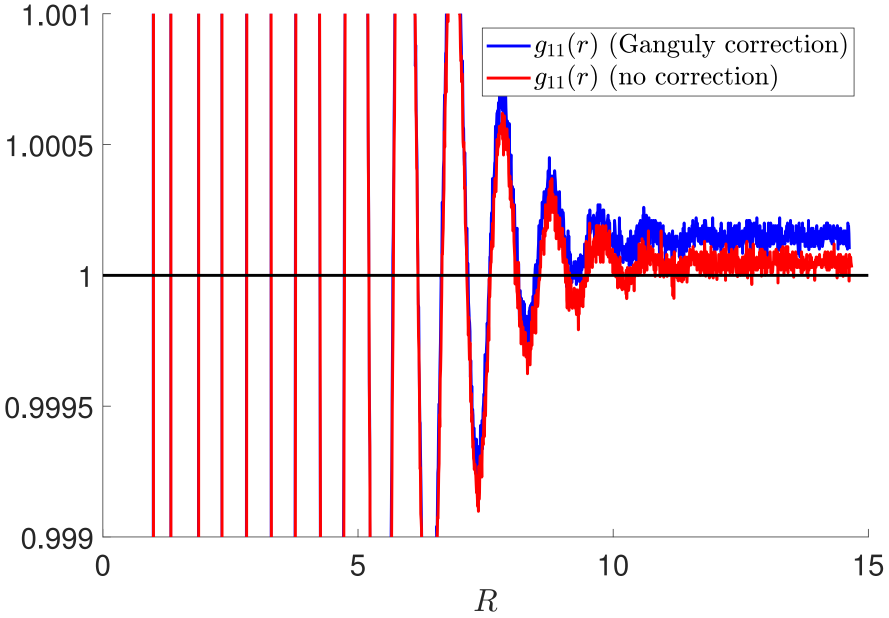

Figure 5.7: Zoomed picture of the RDF of the ideal mixture at $x_1 = 0.50$, with and without the Ganguly correction [8]. The RDF was calculated using 2,000 bins and by sampling every 100 time step.

5.2 Lennard-Jones mixtures - Fluctuation Calculations

From fluctuations in number of particles, the KB integrals in the thermodynamic limit were obtained using different approaches: SSM as described in section 3.2, the method proposed by Cortes-Huerto *et al.* [20] as described in section 3.4, and the method proposed by Galata *et al.* [23] as described in section 3.4.1. The methods proposed by Cortes-Huerto *et al.* [20] and Galata *et al.* [23] contain several factors that differ from the SSM approach. All these factors were isolated and analyzed separately, in order to investigate how each of them affected the final result.

Four different factors were investigated:

1. Calculation of average values of the finite-volume KB integrals.
    - Calculate KB integrals for each subvolume, then calculate average value using equation (3.49), as proposed by Galata *et al.* [23].
    - First calculating the ensemble averages, then combining these in right-hand side of equation (2.19), as normally done in the SSM approach.
2. Definition of subvolume.
    - Superimposed lattice, as suggested by Galata *et al.* [23].
    - Randomly positioned subvolumes, as normally used in the SSM approach.
3. Scaling equation used to obtain KB integrals in the thermodynamic limit.
    - Equation proposed by Cortes-Huerto *et al.* [20].
    - Linear extrapolation, as normally used in the SSM approach.
4. Ensemble of the total simulation box used for sampling of subvolumes.
    - NpT ensemble.
    - NVT ensemble.

In order to validate that the approach suggested by Galata *et al.* [23] was implemented correctly, the KB integrals were first calculated exactly as proposed by the authors. This also included the method for computation of the average KB integral for each subvolume (point 1. in the above list). As explained in section 3.4.1, this method does not follow the principles of equation (2.19), since it calculates fluctuations from time averages for each lattice cell, while this equation requires ensemble averages. The reproduced results showed good overlap with the ones originally obtained by Galata *et al.* [23], confirming that the method was implemented correctly (see appendix B for figures). To directly investigate the effect of the different averaging techniques, these results were also compared to the ones obtained using proper ensemble averages. The two different methods did not give significantly different values for the KB integrals in the thermodynamic limit, but since the method using ensemble averages is considered as most correct, the averaging procedure proposed by Galata *et al.* [23] is not included in the further analysis. See sections 2.2 and 3.4.1 for details on fluctuations and ensemble averages.

Combining the alternatives of points 2. and 3. in the above list made four different combinations available for investigation. The KB integrals in the thermodynamic limit calculated from these four combinations are presented for the ideal LJ mixture in figure 5.8, and for the real LJ mixture in figure 5.9. The effect of using different definition of subvolumes, and the effect of using different scaling equations are further investigated in section 5.2.1. The results presented in figures 5.8-5.9 were obtained by sampling from a total simulation box in the NpT ensemble, while the equivalent results for the NVT ensemble are found in appendix B. How the ensemble of the total simulation box affected the calculated fluctuations is discussed in section 5.2.2. First, the trends as functions of mole fraction are analyzed. Comparison to RDF results are referring to the results obtained using 2,000 bins, without finite-size corrections.

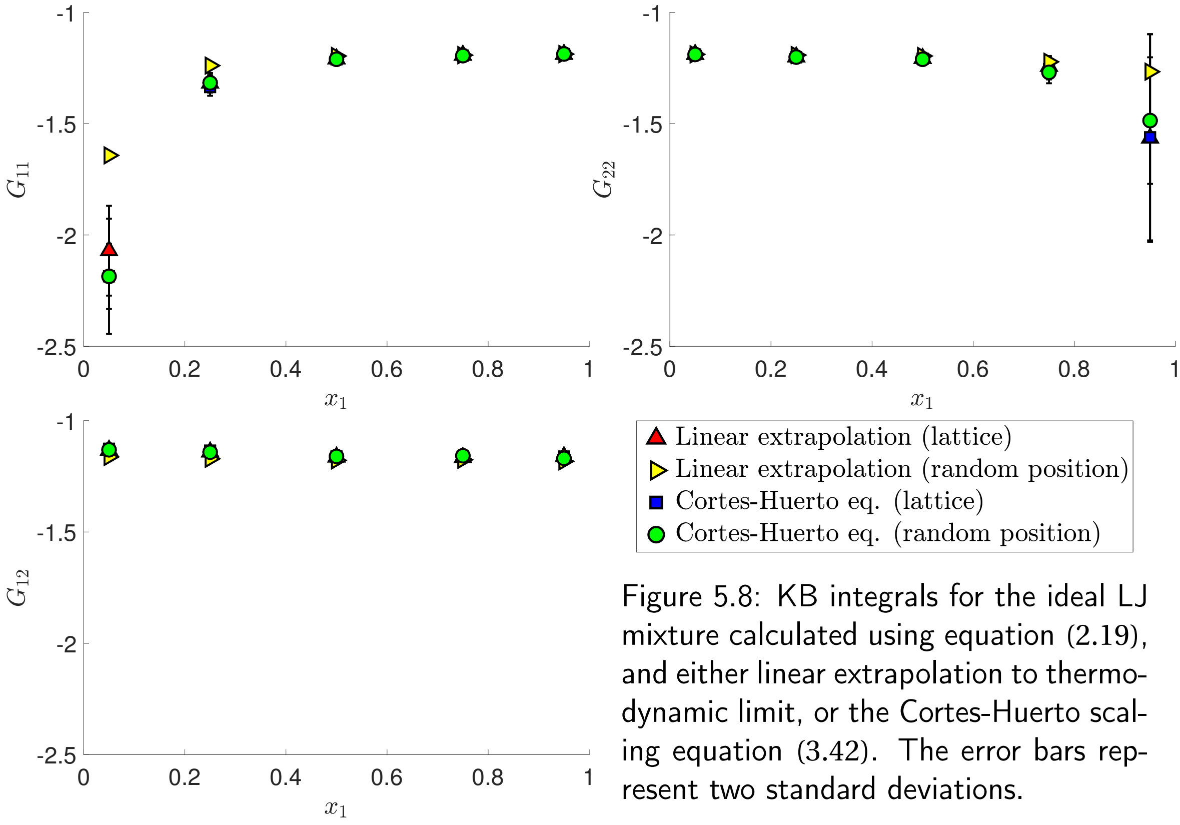

Figure 5.8: KB integrals for the ideal LJ mixture calculated using equation (2.19), and either linear extrapolation to thermodynamic limit, or the Cortes-Huerto scaling equation (3.42). The error bars represent two standard deviations.

For the ideal LJ mixture, the KB integrals calculated from the fluctuations show similar trends to those obtained from integration of RDF. The KB integrals of like components show mostly symmetric behaviour as a function of mole fraction, and larger standard deviations at low mole fractions of their respective component. However, there are some differences. Some of the fluctuation methods vary much more with composition, which resulted in some non-symmetrical values, and the standard deviations are in general larger. For the lattice partitioning especially, the standard deviations are up to five times the magnitude of those obtained by integration of the RDF.

Larger standard deviations in outer ranges of mole fraction scale were observed for all methods, and were probably a result of the statistics of the component of lowest mole fraction, specifically for the computation of the average $\langle N_i^2 \rangle$. When the total number of particles of type i is very low ($N_i = 1000$ for $x_i = 0.05$), the number of pairs of this type in each subvolume will be so low that the average fluctuation $\langle N_i^2\rangle - \langle N_i\rangle^2$ requires longer simulation time to reach its real average value. After $10\cdot 10^6$ time steps, the standard deviation of G_{ii} were half of their value after $4\cdot 10^6$ time steps, while the standard deviations of G_{kk} remained rather unchanged. For these compositions, the average values and their standard deviations could therefore probably have been further improved by longer simulations. This trend in magnitude of standard deviations is not observed for the 1-2 interactions since the number of pairs of unlike particles in each subvolume remain large enough to give good averages and low standard deviations, also in the outer ranges of the mole fraction scale.

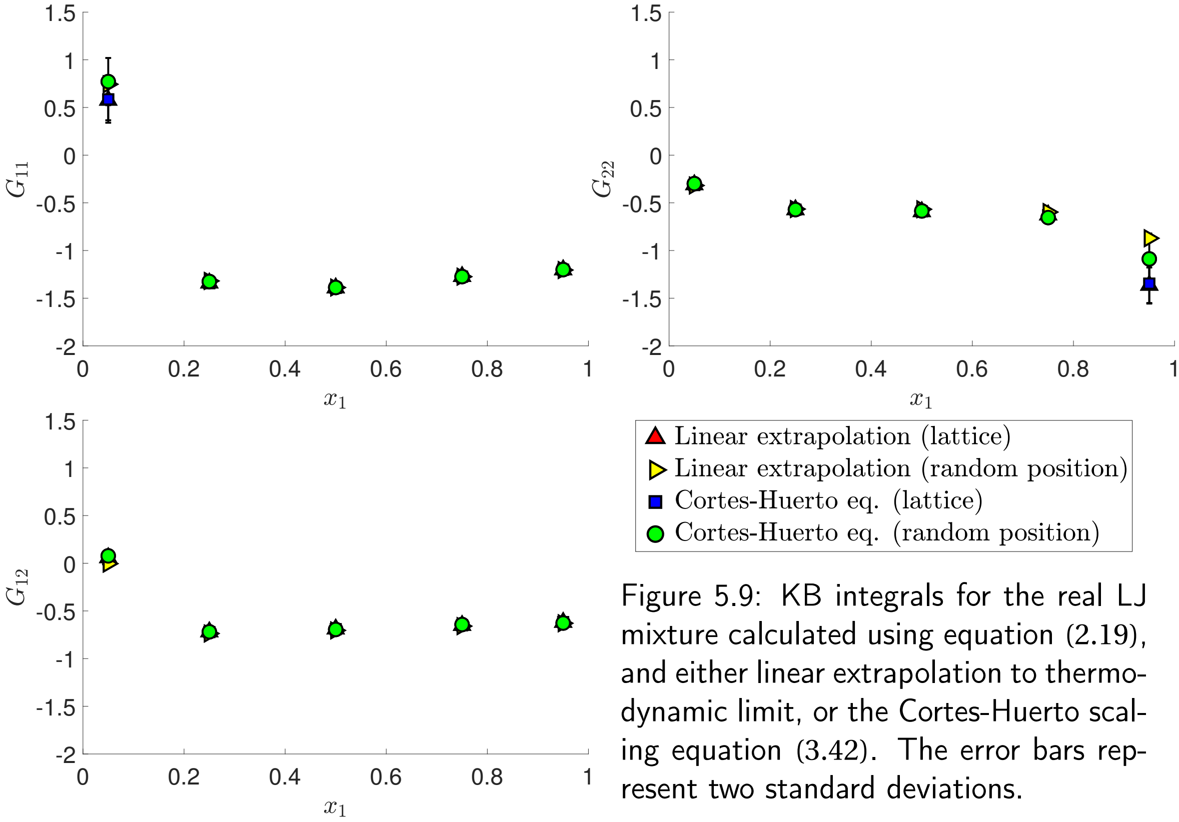

Figure 5.9: KB integrals for the real LJ mixture calculated using equation (2.19), and either linear extrapolation to thermodynamic limit, or the Cortes-Huerto scaling equation (3.42). The error bars represent two standard deviations.

The KB integrals for the real LJ mixture calculated from fluctuations also vary a lot more with composition than the ones obtained by integrating the RDF. The largest difference is observed for G_{11} at $x_1 = 0.05$, where the methods resulted in values that differed by ~ 0.9. This real LJ system at this composition also showed unexpected behaviour in the linear scaling of the property v_{22}, which is further discussed in section 5.2.2.

The overall trend, that can be observed in the results of both LJ mixtures, is that the different fluctuation methods mostly provide values that are within two standard deviations of the other methods. This means that the fluctuation methods do not produce significantly different values of KB integrals in the thermodynamic limit. Another trend that can be observed in figures 5.8-5.9 is that, in general, the results obtained using same subvolumes, but different scaling equations are closer than the results obtained using differently defined subvolumes, but same scaling equation. This suggests that the definition of subvolume has the largest effect on the final value of the KB integrals in the thermodynamic limit.

5.2.1 Effect of Subvolume Definition and Scaling Equation

The averages $\langle N_i \rangle$ and $\langle N_i N_k \rangle$ were calculated for different-sized volumes, using both different subvolume types. These were then used to calculate the properties of interest, n_i, ν_{ik} and G^V_{ik} for all subvolume sizes. After $4 \cdot 10^6$ time steps, these properties showed clear linear regions when plotted as functions of inverse size of the subvolume they were calculated for. An additional $6 \cdot 10^6$ time steps did therefore not result in any visual changes in those plots. However, as already mentioned, it did improve the statistics for the KB integrals in the thermodynamic limit for the components of lowest mole fraction, which means that the statistics were, in fact, improved.

The examples presented in this section are from the real LJ mixture at $x_1 = 0.75$, sampled from a total simulation box in the NpT ensemble. These represent the general trend observed for both LJ mixtures, at all compositions. The one exception is, as already mentioned, the real LJ mixture at $x_1 = 0.05$, which is further discussed in section 5.2.2. The same region was used for curve fitting of the data points obtained from the different subvolume methods, in order to eliminate this as a source of discrepancies in the comparison.

The linear scaling method is based on the principles described in section 3.2. This involved calculating the extensive properties n_i and ν_{ik} for each subvolume size, and plotting these as functions of surface to volume ratio, Ω/V, of their respective systems. The surface contribution was identified as the slope of the linear region of this curve, while the volume contribution was identified as the intersection with the y-axis, also recognized as the value in the thermodynamic limit. These were both determined by linear curve fitting using the `polyfit` function in MATLAB [74] for a first degree polynomial.

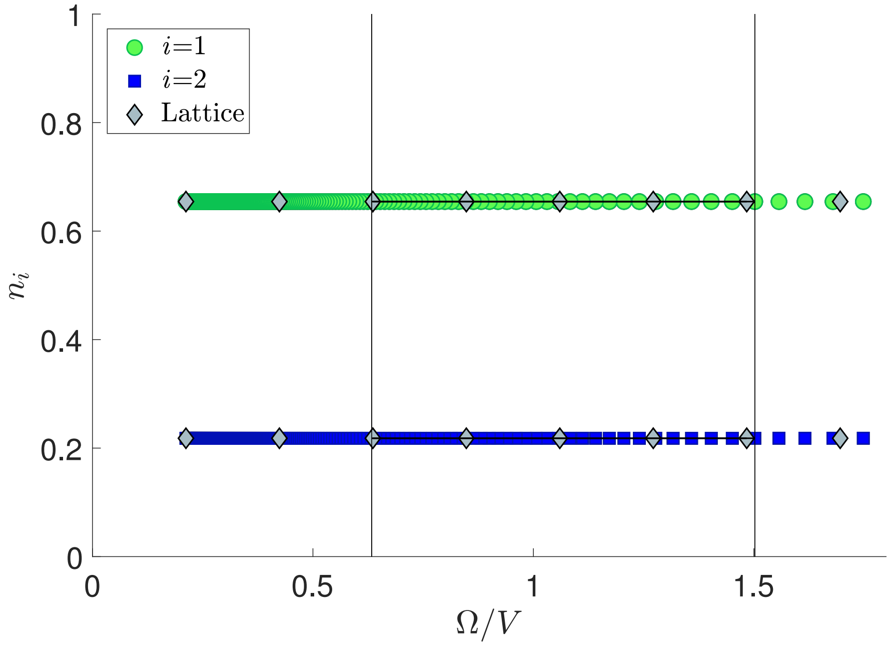

Figure 5.10: n_i for the real LJ mixture at $x_1 = 0.75$ as a function of the surface to volume ratio. Results of both subvolume methods (random positioned and lattice partitioning) are included. Surface and volume contributions were obtained from linear fit to region marked by vertical lines for both subvolume methods.

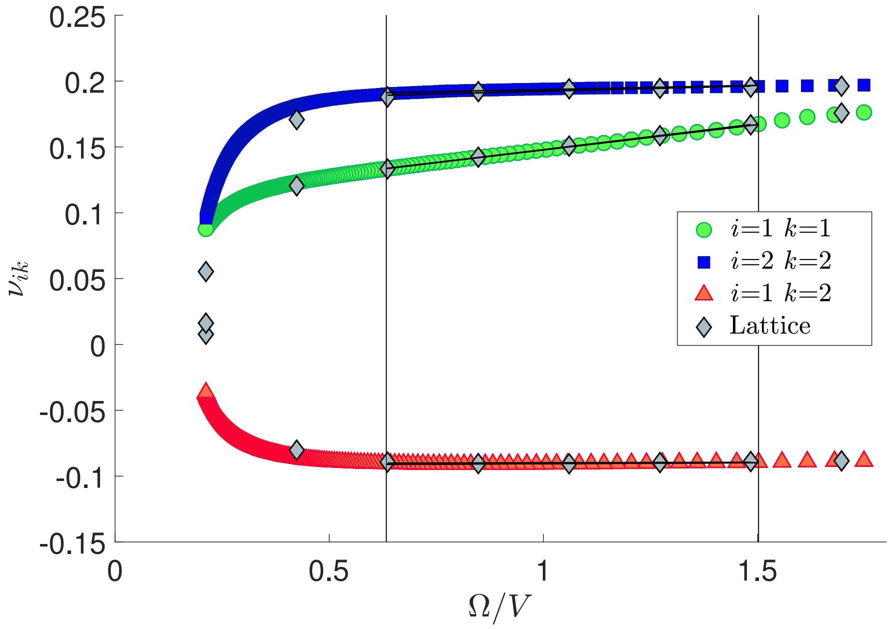

Figure 5.11: ν_{ik} for the real LJ mixture at $x_1 = 0.75$ as a function of the surface to volume ratio. Results of both subvolume methods (random positioned and lattice partitioning) are included. Surface and volume contributions were obtained from linear fit to region marked by vertical lines for both subvolume methods.

In order to directly investigate the effect of the definition of subvolume, the properties n_i and v_{ik} calculated from both subvolume types were plotted together in figures 5.10-5.11. Strøm *et al.* [49] showed in their work that the extensive properties, calculated using subvolumes of different shapes, fall on the same curve when plotted as functions of surface to volume ratio. Figures 5.10-5.11 show that for the LJ mixtures, the data points do fall on the same curve, confirming that both subvolume types provide the same average densities and fluctuations, under the conditions studied here. Some differences can be observed for v_{ik} in figure 5.11 as the size of the subvolume approaches the size of the simulation box. As described in section 3.2, this is because the total simulation box no longer functions as a grand canonical reservoir at this point. For a closed system, a density change in the subvolume can not occur without a corresponding density change in the reservoir [49]. This correlation becomes larger as the subvolume approaches the size of the simulation box, and makes the curves drop off and approach zero. A cube and a sphere of equal area to volume ratio will not have equal volumes. The cube's volume will be larger, meaning that the above mentioned correlation will be larger for the cube. This is why the values calculated from cubic subvolumes drop off for smaller values of Ω/V, while the the ones calculated using spherical subvolumes have a longer linear region. The properties calculated from cubic subvolumes usually also contain shape effects for smaller subvolumes, caused by their nook and corners. These are absent for spheres, which is why spherical subvolumes in general are preferred. For the LJ mixtures studied here, the lattice cells were not small enough to any display nook and corner effects.

The average number density was for both systems, at all mole fractions, observed to be constant as a function of system size, hence $n_i^{\mathrm{s}} = 0$. This means that the KB integrals could be directly calculated for each subvolume size using

$$G_{ik}^{V} = V\left(\frac{\langle N_i N_k\rangle - \langle N_i\rangle\langle N_k\rangle}{\langle N_i\rangle\langle N_k\rangle} - \frac{\delta_{ik}}{\langle N_i\rangle}\right) \tag{5.1}$$

since this expression now becomes a linear combination of the extensive properties. The resulting KB integrals were then extrapolated to the thermodynamic limit by plotting these values as functions of the surface to volume ratio, as shown in figure 5.12. This approach gave the same results as when surface and volume contributions of n_i and v_{ik} were combined in equation (3.30), and then extrapolated to the thermodynamic limit, as shown in figure 5.13. The direct extrapolation of G_{ik}^{V} from equation (5.1) showed smaller standard deviations than the results obtained by splitting into extensive properties.

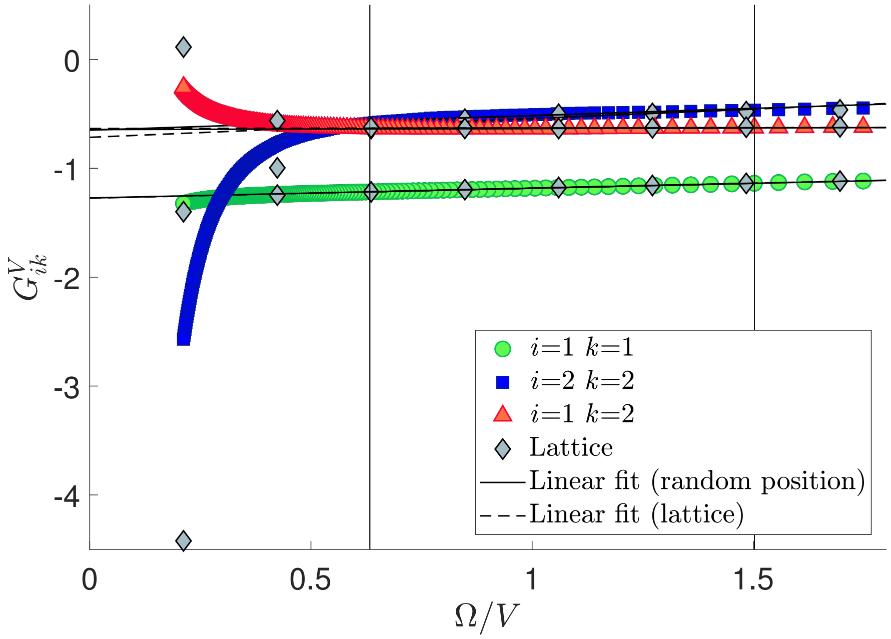

Figure 5.12: G^V_{ik} for the real LJ mixture at $x_1 = 0.75$ calculated directly from equation (5.1), as a function of surface to volume ratio. Results of both subvolume methods (random positioned and lattice partitioning) are included. Thermodynamic limit value G_{ik} was obtained from linear fit to region marked by vertical lines for both subvolume methods.

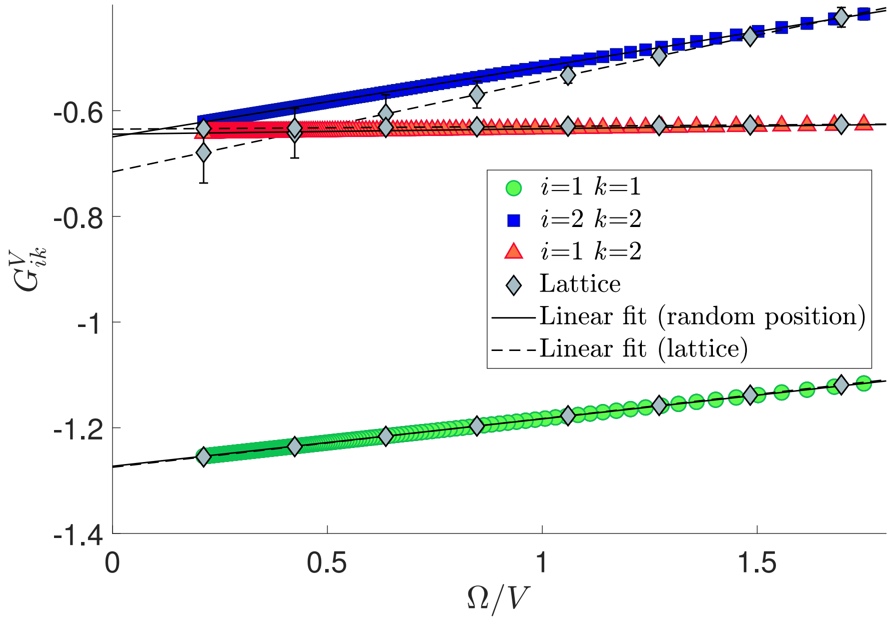

Figure 5.13: G^V_{ik} for the real LJ mixture at $x_1 = 0.75$ calculated from equation (3.30), as a function of surface to volume ratio. Results of both subvolume methods (random positioned and lattice partitioning) are included. Thermodynamic limit value G_{ik} was obtained from linear fit to all data points.

The figures 5.12-5.13 do not show the same response to the finite size of the total simulation box. While KB integrals for $\Omega/V \to 0$ in figure 5.12 were clearly affected by the finite size of the reservoir, the data points in figure 5.13 show no deviations from linearity. This is because these were calculated from the surface and volume contributions of the extensive properties in the linear regions, where the fluctuations were not affected by the finite size of the closed reservoir. Both figures also show that the two subvolume types do not always extrapolate to the same value, even when their curves overlap. This explains the differences between the final values obtained by the different methods observed in figures 5.8-5.9, and is further discussed later in this section.

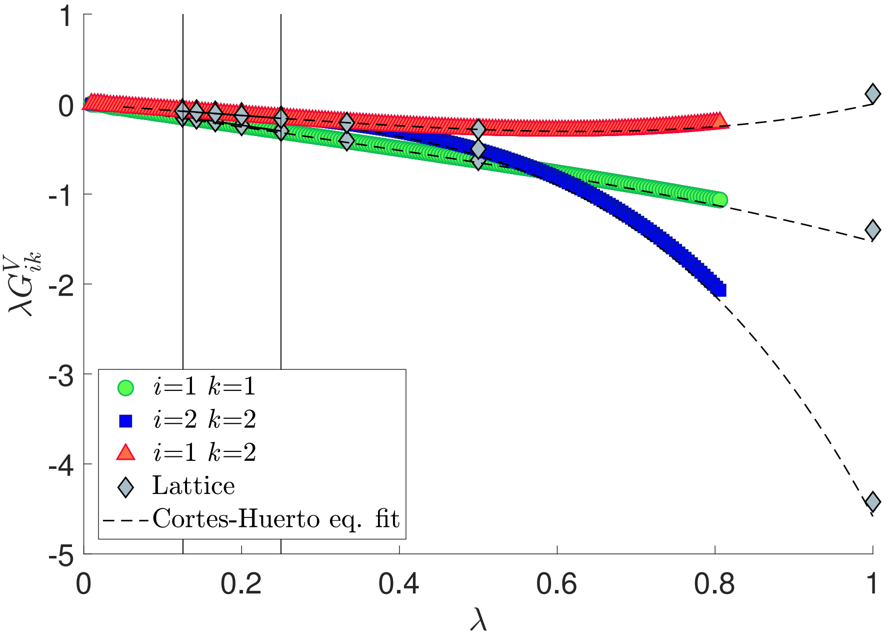

Figure 5.14: λG_{ik}^V for the real LJ mixture at $x_1 = 0.75$ calculated from equation (3.42), as a function of the parameter $\lambda = (V/V_0)^{1/3}$. Results of both subvolume methods (random positioned and lattice partitioning) are included. Thermodynamic limit value G_{ik} was obtained as the slope of linear fit to the region marked by vertical lines.

Figure 5.14 illustrates the procedure of the scaling method proposed by Cortes-Huerto *et al.* [20], where the KB integrals were calculated using equation (5.1), multiplied by the factor λ and plotted as a function of λ. The KB integral in the thermodynamic limit were then determined by finding the slope of the linear region, using the `polyfit` function in MATLAB [74] for a first degree polynomial. This region included approximately the same subvolume sizes as used for linear extrapolation. The two scaling methods are therefore essentially the same, since they both utilize how the KB integrals scale linearly with the inverse size of the subsystems. The only difference is that the reference for the Cortes-Huerto scaling is the volume of the total simulation box, while the reference for the linear extrapolation is the shape dependent coefficient, $c_s = \Omega/V^{2/3}$, discussed by Strøm *et al.*

[49]. The Cortes-Huerto equation does not take the finite size of the reservoir into account as long as only the KB integrals of the smallest subvolumes are used for scaling. A suggestion that could improve the scaling technique is introduced later in this section.

The values obtained in the thermodynamic limit by Cortes-Huerto equation were overall close to the ones obtained by linear scaling, except for at the lowest mole fractions. The differences observed for the different scaling equations were probably related to differences in location of the region used for curve fitting. These cover many of the same sized subvolumes, but they do not completely overlap. When using the Cortes-Huerto scaling equation, the linear region can be harder to identify than when using linear extrapolation. In their original paper, Cortes-Huerto *et al.* [20] introduced the conditions that should be used to determine which data points to include in the curve fitting. These are as follows: the smallest subvolume must be larger than V_ξ, while the upper limit of the scaling factor λ is 0.3. For the systems studied here, the structural correlation length is $\xi \approx 7$, giving a volume, $V_\xi \approx 1,437$, that corresponds to a scaling factor of $\lambda \approx 0.37$. This means that the region that should be used for the extrapolation does not exist for the LJ systems studied here. To get a region that could be used for extrapolation for these systems, according to the condition given by Cortes-Huerto *et al.* [20], the number of particles would need to be double the amount used in this work. If the method requires systems this large to provide trustworthy results for dense systems, the whole point of extracting KB integrals from small sized simulations disappears. Since the scaling normally performed by SSM essentially is the same as the one proposed by Cortes-Huerto *et al.* [20], this would mean that they both fail when using subvolumes smaller than V_ξ. However, figure 5.14 shows that using the values extracted from the linear fit, inserted in equation (3.47) still provide good fit of the remaining data points. The curves also show that λG_{ik} approach $-\delta_{ik}/\rho_i$ when $\lambda = 1$, which the expected behaviour when the KB integrals are calculated for a closed simulation box (see section 2.4). It is therefore possible that the condition for minimum subvolume sizes is not as strict as originally stated by Cortes-Huerto *et al.* [20].

Since all data points obtained from different subvolume types fall on the same curve for both scaling methods, the difference in the values of KB integrals thermodynamic limit were probably a consequence of the accuracy of the curve fitting. This argument is also substantiated by the fact that direct extrapolation of the values calculated from equation (5.1) gave smaller standard deviations than when in addition extrapolating the extensive properties. The difference in the thermodynamic limit value obtained from the different subvolume types are therefore probably a result of their difference in number of data points within the linear region. This number for the lattice partitioning was 5, while the same region contained 32 data points from the randomly positioned subvolumes. This means that each data point calculated from the lattice partitioning had a much larger influence on the fitted curve, compared to the influence of one data point from the randomly positioned subvolumes. As explained in section 4.4, the lattice partitioning also created an unequal number of samples per subvolume, which resulted in large differences in their standard deviations. For the second largest subvolumes, the standard deviations were in general ten times the standard deviation of the smallest subvolumes, meaning that these probably should not have been weighed equally in the curve fitting. As earlier mentioned, longer simulation time affected the standard deviations of the results obtained from lattice subvolumes much more the ones obtained from random positioned subvolumes. This could be related to the fact that the lattice cells were at all times right next to each other, and therefore not producing independent samples. The values obtained from each different-sized subvolume therefore required longer simulation time to reach their real average value, which lead to slower convergence of the linear region. The superimposed lattice is therefore a less robust sampling method.

The accuracy of the results in the thermodynamic limit could be improved in two ways. The first alternative is longer simulations, which would result in improved statistics for each data point. The second alternative is to generate additional data points in the linear region, which would improve the statistics of the fitted curve without having to run longer simulations. When using the lattice partitioning, the number of data points is constrained by the size of the total box, and the only way to improve the statistics of G_{ik} is therefore longer simulation time. When using the randomly positioned subvolumes, all sizes within the total reservoir are available. In addition, the statistics of these data points could be improved by increasing the number of points (samples) used for positioning of the subvolumes. However, this would increase the time spent on these calculations. An even faster method of obtaining more data points in the linear region, would be to use subvolumes with linearly increasing reciprocal sphere radius $1/R$. Since the data points used for curve fitting in linear scaling and Cortes-Huerto scaling cover the approximately the same range of subvolume sizes, this would increase the number of data points used for curve fitting by both methods.

It would also be possible to better exploit the data points already available, had they been calculated from cubic subvolumes. Strøm *et al.* suggested an alternative scaling equation that could be used for the extensive properties, also for those calculated in the largest embedded subvolumes (equation (3.32) described in section 3.2). However, this requires subvolumes of same shape as the reservoir, which means that it could not be used for the data points obtained from the randomly positioned subvolumes used in this work. The Cortes-Huerto equation is already designed to describe the whole range of different-sized subvolumes inside a closed reservoir of finite size. It could therefore be possible to calculate corrected values of G_{ik} and C_{ik} already obtained from the linear curve fitting. This could be obtained by using the values of these as initial guesses, and perform least squares curve fitting of the total equation (3.47) for all data points. This would lead to values of G_{ik} and C_{ik} that were more in agreement with the entire range of subvolumes.

5.2.2 Effect of Ensemble

The underlying principle of sampling fluctuations in this manner is that the subvolumes produce particle fluctuations from the grand canonical ensemble. If the simulation box itself also is in the grand canonical ensemble, this assumption will be valid for all different sizes of subvolumes. However, if the simulation box is in a different ensemble than the grand canonical (for example the NVT, NpT or NVE ensemble), the assumption is only good for subvolumes much smaller than the reservoir. These are therefore the only subvolumes that can be used to extract grand canonical fluctuations. Most previously published work were performed using a reservoir in the NVE or NVT ensemble [20] [21] [24] [25] [49] [51] [78], but a few studies have also reported use of the NpT ensemble [8] [23]. This section presents a direct comparison of the effect of the ensemble of the reservoir. Both the values in the thermodynamic limit, and the values calculated for different-sized subvolumes are compared. It is important to distinguish between ensemble of the simulation box and ensemble of the small systems. When NpT- or NVT-results are mentioned in the following section, it therefore refers to the particle fluctuations calculated the embedded subvolumes in a simulation box in the NpT or NVT ensemble, and not from actual NpT or NVT ensembles. Particle fluctuations can not be directly calculated from NpT or NVT systems, since these both have constant number of particles.

The KB integrals in the thermodynamic limit obtained from the NpT simulations were presented previously in this section, while the results obtained from the NVT simulations are found in appendix B. These values showed some differences in the outer ranges of the mole fractions, but the overall magnitude of the differences were not significant. To directly investigate the effect the ensemble of the reservoir, the values of ν_{ik} for the real LJ mixture at $x_1 = 0.75$ are plotted for both ensembles in figures 5.15-5.16. Figure 5.15 displays the behaviour close to the thermodynamic limit, while figure 5.16 compares the results for the whole range of different-sized subvolumes.

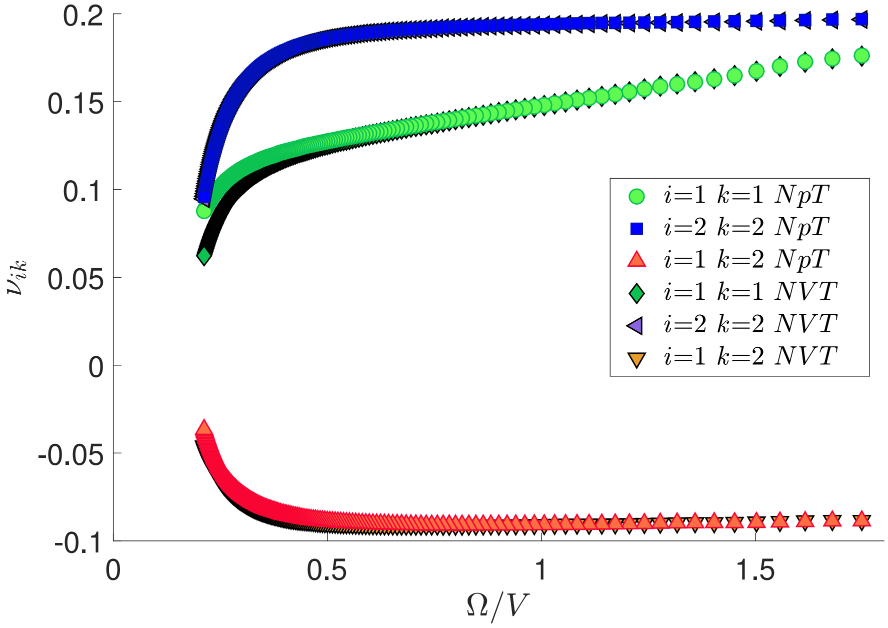

Figure 5.15: ν_{ik} for the real LJ mixture at $x_1 = 0.75$ as a function of the surface to volume ratio. Results obtained by sampling fluctuations from a total simulation box in the NpT and NVT ensemble are both included.

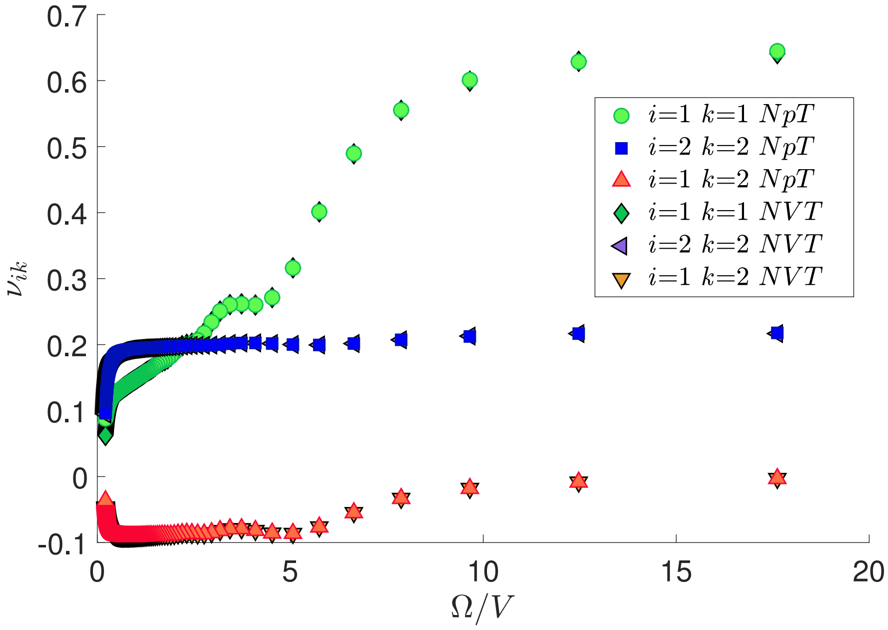

Figure 5.16: ν_{ik} for the real LJ mixture at $x_1 = 0.75$ as a function of the surface to volume ratio for the whole range of volumes used for sampling. Results obtained by sampling fluctuations from a total simulation box in the NpT and NVT ensemble are shown. For $\mathit{\Omega}/V > 1$ these completely overlap.

The expected behaviour of the property ν_{ik} for volumes close to the size of the reservoir was discussed in the last section. At the other end of this scale, for subvolumes of the size of one particle, the property, ν_{ik}, for like components is expected to approach the number density, while it for unlike component should approach zero (see section 3.2). Figure 5.16 clearly shows that the result obtained from NpT and NVT simulations both follow this expected behaviour, since $n_1 = 0.65$ and $n_2 = 0.22$ for this system.

Both figures 5.15-5.16 also show that the data points completely overlap for the smaller subvolumes ($\Omega/V > 1$), but that differences arise for larger subvolumes. This difference is largest for ν_{11}, which probably is because component 1 is in excess for this composition, and therefore also representing the main part of the particle fluctuations. The NpT-results display a longer linear region, while the NVT-results start approaching zero for smaller subvolumes. This behaviour can be explained by the connection between the density changes in subvolume and reservoir, as described in the previous section. The fluctuating volume of the NpT simulation box probably lead to a density change in the reservoir that amplified the particle fluctuations of the largest embedded subvolumes. These fluctuations could be artificial, meaning that they are not represented by any known ensemble, or they could be a result of the direct relation between the μVT ensemble and the NpT ensemble given by equations (2.13)-(2.12). As stated in section 2.2, the relative fluctuations of volume in the NpT ensemble are equal to the relative fluctuations in number of particles in the μVT ensemble, if the constant volume of the μVT system is equal to the average volume of the NpT system. Equation (2.12) is only valid for particle fluctuations in one-component systems, but the behaviour observed in figure 5.15 could suggest that the connection has a similar effect on the particle fluctuations of the excess component in mixtures. It is clear that a connection is reflected by the data, but the question is now whether the extended linear region of the NpT-results represent proper grand canonical fluctuations due to it. In order to investigate this connection directly, the particle and volume fluctuations were calculated for a pure system consisting only of component 1. Since $n^{\mathrm{s}} = 0$ also was valid for the pure system, the isothermal compressibility could be directly extrapolated from

$$\kappa_T T = \frac{V}{k_{\mathrm{B}}} \frac{\langle N^2 \rangle - \langle N \rangle^2}{\langle N \rangle^2} \tag{5.2}$$

The results of the isothermal compressibility for the pure system, calculated for the different-sized subvolumes embedded in total simulation boxes in the NpT and the NVT ensemble are presented in figure 5.17. These both extrapolate close to the same value in the thermodynamic limit, but show very different behaviour for the largest of the embedded volumes. The NVT-results start approaching zero when the size of the subvolume approaches the size of the simulation box, while the NpT-results display a more linear behaviour. However, values of $\kappa_T T$ calculated from these subvolumes deviate from the extrapolated line, which means that they do not represent proper grand canonical fluctuations.

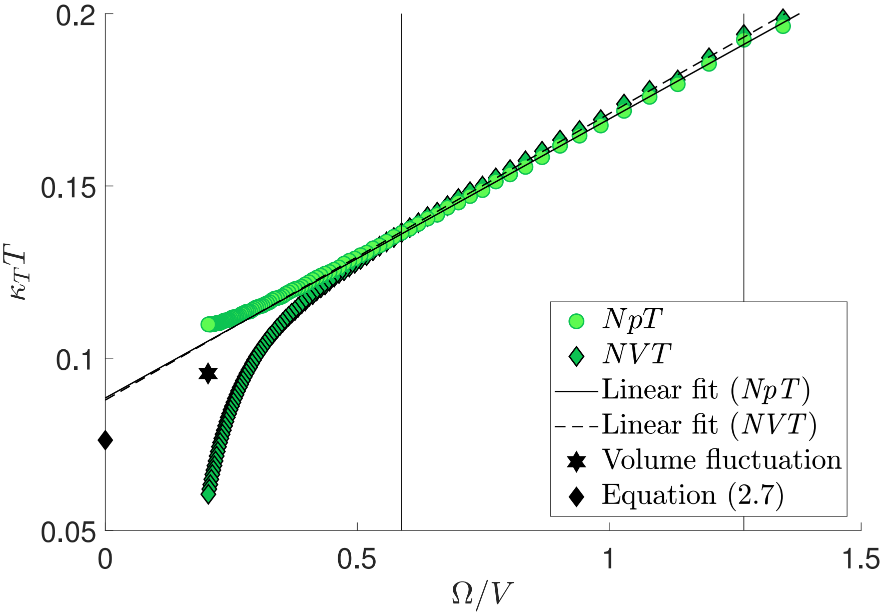

Figure 5.17: $\kappa_T T$ for a pure system of component 1, calculated directly using equation (5.2). Results obtained by sampling fluctuations from a total simulation box in the NpT and NVT ensemble are shown. The hexagon represent value calculated from volume fluctuations by equation (2.13), and the diamond represent the value calculated from solving equation (2.7) numerically.

The particle fluctuations for a volume, V, should give results equal to the volume fluctuations of average volume, $\langle V \rangle$, equal to V. The isothermal compressibility was therefore calculated from volume fluctuations of the total simulation box in the NpT ensemble using equation (2.13), and included in the plot at the position corresponding to the surface to volume ratio of the total simulation box. $\kappa_T T$ was also calculated from equation (2.7), by systematically increasing the pressure of the system, such that the volume was obtained as a function of pressure. The differential $(\partial V/\partial p)$ was then solved numerically, and included in the plot at $\Omega/V = 0$. Figure 5.17 shows that neither of those data points fall on the extrapolated line. This could mean that either, the sampled particle fluctuations did not represented grand canonical distributions properly, or that both alternative methods for computation of $\kappa_T T$ failed. However, the tail corrections could have affected the output value of the pressure, which in turn would have affected the values of $(\partial V/\partial p)_T$

and the average volume fluctuations. This would explain why the values of κ_T calculated from equation (2.7) and (2.13) differ by up to 10%, from the value calculated from particle fluctuations.

It also is possible that the linear region is misrepresented in figure 5.17, and that it actually expands closer to $\Omega/V \to 0$. Including 30 more data points in this direction, and leaving out 10 in the other end, resulted in a line that better fit the two individual data points, as shown in figure 5.18. However, it resulted in much larger difference in the extrapolated value of the NpT-result and the NVT-results, in addition to a poorer fit of the NVT-data than what was shown in figure 5.17.

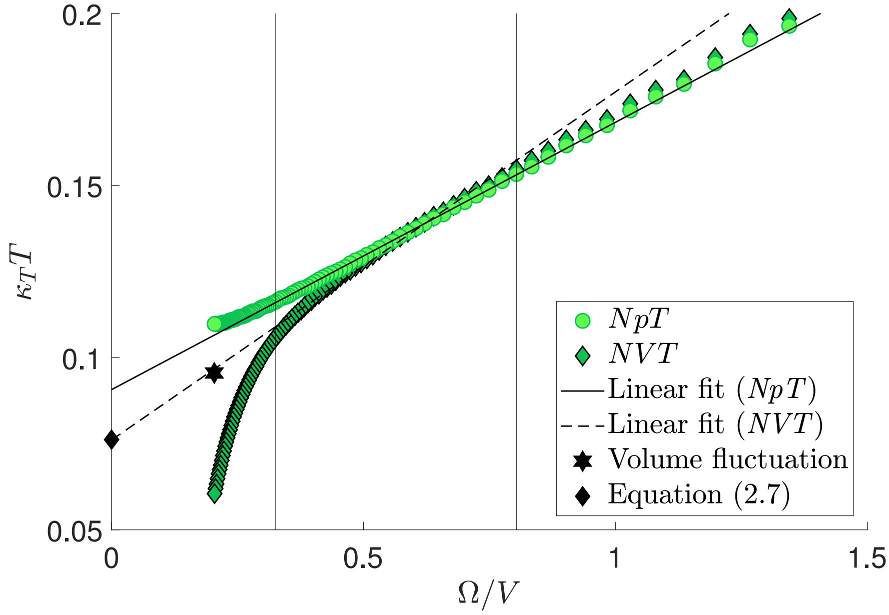

Figure 5.18: $\kappa_T T$ for a pure system of component 1, calculated directly using equation (5.2). Results obtained by sampling fluctuations from a total simulation box in the NpT and NVT ensemble are shown. The hexagon represent value calculated from volume fluctuations by equation (2.13), and the diamond represent the value calculated from solving equation (2.7) numerically.

As already mentioned, the real LJ mixture at $x_1 = 0.05$ showed unexpected behaviour, deviating from the rest of the cases investigated. This is shown in figure 5.19, where both ν_{22} and ν_{12} obtained from a total simulation box the NpT ensemble are increasing for larger sized subvolumes. This strengthens the argument stating that the particle fluctuations are amplified by volume fluctuations, since the volume of the simulation box for this system fluctuated ~ 2% from the mean value, while all other systems only fluctuated ~ 0.5% from the mean value. The fluctuations in number of particles calculated from embedded volumes in the NVT simulation box also show much larger values for this system than what was observed for other compositions (figure 5.11 represents this behaviour). This means that the system is, in fact, characterized by large particle fluctuations at this composition, and that they were not just a result of density changes in the total system. The large fluctuations could indicate that the system is close to the critical point, or that two phases were present. To investigate whether the system was heterogeneous, a histogram of the number of particles in the smallest lattice cells was calculated, and presented in figure 5.20. If two phases were present, this histogram would consist of two peaks, representing the two different densities in the system. Figure 5.20 instead consists of only one, clear peak at ~ 39, recognized as the average number of particles inside the smallest lattice cells. This indicates that the system is homogeneous, but it is still possible that it is close to the critical point, given the large magnitude of the fluctuations.

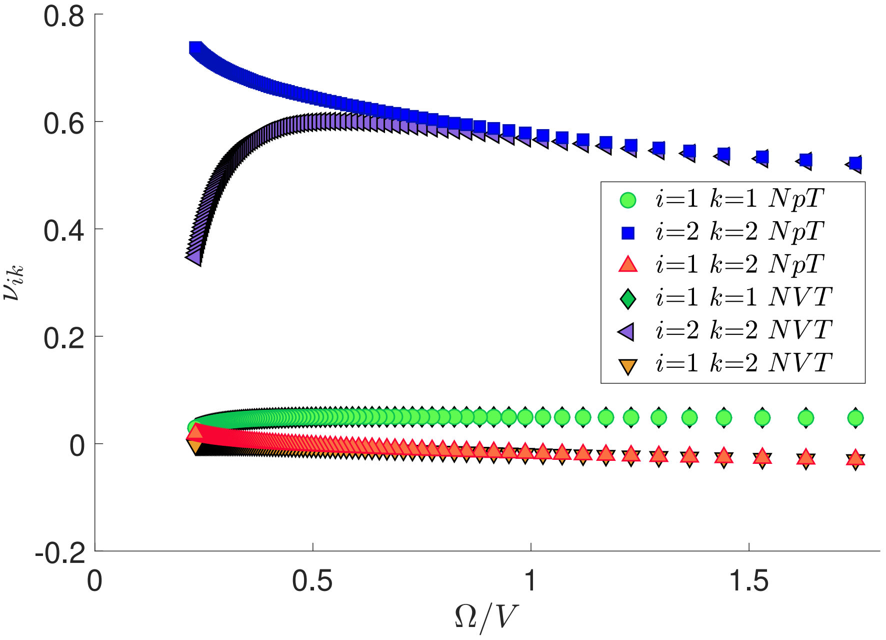

Figure 5.19: ν_{ik} as a function of the surface to volume ratio for the real LJ mixture at $x_1 = 0.05$. Fluctuation sampled from a total simulation box in the NpT and the NVT are both included.

All together, the results presented in this section show that the particle fluctuations calculated in subvolumes are clearly affected by the volume fluctuations in the NpT ensemble. A reservoir in the NVT is therefore considered to be a safer choice. Figures 5.17-5.18 illustrate the difficulty often encountered when attempting to determine the linear region, while the behaviour of the real LJ mixture at $x_1 = 0.05$ showed the importance of checking the behaviour of the extensive properties. The expected behaviour of ν_{ik} is well known, and discrepancies are therefore more easily discovered. The same discrepancies are not as easily observed for plots of the directly calculated KB integrals.

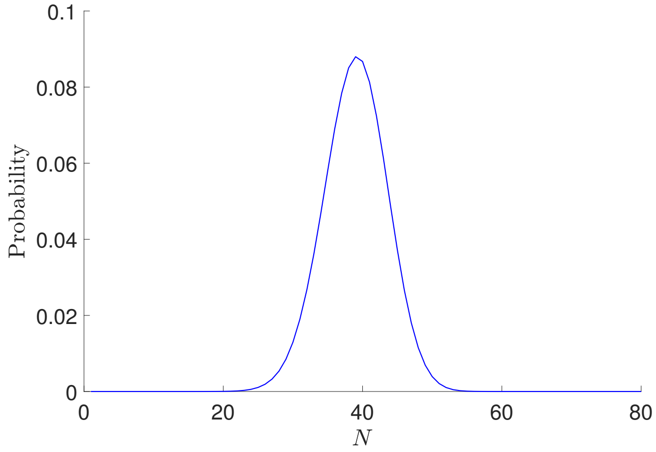

Figure 5.20: Particle number distribution for the smallest lattice cells of the real LJ mixture at $x_1 = 0.05$.

5.3 Lennard-Jones Mixtures - Comparison to Theoretical Values

A large number of different properties are available from combinations of KB integrals of mixtures [79]. Some of the most interesting ones are the derivative of the chemical potential $(\partial\mu_1/\partial x_1)_{T,p}$, isothermal compressibility κ_T, and the partial molar volumes V_1 and V_2. Results obtained from KB integrals in the figures presented in this section are calculated from a total simulation box in the NVT ensemble, and show the results of all different fluctuation methods, and the RDF results using 2,000 bins without finite-size corrections. A more thorough investigation of the accuracy of the methods is given in section 5.3.3.

5.3.1 Ideal Lennard-Jones Mixture

The property $\Delta G_{12} = G_{11} + G_{22} - 2G_{12}$ is a part of many expressions for thermodynamic properties and has a theoretical value of zero for ideal mixtures [4]. The accuracy of this value therefore determines the accuracy of many of the properties calculated. Figure 5.21 shows ΔG_{12} as a function of mole fraction of component 1, where it becomes clear that the integration of RDF gave the results closest to zero. The fluctuation method that show the best accuracy is the one using randomly positioned subvolumes and linear scaling. Figure 5.22 shows $(\partial\mu_1/\partial x_1)_{T,p}$ calculated from (2.20) as a function of mole fraction of component 1, together with the theoretical prediction given by equation (2.6). The KB integrals are only included in the calculation of this property through ΔG_{12}. The accuracy of $(\partial\mu_1/\partial x_1)_{T,p}$ is therefore completely governed by the accuracy of ΔG_{12}, which means that the trends in figures 5.21-5.22 are exactly the same.

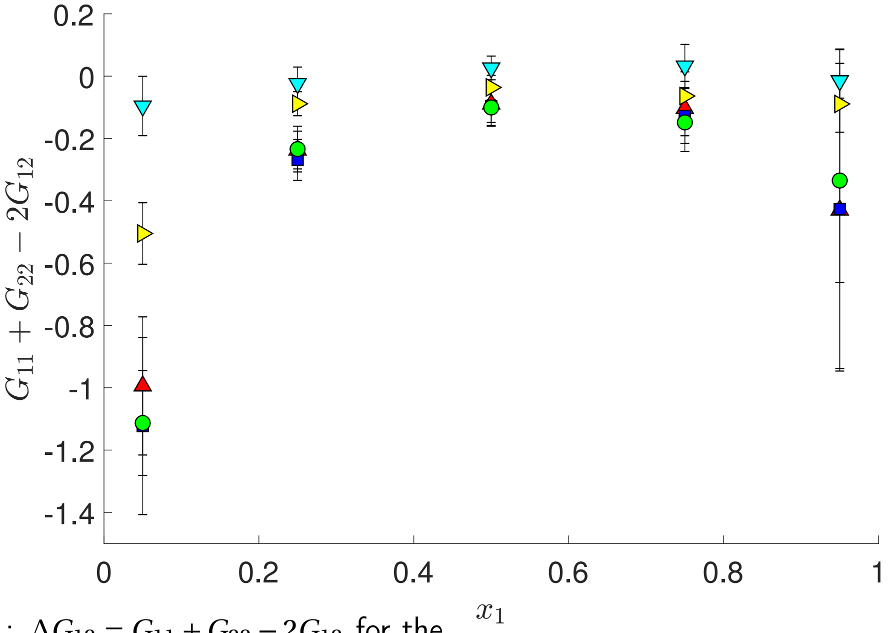

Figure 5.21: $\Delta G_{12} = G_{11} + G_{22} - 2G_{12}$ for the ideal LJ mixture, calculated from KB integrals obtained by the different fluctuation methods and RDF using 2,000 bins. The error bars represent two standard deviations.

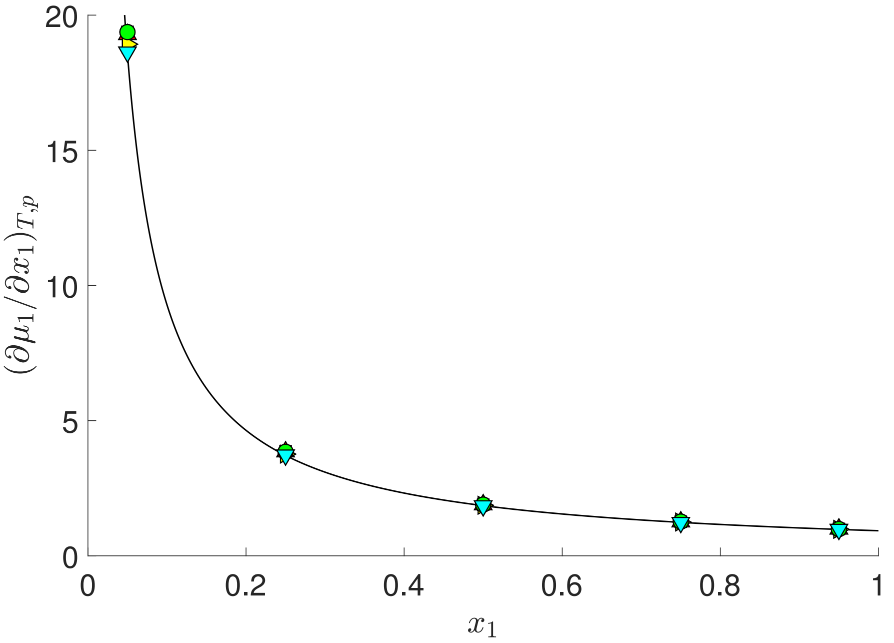

Figure 5.22: $(\partial\mu_1/\partial x_1)_{T,p}$ for the ideal LJ mixture, calculated from KB integrals combined in equation (2.20). Theoretical prediction given by (2.6). The error bars represent two standard deviations.

The isothermal compressibility was calculated by combining the KB integrals in equation (2.21). In addition to ΔG_{12}, this equation also includes the term $G_{11}G_{22} - G_{12}^2$, and the individual KB integrals for like interactions multiplied by their associated concentration. As already mentioned in section 2.2, κ_T is also available through two other methods. The first method uses equation (2.13) to calculate the isothermal compressibility from fluctuation of the total volume of the simulation box. The second is systematically increasing the pressure of the system, in order to solve equation (2.7) numerically. Figure 5.23 shows the results of these two methods, together with the results calculated from the KB integrals. All methods, except the integration of the RDF, provided constant values of κ_T as a function of mole fraction. This was expected for the ideal mixture since it actually was a pure system with differently labelled particles. The standard deviations for all methods are now smaller in the outer ranges of the mole fraction scale, which probably is a result of the weighting of the KB integrals of the different components in equation (2.21).

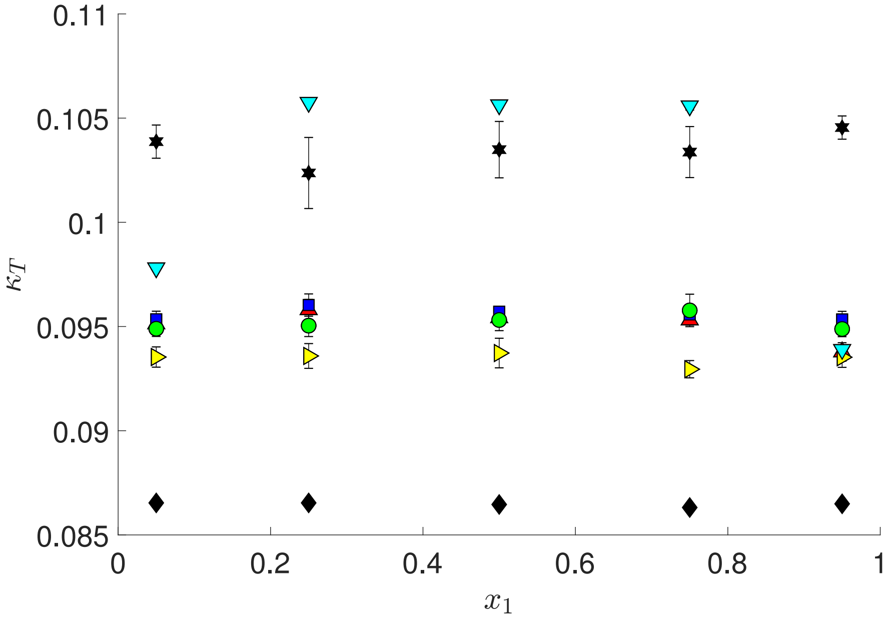

Figure 5.23: κ_T for the ideal LJ mixture, calculated from KB integrals combined in equation (2.21), from volume fluctuations using equation (2.13), and by solving (2.7) numerically. The error bars represent two standard deviations.

Another interesting trend in figure 5.23 is the differences between the results obtained using equation (2.7), volume fluctuations, and particle fluctuations. These show the exact same relative trend as displayed for the pure system in figure 5.17. That is, the value extrapolated from particle fluctuations is found between the values calculated from volume fluctuations and the ones calculated using equation (2.7). This suggests that the value calculated from volume fluctuations does not represent the value in the thermodynamic limit (which is for $V \rightarrow \infty$) but instead gives κ_T for the finite-sized volume of total simulation box. The method involving equation (2.7) is a more well known, established method, known to produce accurate values of the isothermal compressibility [30]. These results are therefore regarded as the reference value of the isothermal compressibility for the LJ systems. However, as already mentioned, it is possible that they are not thermodynamically consistent with the results calculated from the RDF and the particle fluctuations, due to the tail corrections.

Figures 5.24-5.25 show the partial volumes calculated from equation (2.22) of components 1 and 2 respectively. It should be noted that these are presented in reduced LJ units, which means that they are given per particle instead of per mole. They are not proper partial molar volumes, and is therefore instead referred to as partial volumes. For the ideal mixture, the partial volumes of both components should be equal to the volume per particle of the total system [2]. All methods show good overlap, but the closest ones are the results obtained from integrating the RDF, and the results of the fluctuation calculations using random positioned subvolumes. The magnitude of the standard deviations reflect those observed in figure 5.21, meaning that the ΔG_{12} expression has a large influence on the partial volumes. The individual KB integrals for like particles are also included in the expression, but these are multiplied with their associated concentration. The result is that the large standard deviations the KB integrals for like particles, observed at low mole fractions, does not have a large effect on the partial molar volumes.

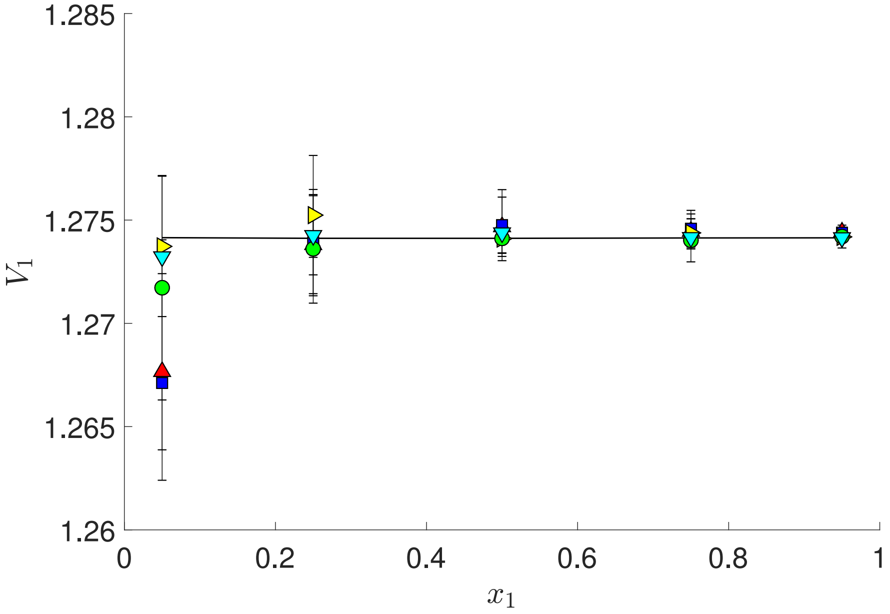

Figure 5.24: V_1 for the ideal LJ mixture, calculated from KB integrals combined in equation (2.22). Black line represents volume per particle for the total system. The error bars represent two standard deviations.

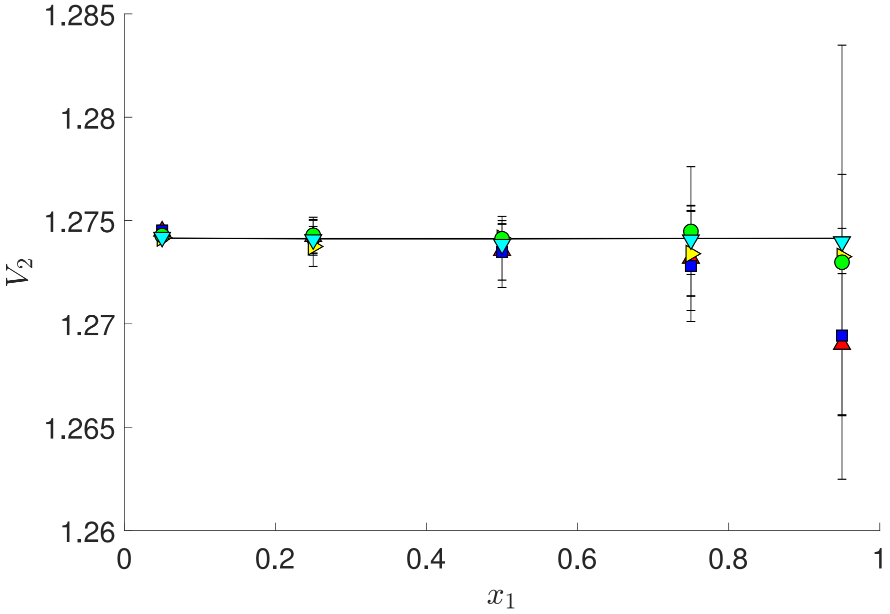

Figure 5.25: V_2 for the ideal LJ mixture, calculated from KB integrals combined in equation (2.22). Black line represents volume per particle for the total system

5.3.2 Real Lennard-Jones Mixture

All of the above introduced properties can also be calculated for the real LJ mixture. No theoretical prediction exists for comparison, but the isothermal compressibility can still be calculated from the volume fluctuations by equation (2.13), and by solving equation (2.7) numerically. Figure 5.26 show a similar trend to the one observed for ideal mixture, but the differences are much larger for low mole fractions of component 1. This reflects the behaviour discussed for this system in section 5.2.2, where very large particle fluctuations and volume fluctuations were observed. The rapid increase in κ_T is another indication that the system gets closer to its critical point as the mole fraction of component 2 increases. As stated in section 3.2, the fluctuation calculations will not work for systems close to the critical point, which could explain the large differences from the reference value.

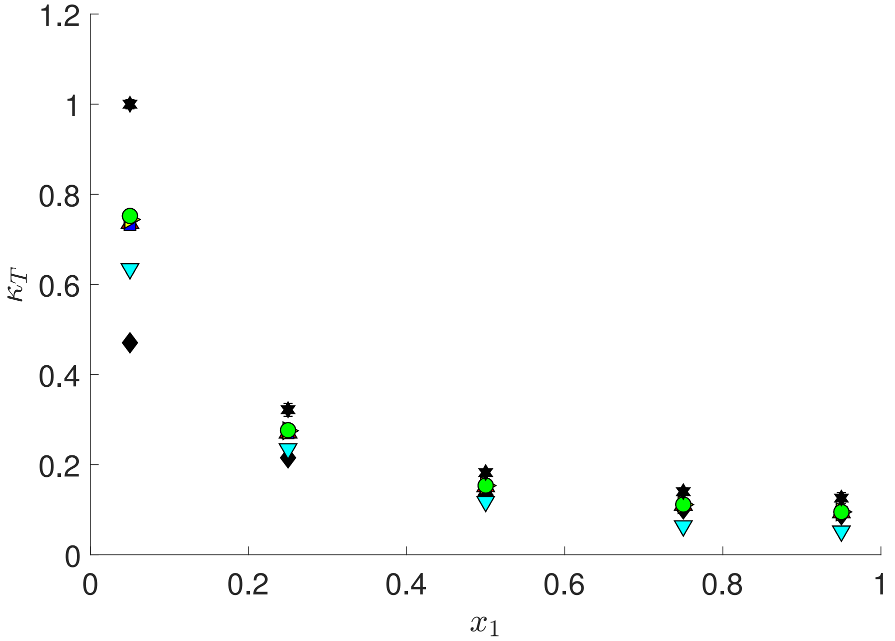

Figure 5.26: κ_T for the real LJ mixture, calculated from KB integrals combined in equation (2.21). Calculated from volume fluctuations using equation (2.13), and by solving (2.7) numerically. The error bars represent two standard deviations.

Figures 5.28-5.29 show the partial molar volumes calculated from equation (2.22) of components 1 and 2 respectively. In contrast to the equivalent results of the ideal LJ mixture, the results for the real LJ mixture show significant variations with mole fraction. This is expected for the real LJ mixture, since the total volume of the simulation box is not constant as a function of mole fraction, and the interparticle forces are different. As stated in section 2.1, the most well known and established method for computation of partial molar properties is from data of the molar property as a function of mole fraction. For the LJ systems, which has all their properties expressed in reduced units, the volume per particle V/N is instead calculated. By finding the slope of this curve, the partial molar volumes can be calculated from

$$\left(\frac{\partial V}{\partial N_1}\right)_{T,p,N_2} = \frac{V}{N} + (1 - x_1)\left(\frac{\partial\left(\frac{V}{N}\right)}{\partial x_1}\right)_{T,p} \tag{5.3}$$

$$\left(\frac{\partial V}{\partial N_2}\right)_{T,p,N_1} = \frac{V}{N} - x_1\left(\frac{\partial\left(\frac{V}{N}\right)}{\partial x_1}\right)_{T,p} \tag{5.4}$$

Equations (5.3)-(5.4) were solved using the `polyfit` function in MATLAB [74] to fit a 4th degree polynomial to V/N. Since only seven data points are used for curve fitting, it will not fit the data points perfectly, and some artificial curvature is introduced, as can be seen in figure 5.27. The results of this method, represented by the solid lines in figures 5.28-5.29, should therefore by no means be regarded as the correct result for the partial molar volumes, but they give an idea of how the trends should look. Both figures 5.28-5.29 show that the results calculated from particle fluctuations give partial volumes closer to those obtained by using equations (5.3)-(5.4).

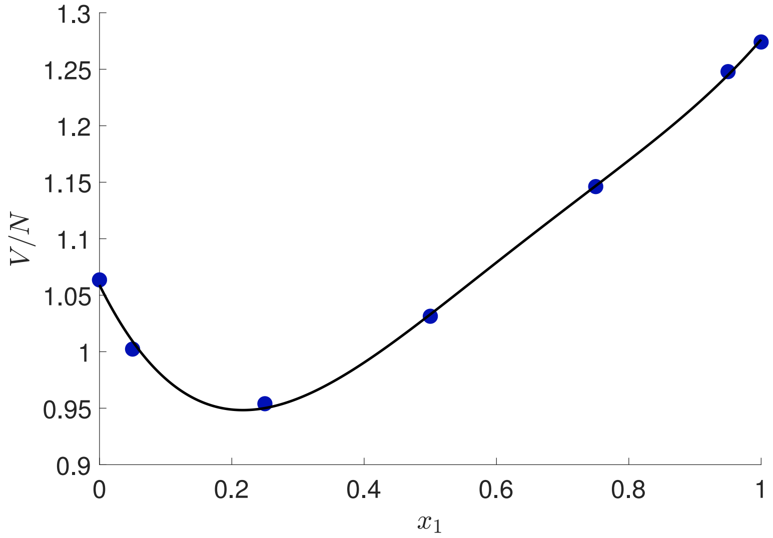

Figure 5.27: Volume per particle for the real LJ mixture, as a function of mole fraction. The lines solid show the 4th degree polynomial fitted to the data points using the `polyfit` function in MATLAB [74].

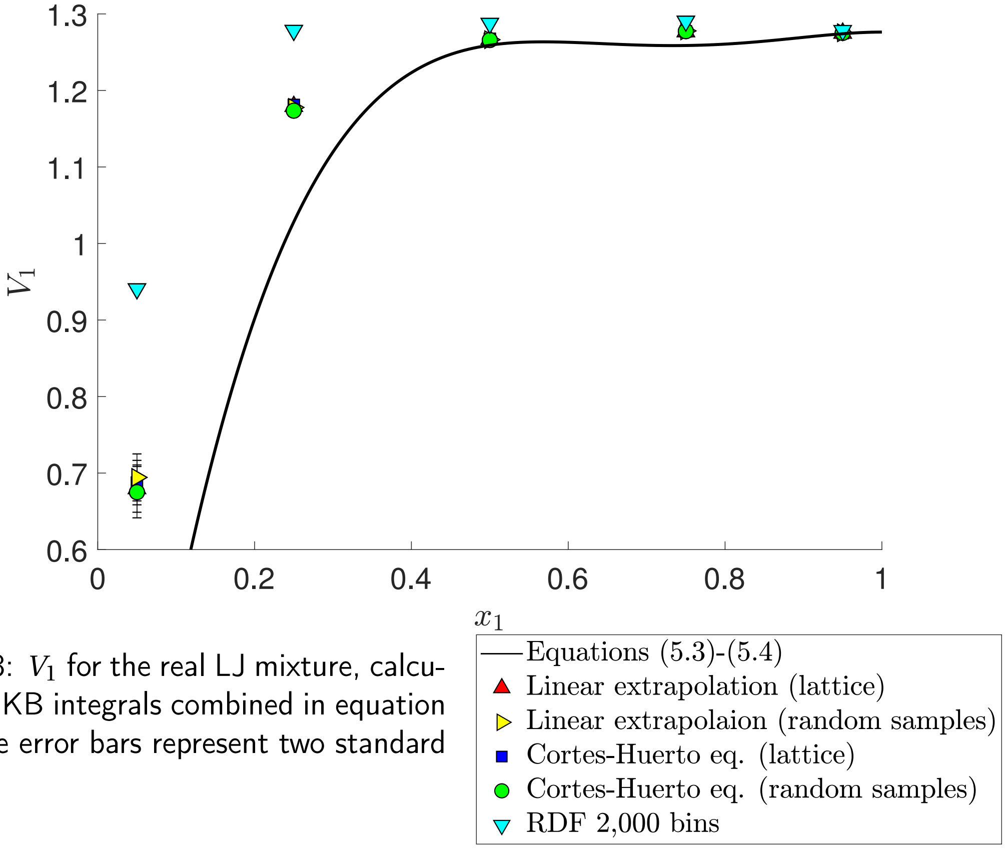

Figure 5.28: V_1 for the real LJ mixture, calculated from KB integrals combined in equation (2.22). The error bars represent two standard deviations.

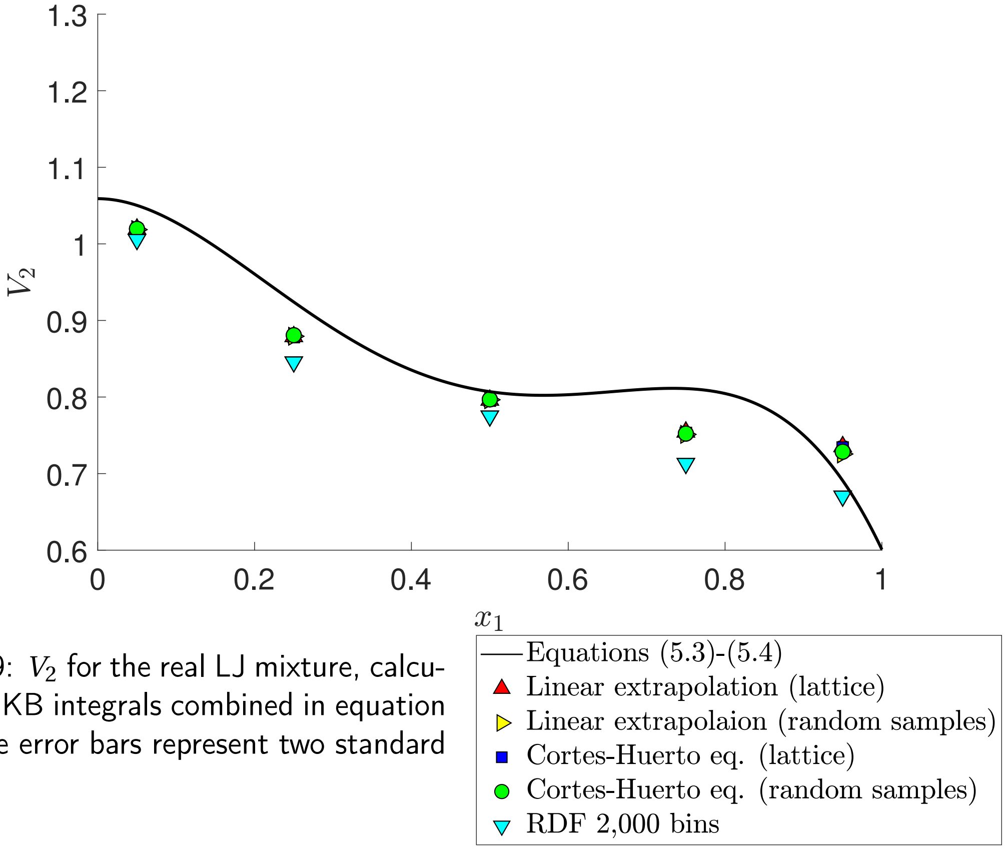

Figure 5.29: V_2 for the real LJ mixture, calculated from KB integrals combined in equation (2.22). The error bars represent two standard deviations.

5.3.3 Evaluation of Methods

Since all results for the two LJ mixtures are presented in LJ units, the absolute values of these results themselves do not give much information about real life properties. The comparison and analysis of the different methods is therefore in this section presented as relative to theoretical values, when these are available, or results of a method that is believed to provide accurate results. Figure 5.30 shows the difference between the theoretical value of $(\partial\mu_1/\partial x_1)_{T,p}$ calculated from equation (2.6), and the ones obtained by using KB integrals and equation (2.20). This figure shows that the method using the integration of the RDF clearly gives the lowest percentage difference, which is below 1% for all mole fractions. The fluctuation method that gives the lowest percentage difference is the one using random sampled subvolumes and linear scaling, which is below 2% for all mole fractions.

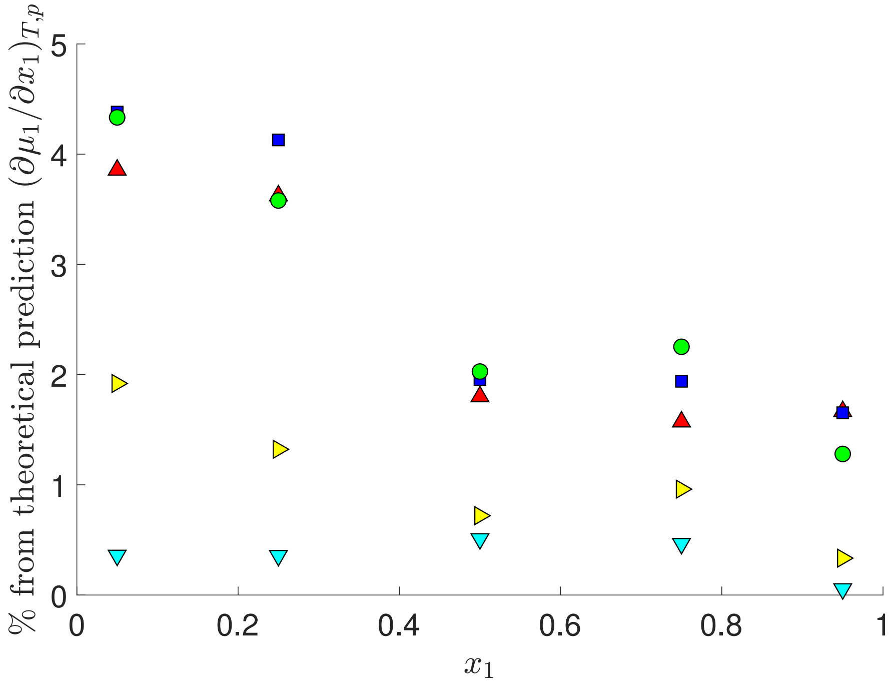

Figure 5.30: Percentage difference between $(\partial\mu_1/\partial x_1)_{T,p}$ calculated from (2.20) and theoretical prediction given by equation (2.6) for the ideal LJ mixture.

The same percentage difference can be calculated for the isothermal compressibility for both the ideal and the real LJ mixture. The results calculated using equation (2.7) are normally expected to give the most correct results of this property. As already mentioned, it is possible that these reference values are not thermodynamically consistent with the results calculated from the RDF and the particle fluctuations, due to the tail corrections.

However, they are the only values available for comparison, and were therefore used to compute the differences presented in figures 5.31-5.32 for the ideal and real LJ mixtures respectively. In contrast to the trends observed for $(\partial\mu_1/\partial x_1)_{T,p}$, the result obtained from RDF now show up to ~ 23% difference from the reference value, while the fluctuation methods are all giving result differentiating by $8-12\%$. As also previously observed, the fluctuation methods using random subvolumes and linear scaling gives the smallest relative difference from the reference value, at about 8% for all mole fractions.

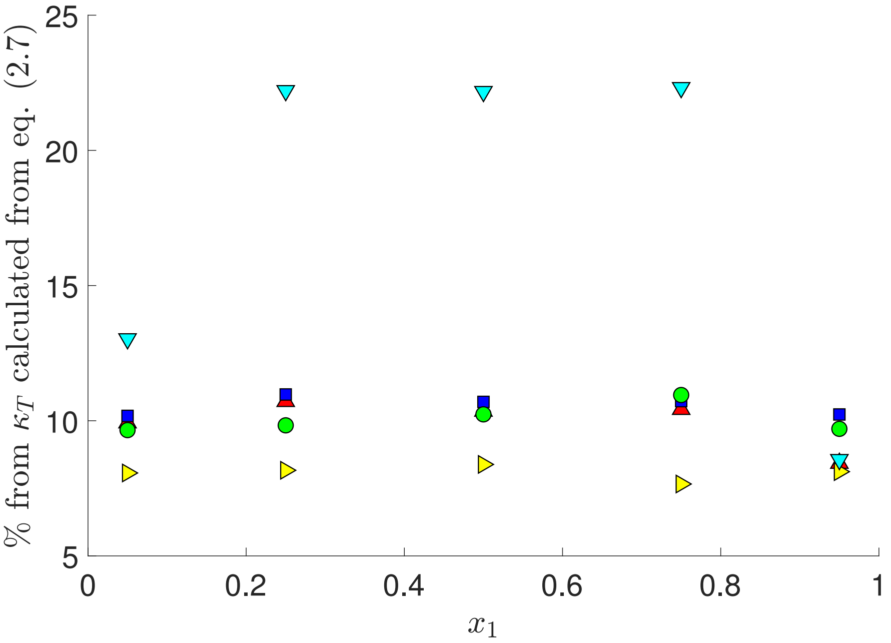

Figure 5.31: Percentage difference between κ_T calculated from (2.21) and calculated by numerical differentiation of equation (2.7) for the ideal LJ mixture.

These trends are again different for real LJ mixture. Both methods show very large differences for $x_1 = 0.05$, which are up to 60% for fluctuation methods and 35% for the result calculated from integration of the RDF. For the fluctuation methods, this difference becomes lower as x_1 increases, and is around 10% for $x_1 = 0.95$, which is similar to the percentage differences observed for the ideal mixture. From figure 5.26, it can be seen that the absolute difference for the results obtained using the RDF is the same for all mole fractions, except $x_1 = 0.05$. This means that the percentage value is larger for high mole fractions of component 1 only because the absolute values become smaller here. Hence, the percentage differences obtained by integration of RDF presented in figure 5.32 do not show how the accuracy of the RDF method vary with mole fraction. However, it does give information about the difference in accuracy between this method and the fluctuation methods, where it can be seen that RDF provides a much more accurate result for low mole fractions of component 1, while the fluctuation methods provide better results for high mole fractions of component 1. This suggests that for computation of isothermal compressibility, the fluctuation method is a better alternative as long as the system is sufficiently far from the critical point.

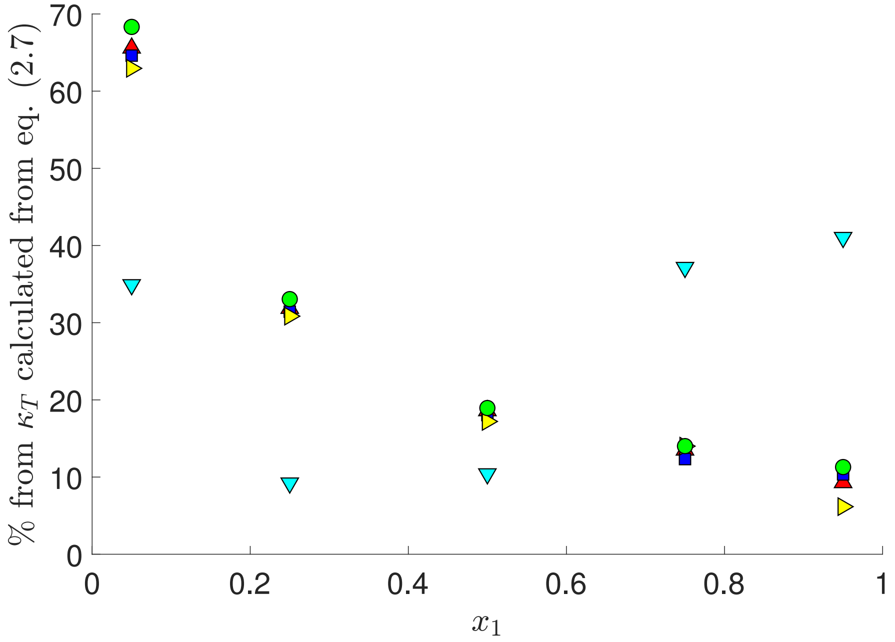

Figure 5.32: Percentage difference between κ_T calculated from (2.21) and calculated by numerical differentiation of equation (2.7) for the real LJ mixture.

The overall accuracy can be analyzed by the factor k, which is calculated as explained in appendix A. The closer to zero this value is, the better is the total accuracy of the calculated results. To check the total accuracy of the calculated partial volumes, the volume per particle was calculated from $V/N = x_1 V_1 + x_2 V_2$, and compared to the actual volume per particle, calculated from the size of the total simulation box. Table 5.1 presents the k-values for $(\partial\mu_1/\partial x_1)_{T,p}$, κ_T and V/N for the ideal LJ mixture, while table 5.2 presents the k-values of κ_T and V/N for the real LJ mixture. The k-values of $(\partial\mu_1/\partial x_1)_{T,p}$ and κ_T for both LJ mixtures reflect the behaviour presented in 5.30-5.32. The property that does show surprising k-values is the volume per particle. These values are so small that they hardly differ from zero, which should mean that all methods work equally well. This was unexpected since differences in these values were observed in figures 5.28-5.29. The volume per particle is clearly not much affected by the values of the component with the lowest mole fraction, since the final result is the same for all methods. The overall accuracy of the properties $(\partial\mu_1/\partial x_1)_{T,p}$ and κ_T are also better for the NVT ensemble.

Table 5.1: k-values for ideal Lennard-Jones mixture, comparing overall accuracy of values calculated from KB integrals to the true values of the same property. Values of k were calculated as explained in appendix A, where equation (2.6) represents the true value for $(\partial\mu_1/\partial x_1)_{T,p}$, the results of solving equation (2.7) numerically represents the true value of κ_T, and the volume per particle is calculated using $V/N = x_1V_1 + x_2V_2$. Significant numbers were determined by the standard deviations of k.

Method NpT	$k(\partial\mu_1/\partial x_1)_{T,p}$	$k(\kappa_T)\cdot 10^5$	$k(V/N)\cdot 10^{31}$
Linear extrapolation (lattice)	0.11	7.5	1.6
Linear extrapolation (random position)	0.027	4.9	2.0
Cortes-Huerto eq. (lattice)	0.15	8.4	1.8
Cortes-Huerto eq. (random position)	0.14	7.6	2.0
RDF (2000 bins)	0.0021	25.8	1.8
NVT			
Linear extrapolation (lattice)	0.09	7.0	2.6
Linear extrapolation (random position)	0.02	1.65	3
Cortes-Huerto eq. (lattice)	0.07	4.5	1.0
Cortes-Huerto eq. (random position)	0.08	3.6	1.6
RDF (2000 bins)	0.0021	25.8	1.8

Table 5.2: k-values for real Lennard-Jones mixture, comparing overall accuracy of values calculated from KB integrals to the true values of the same property. Values of k were calculated as explained in appendix A, where the results of solving equation (2.7) numerically represents the true value of κ_T, and the volume per particle are calculated using $V/N = x_1V_1 + x_2V_2$. Significant numbers were determined by the standard deviations of k.

Method NpT	$k(\kappa_T)\cdot 10^2$	$k(V/N)\cdot 10^{31}$
Linear extrapolation (lattice)	2.02	2.4
Linear extrapolation (random position)	1.96	2.5
Cortes-Huerto eq. (lattice)	2.19	2.4
Cortes-Huerto eq. (random position)	1.86	1.5
RDF (2000 bins)	1.24	1.9
NVT		
Linear extrapolation (lattice)	1.574	2.2
Linear extrapolation (random position)	1.484	1.9
Cortes-Huerto eq. (lattice)	1.54	2.0
Cortes-Huerto eq. (random position)	1.57	2.5
RDF (2000 bins)	1.24	1.9

There are two important factors to evaluate when comparing the output values of different methods. These are accuracy (meaning that the results are close to reference values) and precision (which means low standard deviations). Between the fluctuation methods, the method involving randomly positioned subvolumes and linear scaling was by far providing the most precise precise results. The standard deviations of this method were of approximately the same magnitude as those obtained by integration of RDF. The accuracy, on the other hand, must be evaluated from the figures 5.30-5.32 and tables 5.1-5.2 combined. From the values presented in tables 5.1-5.2, there is no clear method that is consistently providing the most accurate results, neither between the integration of the RDF and the fluctuations methods, nor between the fluctuation methods themselves. However, the differences observed in figures 5.31-5.32 show that the isothermal compressibility is much more accurately calculated from the fluctuation methods, provided that the system is sufficiently far from its critical point. Among the fluctuation methods, the linear extrapolation and randomly positioned subvolumes is more often than the remaining methods providing results closer to the reference values

Another thing that is of importance when evaluating the effectiveness of the different methods, is the computational effort and the amount of postprocessing that has to be done to get the final results. Initialization, equilibriation and production runs made up 600 CPU hours per composition. For a total simulation box in the NVT ensemble, the instantaneous number of particles in the lattice subvolumes were tracked by LAMMPS during the simulation. This was therefore the most cost effective method, since it only required one additional CPU hour for postprocessing of the LAMMPS output files. The RDF calculations for both ensembles, and the instantaneous particle numbers for the lattice in the NpT ensemble, were calculated by the `rerun` command in LAMMPS, which required an additional 400 CPU hours. The Python code calculating particle fluctuations in randomly positioned subvolumes spent 125 CPU hours. LAMMPS has a command for the purpose of computing the RDF, and there are several other postprocessing tools available [80] [81]. For fluctuation calculations, no such tool exits, and the method must be implemented by oneself. The integration of RDF is therefore the most available and established method. The computational time spent on calculations of fluctuations of the randomly positioned subvolumes could probably have been shortened by using a programming language that is more efficient in performing computations on large amounts of data. This could also be used to compute RDF, which would require less computational time, and the problems regarding the behaviour of the calculated RDF, discussed in section 3.3.1, could probably have been avoided.

One last question that should be answered, is whether the methods are easy to use. The largest problem when using the fluctuation methods is how sensitive these are to the extrapolation, as shown in section 5.2.2. It can be difficult to visually determine how close to $\Omega/V \to 0$ the linear region expands, especially in the cases where it is gradually changing. Cortes-Huerto *et al.* suggested restrictions that should be used to determine the location of the linear region, but as explained in section 5.2.1, this was not possible to apply for the LJ systems studied here. The linear region of the KB integrals calculated from RDF also needs to be determined visually, but this one is normally more distinct than those of the fluctuation methods.

5.4 Methanol-Water Mixture

Even though differences could be observed in the values of $(\partial\mu_1/\partial x_1)_{T,p}$, κ_T and V_i calculated from KB integrals of the LJ mixtures, the KB integrals themselves were mostly within two standard deviations of those calculated from other methods. To investigate if larger differences arise for more complex mixtures, all the above evaluated methods were also tested for the methanol-water mixture. In section 2.2 it was determined that the particle fluctuations of the largest of the embedded systems in a NpT-simulation box were affected by the fluctuating volume. This made it harder to determine the extent of the linear region, and is probably also why tables 5.1-5.2 show that the NVT simulations provided better overall accuracy for all properties investigated. The simulations of the methanol-water mixture were therefore performed in the NVT ensemble.

By following the procedure suggested in section 5.1.1, 3,000 bins was determined to be sufficient for the RDF calculations. The behavior of the RDF at large R, described in section 5.1.3, was also observed for the methanol-water mixture. This means that it is even more likely that the computation of RDF by LAMMPS caused this difference. Finite size corrections were therefore not applied for the RDFs of methanol-water mixture either. The way the sampling was performed for LJ mixtures was not optimal, since it resulted in a large number of data points in regions not available for extrapolation. For the methanol-water mixture, a larger number of different sizes was therefore investigated. In order to optimize number of data points available for curve fitting, the sizes of the subvolumes were also constrained to those that were expected to produce values in the linear region. The maximum radius of spherical subvolumes used for the methanol-water mixtures was therefore $R = 20$ Å, which was based on where the linear region was observed for the LJ mixtures. This extended to $\Omega/V \approx 0.75$ for most systems, which corresponds to a subsystem of radius ~ 20Å. The linear region is not guaranteed to be found for the same sized subvolumes for all systems, but the LJ result does give an idea of its location for other systems. As later shown in figures 5.34-5.35, the subvolumes chosen here sufficiently covered the linear region for the methanol-water mixture. First, the KB integrals in the thermodynamic limit, calculated from all different methods are analyzed as function of mole fraction.

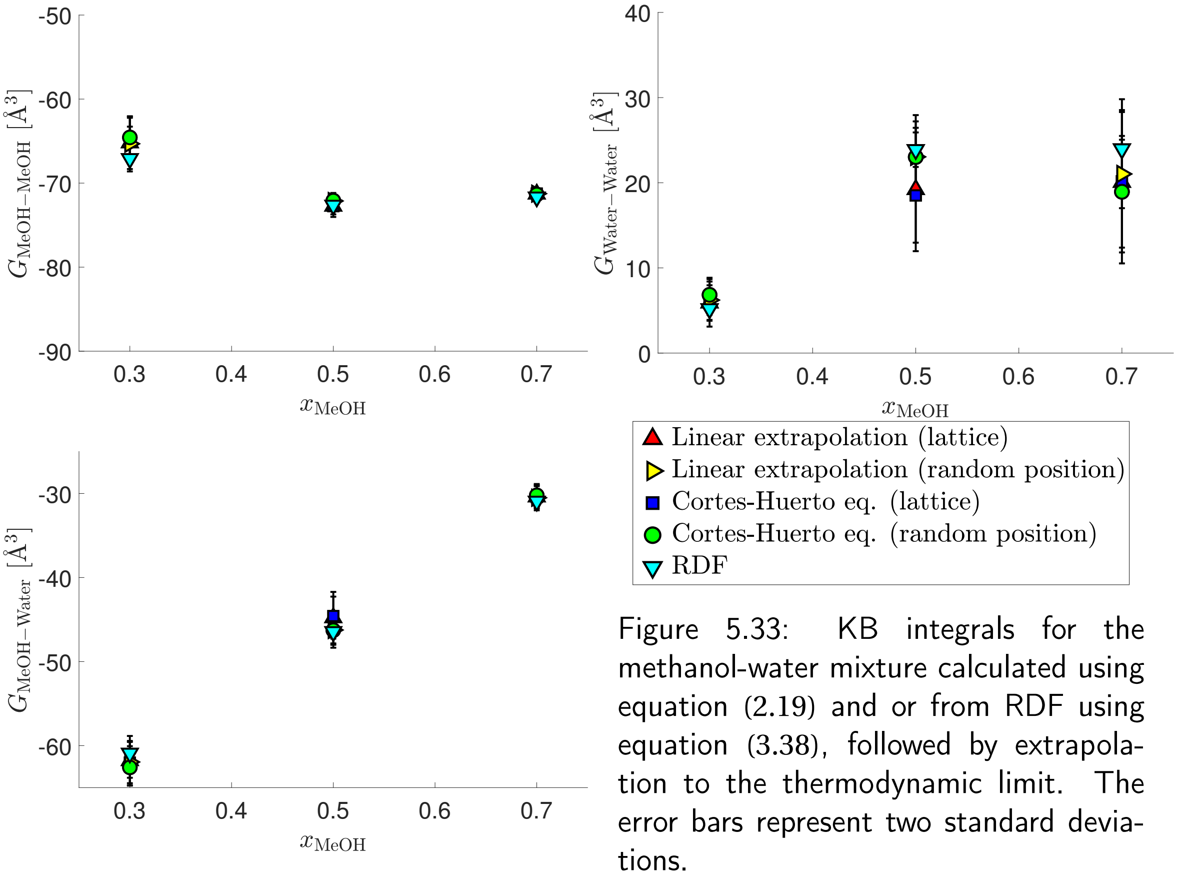

Figure 5.33: KB integrals for the methanol-water mixture calculated using equation (2.19) and or from RDF using equation (3.38), followed by extrapolation to the thermodynamic limit. The error bars represent two standard deviations.

Figure 5.33 show the thermodynamic limit value of KB integrals calculated from the fluctuation methods and by integration of RDF. All data points are within two standard deviations of the ones obtained by other methods, which means they still did not give significantly different results. The methods used in this work also provided results that were close to those reported by Weerasinghe and Smith [9], in the original paper that first introduced the methanol force field. See appendix B for figures of excess coordination number that can be directly compared with the result reported by Weerasinghe and Smith [9].

Similarly to the LJ mixture, the results obtained by lattice partitioning show the largest standard deviations, at around twice the magnitude of those obtained from randomly positioned subvolumes. The linear scaling is still the most precise fluctuation method, with standard deviations comparable to those obtained by integration of RDF. The y-axis of the plots in figure 5.33 do not cover the same range, but the data points are displayed for ranges of the same size. This means that how the properties vary with mole fraction, and the magnitude of the standard deviations can be directly compared. The KB integrals representing the water-water interactions are consistently positive, while the other two are negative. The one that varies the most as a function of mole fraction is $G_{\text{MeOH−Water}}$, while the standard deviations of $G_{\text{Water−Water}}$ are larger than for the other two. For the composition with highest mole fraction of water ($x_{\text{MeOH}} = 0.30$), the magnitude of the standard deviation of $G_{\text{Water−Water}}$ is actually comparable to the one of $G_{\text{MeOH−MeOH}}$. This is different from what was observed for the LJ mixtures, where the component in excess always showed better statistics, both for G_{11} and for G_{22}. However, the results presented here are similar to the results obtained by Weerasinghe and Smith [9], which means that they could have been caused by precision of the force field.

As for the LJ mixtures, the number density was constant for all subvolume sizes. Hence, the KB integrals could be calculated directly using equation (5.1). Figures 5.34-5.35 show the procedure of the curve fitting using the linear extrapolation and Cortes-Huerto scaling equation respectively, for the methanol-water mixture at $x_{\text{MeOH}} = 0.50$.

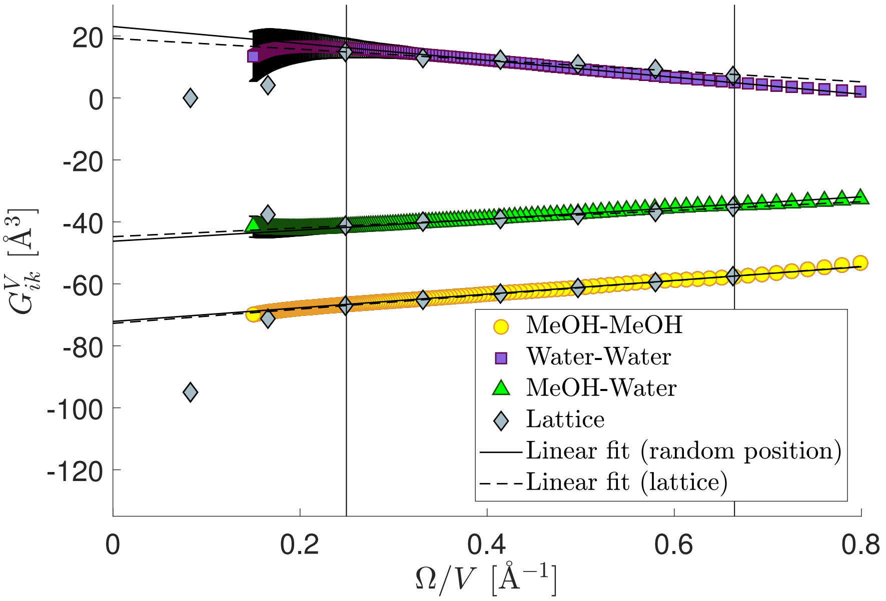

Figure 5.34: G_{ik}^V for the methanol-water mixture at $x_{\text{MeOH}} = 0.50$ calculated directly from equation (5.1), as a function of the surface to volume ratio. Results of both subvolume methods (random positioned and lattice partitioning) are included. Thermodynamic limit value G_{ik} was obtained from linear fit to region marked by vertical lines for both subvolume methods.

In contrast to the LJ mixtures, the lattice partitioning and randomly positioned subvolumes are now giving different results in the linear region in figure 5.34. The difference is probably due to the correlated fluctuations in the lattice subvolumes, as discussed earlier. This correlation could be larger for the methanol-water system because it consists of more complex components than the LJ system. It is also possible that fluctuations simply are more correlated because the lattice subvolumes are smaller for this system, due to the smaller size of the total simulation box. The length of the total simulation box varies with composition, but its average value for the methanol-water mixture was ~ 70 Å, while it was ~ 145 Å for the LJ systems.

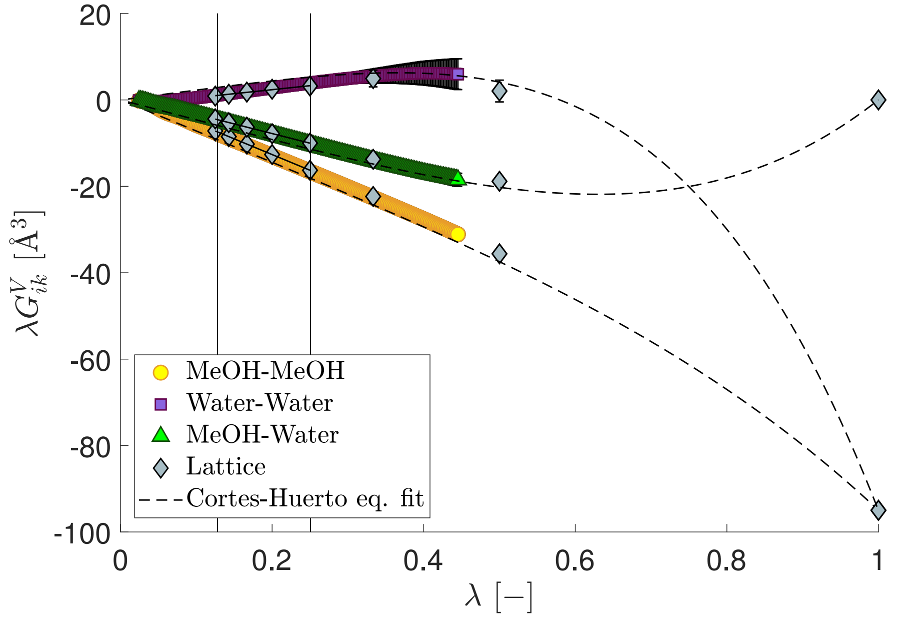

Figure 5.35: λG_{ik}^{V} for methanol-water mixture at $x_1 = 0.50$ calculated from equation (3.42), as a function of the parameter λ. Results of both subvolume methods (random positioned and lattice partitioning) are included. Thermodynamic limit value G_{ik} was obtained as the slope of linear fit to the region marked by vertical lines.

Recall that, the restriction for subvolume sizes introduced by Cortes-Huerto *et al.* [20], stated the smallest subvolume should not be smaller than V_ξ (which corresponds to a spherical subvolume of radius equal to ξ) and that the largest subvolume used for curve fitting should correspond to $\lambda = 0.3$. The RDF for the methanol-water system at $x_{\text{MeOH}} = 0.50$ in figure 5.36, shows that this system has a structural correlation length of $\xi \approx 12$ Å, which corresponds to $\lambda \approx 0.27$, and a surface to volume ratio of $\Omega/V = 0.25$ for spherical subvolumes. Figure 5.34 shows that the fluctuations calculated for volumes much larger than this, clearly were affected by the finite size of the simulation box, and should not be used for extrapolation. This means that the lower limit of the restriction introduced by

Cortes-Huerto *et al.* [20] does not accurately describe the location of the linear region of this system either. A linear region still appears, despite the fact that the assumptions these restrictions arise from were not fulfilled (see derivation of equation (3.47) in section 3.4). It is therefore instead believed to arise due to the thermodynamics of small systems, and the proportionality between their extensive properties and the inverse systems size. As already mentioned in section 5.2.1, this shows, that for finite size KB integrals, the linear scaling and the Cortes-Huerto equation are essentially the same.

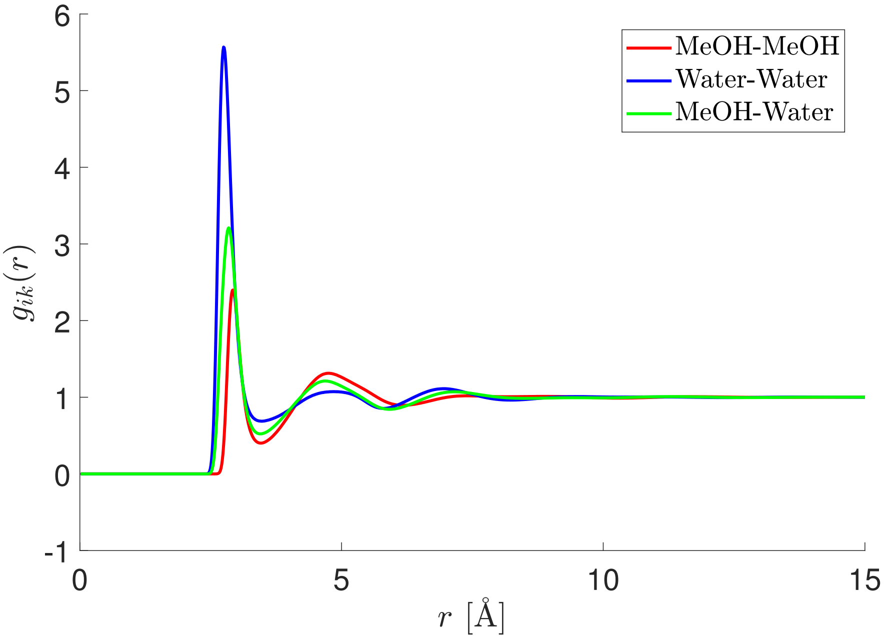

Figure 5.36: RDF of the methanol-water mixture at $x_{\text{MeOH}} = 0.50$.

The isothermal compressibility and the partial molar volumes were calculated from combinations of KB integrals by equations (2.21) and (2.22) respectively. The isothermal compressibility was in addition computed by solving equation (2.7) numerically, and presented in figure 5.37. The results computed from KB integrals for the compositions $x_{\text{MeOH}} = 0.50$ and $x_{\text{MeOH}} = 0.70$ are not significantly different from the results computed by equation (2.7). However, the results at the lowest mole fraction show larger differences. As introduced in section 2.1.2, an approximate value of κ_T for mixtures can be calculated from

$$\kappa_T = \phi_i \kappa^0_{T,i} + \phi_k \kappa^0_{T,j} \tag{5.5}$$

where ϕ_i is the volume fraction of component i, and $\kappa^0_{T,i}$ is the isothermal compressibility of a pure system of component i. Since the real life values of isothermal compressibility for pure methanol and water are $\kappa^0_{T,\text{MeOH}} = 12.3 \cdot 10^{-5}\text{atm}^{-1}$ $\kappa^0_{T,\text{Water}} = 4.6 \cdot 10^{-5}\text{atm}^{-1}$ [82], it is more likely that the values calculated by KB integrals represent the real life values for methanol-water mixtures at this compositions. This argument is also substantiated by the fact that the force field was developed for the purpose of providing accurate estimates of the KB integrals of the system, and not necessary other properties of a true methanol-water mixture. It is therefore believed that the isothermal compressibility computed by solving equation (2.7) is more accurate for this specific force field, but that the KB integrals better represent the true values of a real life mixture of methanol and water.

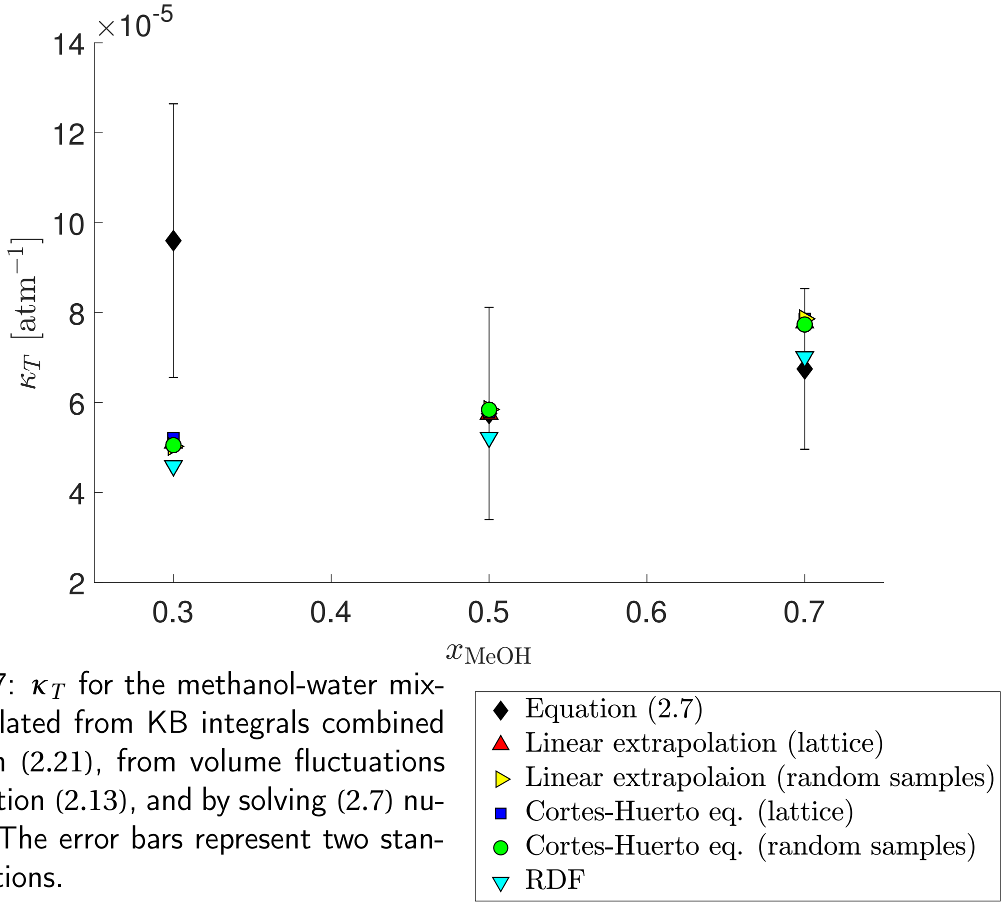

Figure 5.37: κ_T for the methanol-water mixture, calculated from KB integrals combined in equation (2.21), from volume fluctuations using equation (2.13), and by solving (2.7) numerically. The error bars represent two standard deviations.

Both the values of κ_T presented in figure 5.37 and the partial molar volumes in figures 5.38-5.39 closely resemble the results reported by Weerasinghe and Smith [9] and experimental data [83] [84]. Through the use of KB integrals, a simple molecule model is therefore able to accurately reproduce experimental values just as well as more complex and polarized force fields [85] [86]. However, the experimental data can not be used to evaluate the accuracy of the different methods, since the exact values of the KB integrals for the force field are not necessarily equal to the exact values of the KB integrals for a real life methanol-water mixture.

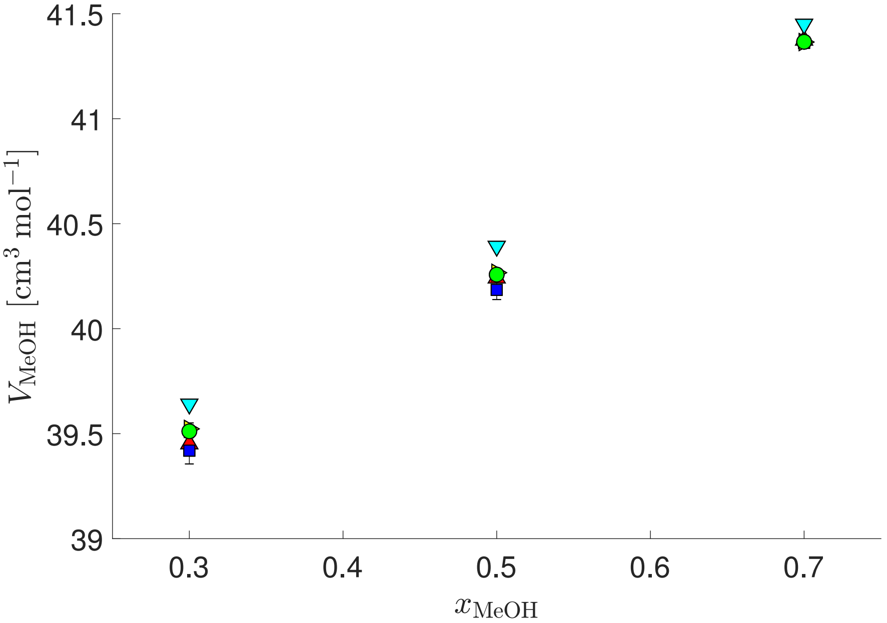

Figure 5.38: V_{MeOH} for the methanol-water mixture, calculated from KB integrals combined in equation (2.22). The error bars represent two standard deviations.

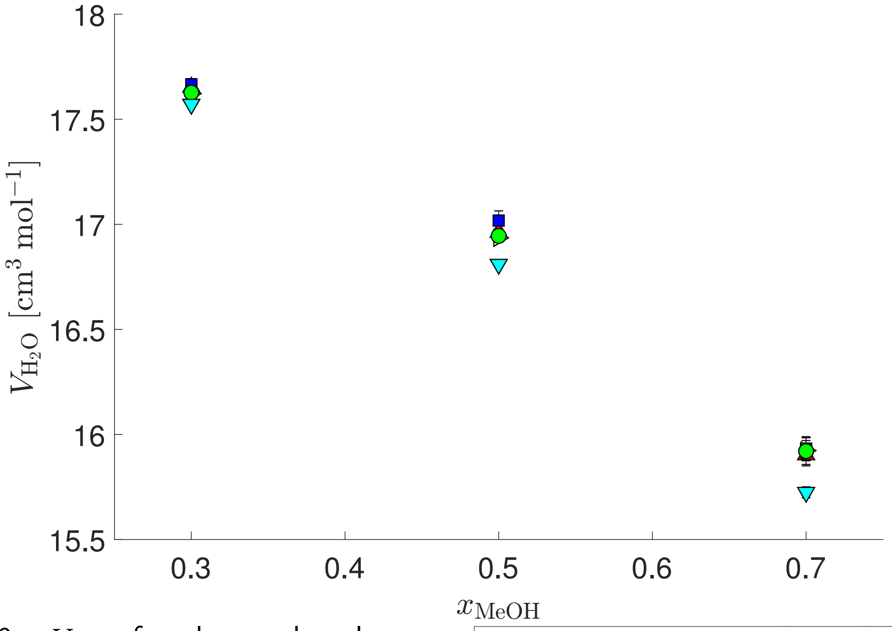

Figure 5.39: $V_{\mathrm{H_2O}}$ for the methanol-water mixture, calculated from KB integrals combined in equation (2.22). The error bars represent two standard deviations.

The reference values for κ_T showed so large standard deviations that comparison to the values calculated from KB integrals does not provide much information about their accuracy. The most accurate method for the methanol-water system could therefore not be determined by studying only these three compositions. The values of κ_T and V_i calculated from the fluctuation methods barely differ from each other, but the difference between these, and the ones obtained from integration of RDF is $\sim$ 10% for κ_T, and $\sim$ 1% for the partial molar volumes. These percentage values are approximately the same as observed for the LJ mixtures. This suggests that the relative errors presented for the LJ mixtures in section 5.3.3 are representative for the different methods, and that approximately the same relative errors can be expected for other mixtures. However, the behaviour of only the two properties κ_T and V_i do not provide enough information to make a final conclusion.

6 Conclusion and Perspectives

A systematic comparison of the different methods available for computation of KB integrals in the thermodynamic limit was conducted. The methods were represented by two categories: integration of RDF and particle fluctuations in grand canonical systems. Both involved extracting the KB integrals in the thermodynamic limit from the ones computed for different-sized finite volumes. The variations within each category, and their impact on the KB integrals in the thermodynamic limit were thoroughly analyzed. Consequently, the most efficient method could be determined by considering four factors: accuracy, precision, computational time and availability.

The method involving integration of the RDF of the system showed dependence on the following: 1) number of shells used for computation of the RDF 2) how often the simulation trajectory was sampled and 3) applying finite size corrections to the RDF. The common factor of the first two was found to be the good resolution of the first peak of the RDF. This was achieved by sampling the simulation trajectory every 100 step and using a large enough number of bins. Sufficient number of bins was found when the integral under the total curve no longer changed upon increasing the number of bins. At this point, the trapezoid method provides precise integration. The finite size correction proposed by Ganguly *et al.* [8] could not correct for the drifting asymptotes of the finite-size KB integrals. However, the source of this is not believed to be the finite size correction, but rather how the RDF is computed in LAMMPS. These RDFs showed unexpected behaviour, that became evident at large *R*, where it did not show the expected depletion of particles for a closed system. The finite size correction, which assumes that this depletion is present, is therefore believed to have enlarged this behaviour.

The factors investigated for the fluctuation based methods were: 1) definition of subvolume and 2) scaling equation used to obtain estimates of the KB integrals in the thermodynamic limit. The different subvolume types investigated were the one normally used by SSM, which is placing the subvolumes at random positions inside the simulation box, and the superimposed lattice proposed by Galata *et al.* [23], where each lattice cell is regarded as a subvolume. For all systems studied here, the randomly positioned subvolumes resulted in better precision than the ones defined by lattice partitioning. This is related to the quality of the curve fitting used to extract the KB integrals in the thermodynamic limit. The different subvolume sizes available from lattice partitioning is constrained by the size of the total simulation box, which resulted in few data points, and poor curve fitting. The statistics of each data point calculated from the lattice subvolumes are also affected by the correlated fluctuations in neighbouring boxes. The method using randomly positioned subvolumes is therefore more flexible, and can achieve better precision with less computational effort.

The different techniques used to obtain the KB integrals in the thermodynamic limit from fluctuation calculations were the linear extrapolation, as normally used in SSM, and a recently proposed equation by Cortes-Huerto *et al.* [20]. Neither method could be determined to outperform the other in both accuracy and precision. However, it was found that the uncertainties in KB integrals in the thermodynamic limit obtained by both methods were caused by the uncertainties of the curve fitting. Direct extrapolation of the finite volume KB integrals therefore provide better precision than by first extrapolating the proper extensive properties individually. It is important to keep in mind that this only can be performed for quantities that can be expressed as a linear combination of the proper extensive properties. Also, the expected behaviour these extensive properties is better known for both small and large embedded volumes, meaning that discrepancies are more easily revealed. Both methods suffer from difficulty in determining the linear region, which in some cases may have affected the extracted value of the KB integral in the thermodynamic limit. When calculating KB integrals, the scaling methods were therefore determined to work equally well.

Comparison of fluctuations obtained from subvolumes in a total simulation box in the NpT and NVT ensemble showed that the fluctuating volume of the NpT ensemble introduced unwanted contributions to the fluctuations in number of particles. Only ensembles with constant volumes should therefore be used for sampling of subvolumes. Here, the RDF results show better accuracy since these did not depend on the ensemble of the reservoir. The ensemble of the reservoir was discovered to have the largest impact on the fluctuations in the systems close to the critical point, while the RDF provided better accuracy here as well. Integration of RDF is also the most available method, since several tools exist for its computation, while the fluctuation method must be implemented by oneself. However, the only way to obtain better statistics is by longer simulation time, while the fluctuation method can faster achieve better statistics by sampling a larger number of subvolumes per simulation snapshot. The accuracy of two methods depend on which property is calculated, but the errors introduced to the isothermal compressibility by the RDF method is much larger than the error introduced to the derivative of the chemical potential by the fluctuation method.

Considering all factors above, it was determined that the fluctuation method, using randomly positioned subvolumes is the most efficient way of obtaining KB integrals in the thermodynamic limit, provided that the system is sufficiently far from the critical point. The way it was implemented in this work was not optimal, and suggestions for improvement are therefore: 1) sampling subvolumes with linearly increasing reciprocal sphere radius $1/R$ in order to maximize the number of data points in the regions used to extract KB integrals in the thermodynamic limit or 2) take into account the finite size of the reservoir by using an equation designed for this purpose, and the fluctuations calculated for all different-sized subvolumes, meaning either the one proposed by Cortes-Huerto *et al.* [20] (equation 3.47) or the one proposed by Strøm *et al.* [49] (equation (3.32)).

The combination of linear extrapolation and randomly positioned subvolumes represent the procedure used by SSM. This method also has been showing promising potential in ability to study properties of small systems. This argument is based on the fact that it is able to accurately reproduce values in the thermodynamic limit from fluctuations in small systems. In previously published work, this has been showing good overlap [49] [51], but the results presented here show larger differences. The reason for this could be further studied by direct comparison of proper grand canonical fluctuations from Monte Carlo simulations. However, it is possible that the discrepancies only were caused by the long-range tail corrections, since it gives output values that are not thermodynamically consistent with the truncated LJ potential, and therefore probably not consistent with the particle fluctuations either. This should therefore be investigated by running some similar simulations without long-range tail corrections.

References

[1] Ben-Naim A. Chapter 2 - Molecular distribution functions. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 21–75.

[2] Helbæk M, Kjelstrup S. Kapittel 6 - Faselikevekter og egenskaper for løsninger og blandinger. In: Helbæk M, editor. Fysikalsk kjemi. 2nd ed. Fagbokforlaget Vigmostad & Bjørke AS; 2006. p. 237–324.

[3] Kirkwood JG, Buff FP. The Statistical Mechanical Theory of Solutions. J Chem Phys. 1951;19(6):774–777.

[4] Ben-Naim A. Chapter 4 - The Kirkwood-Buff theory of solutions. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 113–135.

[5] Chitra R, Smith PE. Molecular Association in Solution: A KirkwoodBuff Analysis of Sodium Chloride, Ammonium Sulfate, Guanidinium Chloride, Urea, and 2,2,2-Trifluoroethanol in Water. J Phys Chem B. 2002;106(6):1491–1500.

[6] Shulgin I, Ruckenstein E. KirkwoodBuff Integrals in Aqueous Alcohol Systems: Comparison between Thermodynamic Calculations and X-Ray Scattering Experiments. J Phys Chem B. 1999;103(13):2496–2503.

[7] Schnell SK, Englebienne P, Simon JM, Krüger P, Balaji SP, Kjelstrup S, et al. How to apply the Kirkwood–Buff theory to individual species in salt solutions. Chem Phys Lett. 2013;582:154 – 157.

[8] Ganguly P, van der Vegt NFA. Convergence of Sampling Kirkwood–Buff Integrals of Aqueous Solutions with Molecular Dynamics Simulations. J Chem Theory Comput. 2013;9(3):1347–1355.

[9] Weerasinghe S, Smith PE. A Kirkwood-Buff derived force field for methanol and aqueous methanol solutions. J Phys Chem B. 2005;109(31):15080–15086.

[10] Weerasinghe S, Smith PE. A KirkwoodBuff Derived Force Field for Mixtures of Urea and Water. J Phys Chem B. 2003;107(16):3891–3898.

[11] Vlugt TJH, Martin MG, Smit B, Siepmann JI, Krishna R. Improving the efficiency of the configurational-bias Monte Carlo algorithm. Molecular Physics. 1998;94(4):727–733.

[12] Dubbeldam D, Torres-Knoop A, Walton KS. On the inner workings of Monte Carlo codes. Molecular Simulation. 2013;39(14-15):1253–1292.

[13] Torres-Knoop A, Balaji SP, Vlugt TJH, Dubbeldam D. A Comparison of Advanced Monte Carlo Methods for Open Systems: CFCMC vs CBMC. J Chem Theory Comput. 2014;10(3):942–952.

[14] Torres-Knoop A, Poursaeidesfahani A, Vlugt TJH, Dubbeldam D. Behavior of the Enthalpy of Adsorption in Nanoporous Materials Close to Saturation Conditions. J Chem Theory Comput. 2017;13(7):3326–3339.

[15] Weerasinghe S, Smith PE. A Kirkwood-Buff derived force field for the simulation of aqueous guanidinium chloride solutions. J Chem Phys. 2004;121(5):2180–2186.

[16] Smith PE. Cosolvent Interactions with Biomolecules: Relating Computer Simulation Data to Experimental Thermodynamic Data. J Phys Chem B. 2004;108(48):18716–18724.

[17] Nichols JW, Moore SG, Wheeler DR. Improved implementation of Kirkwood-Buff solution theory in periodic molecular simulations. Phys Rev E. 2009 Nov;80:051203.

[18] Wedberg R, O'Connell JP, Peters GH, Abildskov J. Accurate Kirkwood–Buff integrals from molecular simulations. Molecular Simulation. 2010;36(15):1243–1252.

[19] Krüger P, Schnell SK, Bedeaux D, Kjelstrup S, Vlugt TJH, Simon JM. Kirkwood–Buff Integrals for Finite Volumes. J Phys Chem Lett. 2013;4(2):235–238.

[20] Cortes-Huerto R, Kremer K, Potestio R. Communication: Kirkwood-Buff integrals in the thermodynamic limit from small-sized molecular dynamics simulations. J Chem Phys. 2016;145(14):141103.

[21] Schnell SK, Liu X, Simon JM, Bardow A, Bedeaux D, Vlugt TJH, et al. Calculating Thermodynamic Properties from Fluctuations at Small Scales. J Phys Chem B. 2011;115(37):10911–10918.

[22] Milzetti J, Nayar D, van der Vegt NFA. Convergence of Kirkwood–Buff Integrals of Ideal and Nonideal Aqueous Solutions Using Molecular Dynamics Simulations. J Phys Chem B. 2018;122(21):5515–5526.

[23] Galata AA, Anogiannakis SD, Theodorou DN. Thermodynamic analysis of Lennard-Jones binary mixtures using Kirkwood-Buff theory. Fluid Ph Equilibria. 2017;.

[24] Heidari M, Kremer K, Potestio R, Cortes-Huerto R. Fluctuations, Finite-Size Effects and the Thermodynamic Limit in Computer Simulations: Revisiting the Spatial Block Analysis Method. Entropy. 2018;20(4).

[25] Schnell SK, Vlugt TJH, Simon JM, Bedeaux D, Kjelstrup S. Thermodynamics of a small system in a μT reservoir. Chem Phys Lett. 2011;504(4):199 – 201.

[26] Tester JW, Modell M. Chapter 9 - Property Relationships of Mixtures. In: Caren KM, editor. Thermodynamics and Its Applications. 3rd ed. Prentice Hall PTR; 1997. p. 318–387.

[27] DeHoff R. Chapter 8 - Multicomponent Homogeneous Nonreacting Systems: Solutions. In: DeHoff R, editor. Thermodynamics in Materials Science. Taylor and Francis Group; 2006. p. 199–248.

[28] Kjelstrup S, Bedeaux D, Johannessen E, Gross J. Chapter 3 - The entropy production of one-dimensional transport processes. In: Non-Equilibrium Thermodynamics for Engineers. World Scientific Publishing Co. Pte. Ltd.; 2010. p. 22–44.

[29] Widom B. Some Topics in the Theory of Fluids. J Chem Phys. 1963;39(11):2808–2812.

[30] Tester JW, Modell M. Chapter 8 - Properties of Pure Materials. In: Caren KM, editor. Thermodynamics and Its Applications. 3rd ed. Prentice Hall PTR; 1997. p. 239–317.

[31] Dill KA, Bromberg S. Chapter 10 - The Boltzmann Distribution Law. In: Dill KA, editor. Molecular Driving Forces. 2nd ed. Garland Science; 2011. p. 169–192.

[32] Hansen JP, McDonald IR. Chapter 2 - Statistical Mechanics and Molecular Distribution Functions. In: Hansen JP, McDonald IR, editors. Theory of Simple Liquids. 4th ed. Oxford: Academic Press Inc.; 2013. p. 13–59.

[33] Ben-Naim A. Chapter 1 - Introduction. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 1–20.

[34] Frenkel D, Smit B. Chapter 6 - Molecular Dynamics in Various Ensembles. In: Frenkel D, Smit B, editors. Understanding Molecular Simulation (Second Edition). second edition ed. San Diego: Academic Press; 2002. p. 139 – 163.

[35] Callen HB. Thermodynamics and an Introduction to Thermostatistics. 2nd ed. New York, USA: John Wiley and Sons; 1985.

[36] Hansen JP, McDonald IR. Chapter 2 - Statistical Mechanics and Molecular Distribution Functions. In: Hansen JP, McDonald IR, editors. Theory of Simple Liquids. London: Academic Press Inc.; 1967. p. 14–44.

[37] Frenkel D, Smit B. Chapter 4 - Molecular Dynamics Simulations. In: Frenkel D, Smit B, editors. Understanding Molecular Simulation (Second Edition). second edition ed. San Diego: Academic Press; 2002. p. 63 – 107.

[38] Leach A. Chapter 6 - Computer Simulation Methods. In: Leach A, editor. Molecular Modelling : Principles and Applications. 2nd ed. Edinburgh Gate: Pearson Education Limited; 2001. p. 303–352.

[39] Ben-Naim A. Appendix G - The long-range behaviour of the pair correlation function. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 323–332.

[40] Leach A. Chapter 7 - Molecular Dynamics Simulation Methods. In: Leach A, editor. Molecular Modelling : Principles and Applications. 2nd ed. Edinburgh Gate: Pearson Education Limited; 2001. p. 353–409.

[41] Atkins P, de Paula J. Atkins' Physical Chemistry. 10th ed. Oxford: Oxford University Press; 2014.

[42] Allen MP, Tidesely DJ. Chapter 3 - Molecular Dynamics. In: Allen MP, Tidesely DJ, editors. Computer Simulations of Liquids. Oxford University Press Inc.; 1987. p. 71–109.

[43] Leach A. Chapter 6 - Empirical Force Fields: Molecular Mechanics. In: Leach A, editor. Molecular Modelling : Principles and Applications. 2nd ed. Edinburgh Gate: Pearson Education Limited; 2001. p. 165–252.

[44] Hansen JP, McDonald IR. Chapter 3 - Computer Experiments on Liquids. In: Hansen JP, McDonald IR, editors. Theory of Simple Liquids. London: Academic Press Inc.; 1976. p. 45–62.

[45] Ewald PP. Die Berechnung optischer und elektrostatischer Gitterpotentiale. Ann Phys;369(3):253–287.

[46] Hill TL. Thermodynamics of Small Systems, Parts 1 and 2. New York: Dover Publications, Inc.; 2013.

[47] Hill TL. Thermodynamics of Small Systems. J Chem Phys. 1962;36(12):3182–3197.

[48] Stukowski A. Modelling Simul. Mater Sci Eng. 2010;(18):015012.

[49] Strøm BA, Simon JM, Schnell SK, Kjelstrup S, He J, Bedeaux D. Size and shape effects on the thermodynamic properties of nanoscale volumes of water. Phys Chem Chem Phys. 2017;19:9016–9027.

[50] Gibbs JW. The Scientific Papers of J. Willard Gibbs, Volume 1, Thermodynamics. Woodbridge, Connecticut: Ox Bow Press; 1993.

[51] Schnell SK, Skorpa R, Bedeaux D, Kjelstrup S, Vlugt TJH, Simon JM. Partial molar enthalpies and reaction enthalpies from equilibrium molecular dynamics simulation. J Chem Phys. 2014;141(14):144501.

[52] Klain DA. A short proof of Hadwiger's characterization theorem. Mathematika. 1995;42(2):329–339.

[53] Blokhuis EM. Existence of a bending rigidity for a hard-sphere liquid near a curved hard wall: Validity of the Hadwiger theorem. Phys Rev E. 2013 Feb;87:022401.

[54] Dawass N, Krüger P, Schnell SK, Bedeaux D, Kjelstrup S, Simon JM, et al. Finite-size effects of Kirkwood–Buff integrals from molecular simulations. Mol Sim. 2017;0(0):1–14.

[55] Dawass N, Krüger P, Simon JM, Vlugt TJH. Kirkwood–Buff integrals of finite systems: shape effects. Mol Phys. 2018;116(12):1573–1580.

[56] Ben-Naim A. Chapter 3 - Thermodynamic quantities expressed in terms of molecular distribution functions. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 76–111.

[57] Román FL, White JA, González A, Velasco S. Ensemble Effects in Small Systems. In: Mulero Á, editor. Theory and Simulation of Hard-Sphere Fluids and Related Systems. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. p. 343–381.

[58] Rovere M, Heermann DW, Binder K. The gas-liquid transition of the two-dimensional Lennard-Jones fluid. J Phys Condens Matter. 1990;2(33):7009.

[59] van Rossum G. Python tutorial, Technical Report CS-R9526. Amsterdam: Centrum voor Wiskunde en Informatica (CWI); 1995.

[60] Plimpton S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. J Comp Phys. 1995;117:1–19.

[61] LAMMPS Molecular Dynamics Simulator [Internet]. Sandia Corporation; 2018. [cited 2018 May 31]. Available from: `http://lammps.sandia.gov/`.

[62] Tuckerman M, Alejandre J, L{'opez-Rend{'on R, Jochim A, Martyna G. A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal-isobaric ensemble. J Phys A. 2006 5;39(19):5629–5651.

[63] Shinoda W, Shiga M, Mikami M. Rapid estimation of elastic constants by molecular dynamics simulation under constant stress. Phys Rev B. 2004;69:134103.

[64] Martyna GJ, Tobias DJ, Klein ML. Constant pressure molecular dynamics algorithms. J Chem Phys. 1994;101(5):4177–4189.

[65] Tuckerman ME, Martyna GJ. Understanding Modern Molecular Dynamics: Techniques and Applications. J Phys Chem B. 2000;104(2):159–178.

[66] Sun H. COMPASS: An ab Initio Force-Field Optimized for Condensed-Phase ApplicationsOverview with Details on Alkane and Benzene Compounds. J Phys Chem B. 1998;102(38):7338–7364.

[67] LAMMPS 11 May 2018 Documentation [Internet]. Sandia Corporation; 2013. [cited 2018 May 31]. Available from: `http://lammps.sandia.gov/doc/pair_modify.html`.

[68] Lorentz HA. Ueber die Anwendung des Satzes vom Virial in der kinetischen Theorie der Gase. Ann Phys;248(1):127–136.

[69] Berthelot D. Sur Le Mélange Des Gaz. Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences. 1898;(126):1703–1855.

[70] Daura X, Mark A, van Gunsteren W. Parametrization of aliphatic CH_n united atoms of GROMOS96 force field. J Comput Chem. 1998;19:535–547.

[71] Berendsen HJC, Grigera JR, Straatsma TP. The missing term in effective pair potentials. J Phys Chem. 1987;91(24):6269–6271.

[72] Chatterjee S, Debenedetti PG, Stillinger FH, Lynden-Bell RM. A computational investigation of thermodynamics, structure, dynamics and solvation behavior in modified water models. J Chem Phys. 2008;128(12):124511.

[73] LAMMPS Molecular Dynamics Simulator [Internet]. Sandia Corporation; 2013. [cited 2018 June 15]. Available from: `http://lammps.sandia.gov/doc/compute_rdf.html`.

[74] MATLAB. version 8.1.0.604 (R2013a). Natick, Massachusetts: The MathWorks Inc.; 2013.

[75] Ben-Naim A. Chapter 5 - Ideal Solutions. In: Ben-Naim A, editor. Molecular Theory of Solutions. OUP Oxford; 2006. p. 136–155.

[76] Kreyszig E. Advanced Engineering Mathematics. 10th ed. Jefferson City, MO: John Wiley & Sons, Inc.; 2011.

[77] Alsberg BK. Chemometrics, Compendium for TKJ4175/KJ8175;.

[78] Schnell SK, Vlugt TJH, Simon JM, Bedeaux D, Kjelstrup S. Thermodynamics of small systems embedded in a reservoir: a detailed analysis of finite size effects. Mol Phys. 2012;110(11-12):1069–1079.

[79] Ruckenstein E, Shulgin I. Entrainer effect in supercritical mixtures. Fluid Ph Equilibria. 2001;180(1):345 – 359.

[80] Brehm M, Kirchner B. TRAVIS - A Free Analyzer and Visualizer for Monte Carlo and Molecular Dynamics Trajectories. J Chem Inf Model. 2011;51(8):2007–2023.

[81] Humphrey W, Dalke A, Schulten K. VMD – Visual Molecular Dynamics. Journal of Molecular Graphics. 1996;14:33–38.

[82] Lide DR. Section 6 - Fluid Properties "Pressure and Temperature Dependence of Liquid Density". In: Lide DR, editor. CRC Handbook of Chemistry and Physics, Internet Version 2005. Boca Raton, FL: CRC Press; 2005. p. **6**–140–**6**–141.

[83] Easteal AJ, Woolf LA. (p, Vm, T, x) measurements for [(1 x)H2O+xCH3OH] in the range 278 to 323 K and 0.1 to 280 MPa I. Experimental results, isothermal compressibilities, thermal expansivities, and partial molar volumes. J Chem Thermodyn. 1985;17(1):49 – 62.

[84] Douheret G, Khadir A, Pal A. Thermodynamic characterization of the water + methanol system, at 298.15K. Thermochimica Acta. 1989;142(2):219 – 243.

[85] Soetens JC, Bopp PA. Water–Methanol Mixtures: Simulations of Mixing Properties over the Entire Range of Mole Fractions. J Phys Chem B. 2015;119(27):8593–8599.

[86] Lin B, He X, MacKerell Jr AD. A Comparative Kirkwood-Buff Study of Aqueous Methanol Solutions Modeled by the CHARM Additive and Drude Polarizable Force Fields. J Phys Chem B. 2013;117(36):10572–10580.

[87] Taylor JR. Error Analysis - The Study og Uncertainties in Physical Measurements. 2nd ed. McGuire A, editor. Colorado: University Science Books; 1997.

Symbol Lists

List of Latin symbols.

Symbol	Description
$\mathbf{a}$	Acceleration vector for particle
b	Density of general property B
b^{s}	Surface contribution of density of property B
b^{∞}	Volume contribution of density of property B
B	General extensive property
$\widetilde{B}$	General extensive property
B^{GC}	General property in the grand canonical ensemble
B^{ID}	B in ideal state
B_i	Partial molar B
B_{t}	General property for the total ensemble
c_i	Bulk molecular concentration of component i
C_{ik}	Proportionality constant in Cortes-Huerto scaling equation [20]
C_V	Heat capacity at constant volume
E	Energy
E_{LTC}	Long-range tail corrections for energy
F_i	Force vector acting on particle i
g_{ik}	Radial distribution function
g_{ideal}	Radial distribution function of ideal system
$g_{ik}^{\mathrm{Ganguly}}$	Radial distribution function with finite size correction by Ganguly *et al.* [8]
g_{ik}^{∞}	Radial distribution function of open, infinite system
G	Gibbs energy
G_{ik}	Kirkwood-Buff integral of components i and k in thermodynamic limit
G_{ik}^{V}	Kirkwood-Buff integral of components i and k for finite volume V
$\mathscr{H}$	Hamiltonian
k	Parameter measuring overall fit of curve
k_{B}	Boltzmann constant
$\mathbf{k}$	Reciprocal vectors in Ewald sum
$\mathscr{K}$	Kinetic energy
L	Length
L_0	Size in one dimension of reservoir (total simulation box)
m_i	Mass of component i
n	Number of moles
n_i	Number density of particle i
$\mathbf{n}$	Vector of positions of image boxes used in Ewald summation
N_i	Number of particles of type i
$\mathscr{N}$	Number of replicas of small systems constructing an ensemble
p	Pressure
$\widehat{p}$	Integral pressure
p_{LTC}	Long-range tail corrections for pressure
$\mathbf{p}_i$	Momentum vector for particle i
q_i	Charge of particle i

r_{ij}	Distance between particles *i* and *j*
r_c	Cut-off for pair potential
$\mathbf{r}$	Position vector
$\mathbf{r}_i$	Position vector for particle *i*
R	Radius
S	Entropy
t	Time
T	Absolute temperature
U	Internal energy
v_1	External field contribution to potential energy
v_2	Pair interaction contribution to potential energy
$v_{2,ik}$	Pair potential between particles of type *i* and *k*
v_3	Triplet interaction contribution to potential energy
$\mathbf{v}$	Velocity vector
V	Volume
V_0	Volume of reservoir (total simulation box)
V_ξ	Volume with radius equal to structural correlation length
$\mathcal{V}$	Potential energy
$\mathcal{V}_{\text{Coulomb}}$	Coulombic interactions
$\mathcal{V}_{\text{Ewald}}$	Ewald sum potential
$\mathcal{V}_{\text{LJ}}$	Lennard-Jones potential
$\mathcal{V}_{\text{pair}}$	Pair potential
w	Geometrical function in the method by Krüger *et al.* [19]
x	Coordinate
x_i	Mole fraction of component *i*
X	Replica energy
y	Coordinate
z	Coordinate

List of Greek symbols.

Symbol	Description
δ_{ik}	Kronecker-Delta
ε	Potential depth in Lennard-Jones potential
ε_0	Dielectric constant
κ_T	Isothermal compressibility
λ	Scaling factor in the Cortes-Huerto scaling equation [20] (3.47)
μ_i	Chemical potential of component i
ν_{ik}	Fluctuation expression based on densities of particles i and k
ρ_i	Bulk molecular concentration of component i
$\rho_i^{(1)}$	Singlet distribution function
$\rho_{ik}^{(2)}$	Pair distribution function
Ω	Surface area
σ	Molecular diameter in Lennard-Jones potential
σ_B^2	Average fluctuation of property B
ξ	Structural correlation length
Ξ	Grand canonical partition function

Definitions of reduced quantities in Lennard-Jones units.

Quantity	Definition in Lennard-Jones Units
Density	$n^* = \frac{N}{V}(\sigma_{11})^3$
Energy	$U^* = \frac{U}{\varepsilon_{11}}$
Length	$x^* = \frac{x}{\sigma_{11}}$
Mass	$m^* = \frac{m}{m_1}$
Pressure	$p^* = \frac{p}{\varepsilon_{11}}\sigma_{11}$
Temperature	$T^* = \frac{k_\mathrm{B}T}{\varepsilon_{11}}$
Time	$t^* = \frac{t}{\sigma_{11}}\left(\frac{\varepsilon_{11}}{m_1}\right)^{(1/2)}$
Velocity	$v^* = v\left(\frac{m_1}{\varepsilon_{11}}\right)^{(1/2)}$

A Statistical Analysis

All results were computed in five parallels. The expression for the standard deviation of the mean of one data point is

$$\sigma_{\bar{a}} = \sqrt{\frac{\sum(a_i - \bar{a})^2}{N(N-1)}} \tag{A.1}$$

where a_i is a separate measurement, $\bar{a}$ is the mean value and N is the number of parallels [87]. When the final properties are parameters found from linear curve fitting with an expression on the form

$$y = A + Bx \tag{A.2}$$

the standard deviation of A and B can also be calculated by measuring how well the fitted curve represent the data points x. As explained in reference [87] the standard deviations of the parameters A and B are

$$\sigma_A = \sigma_y \sqrt{\frac{\sum x_i^2}{\Delta}} \tag{A.3}$$

$$\sigma_B = \sigma_y \sqrt{\frac{N}{\Delta}} \tag{A.4}$$

where

$$\sigma_y = \sqrt{\frac{1}{N-2}\sum_{i=1}^{N}(y_i - A - Bx_i)^2} \tag{A.5}$$

and

$$\Delta = N\sum x_i^2 - \left(\sum x\right)^2 \tag{A.6}$$

The curve fitting is performed by MATLAB, which applies the following expressions to find A and B

$$A = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{\Delta} \tag{A.7}$$

$$B = \frac{N \sum x_i y_i - \sum x_i \sum y_i}{\Delta} \tag{A.8}$$

The total fit the different methods can be expressed by how close it is to the theoretical values, or a method that is known to provide accurate results. For the whole range of mole fractions, the fit can be summarized by the factor k. This factor is the sum of the squared difference between the each data point and its corresponding theoretical value, divided by the total number of data points, which in this case is five.

$$k = \frac{\sum_i \Delta x_i^2}{5} = \frac{\sum_i (x_{i,\text{computation}} - x_{i,\text{theory}})^2}{5} \tag{A.9}$$

B Additional Figures

The following sections contain additional figures containing supplementary information. This includes the results of the computations using a total reservoir in the NVT ensemble and the comparison to the original results obtained by Galata *et al.* [23]. The latter was performed in order to make sure that the lattice partitioning suggested by the authors was correctly implemented.

Figures B.1-B.2 are the results obtained in the NVT ensemble, equivalent to figures 5.2-5.3 with results obtained in the NpT ensemble. Figures B.3-B.4 are the results obtained in the NVT ensemble, equivalent to figures 5.8-5.9 with results obtained in the NpT ensemble. Figures B.5-B.6 show the comparison of the original results obtained by Galata *et al.* [23] together with the results obtained applying the lattice partitioning and the averaging method they proposed, in addition to the results obtained using ensemble averages and lattice partitioning.

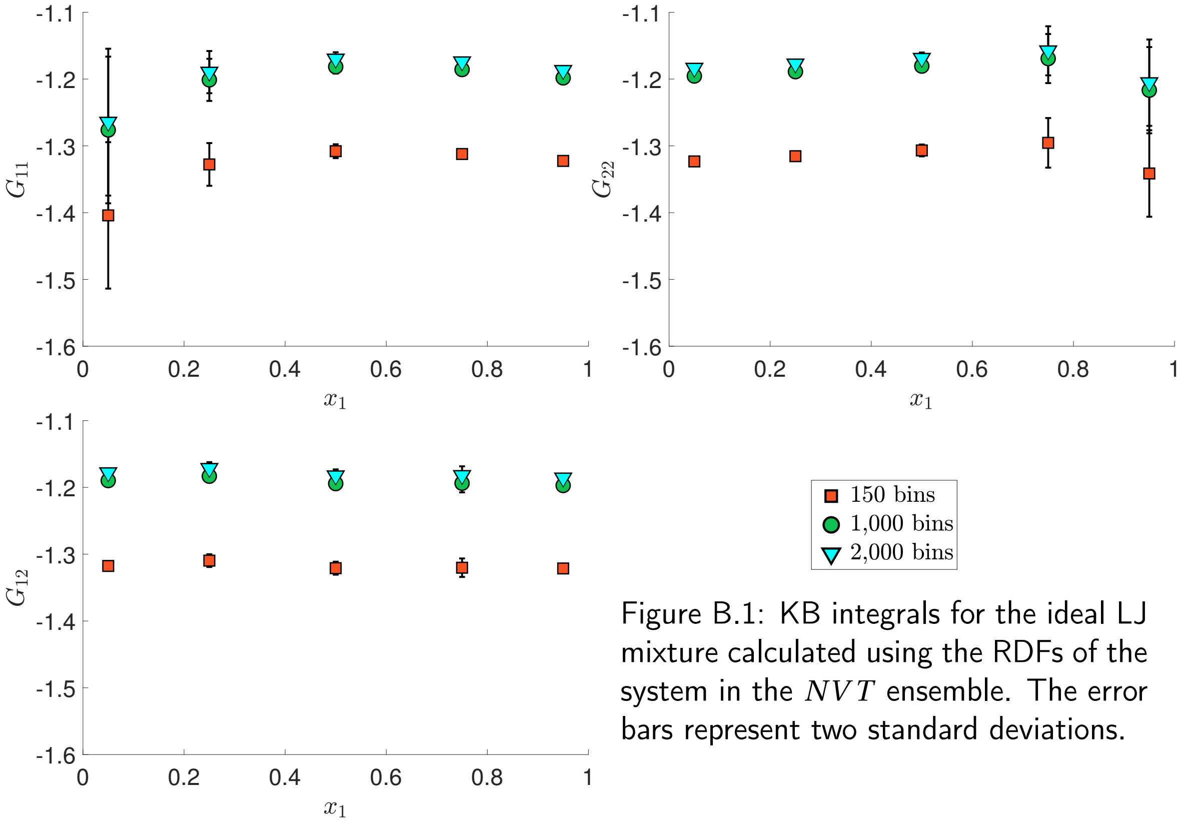

Figure B.1: KB integrals for the ideal LJ mixture calculated using the RDFs of the system in the NVT ensemble. The error bars represent two standard deviations.

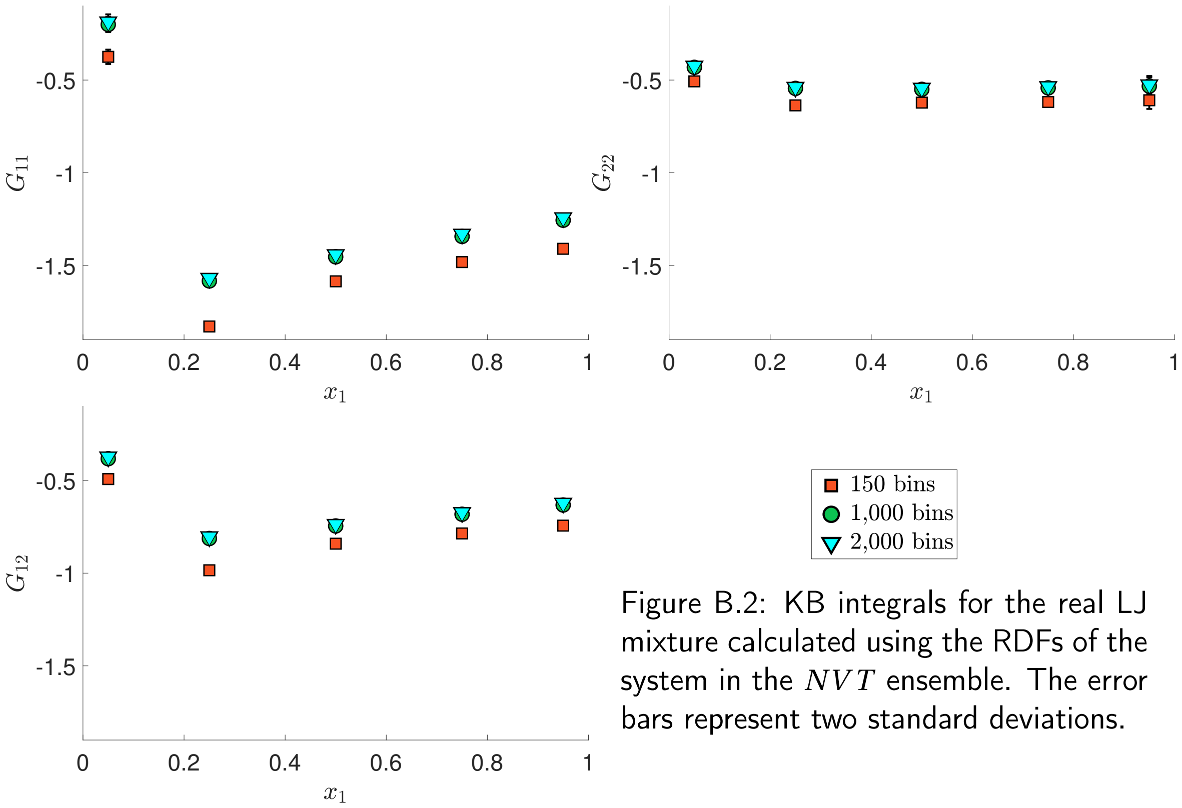

Figure B.2: KB integrals for the real LJ mixture calculated using the RDFs of the system in the NVT ensemble. The error bars represent two standard deviations.

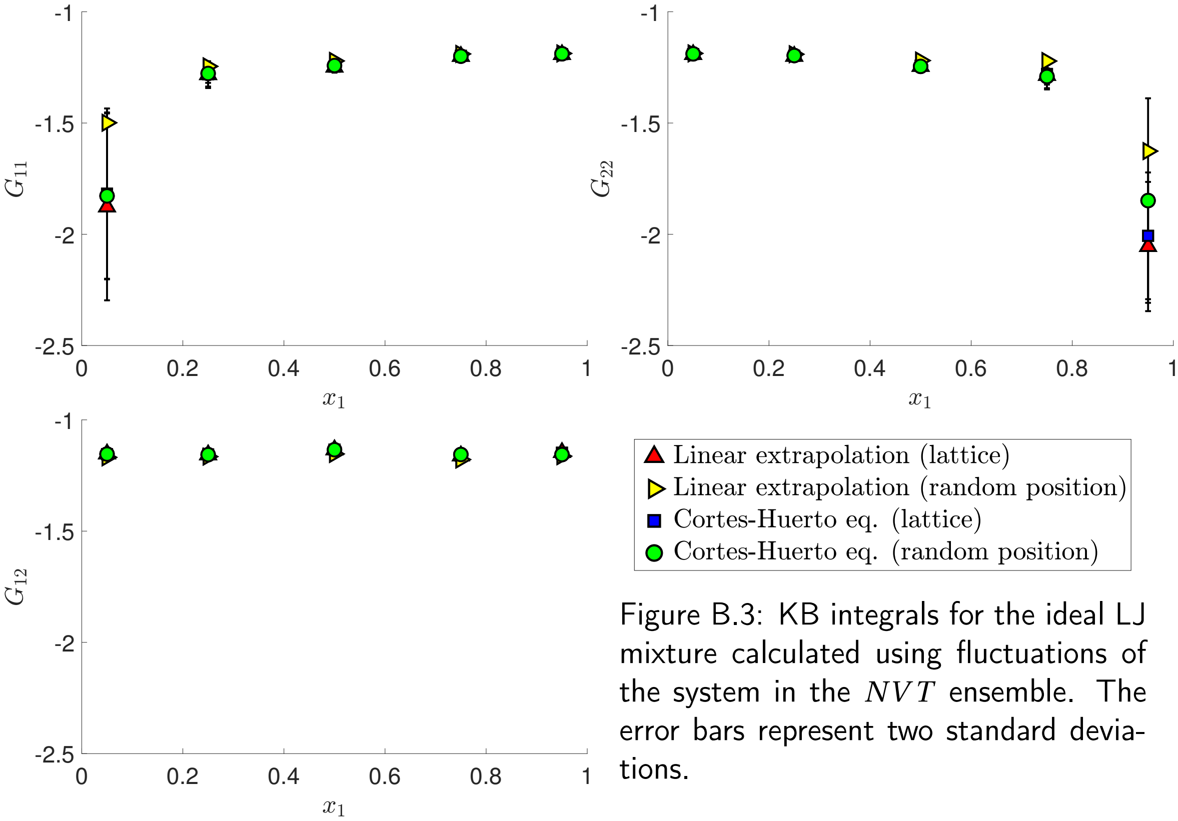

Figure B.3: KB integrals for the ideal LJ mixture calculated using fluctuations of the system in the NVT ensemble. The error bars represent two standard deviations.

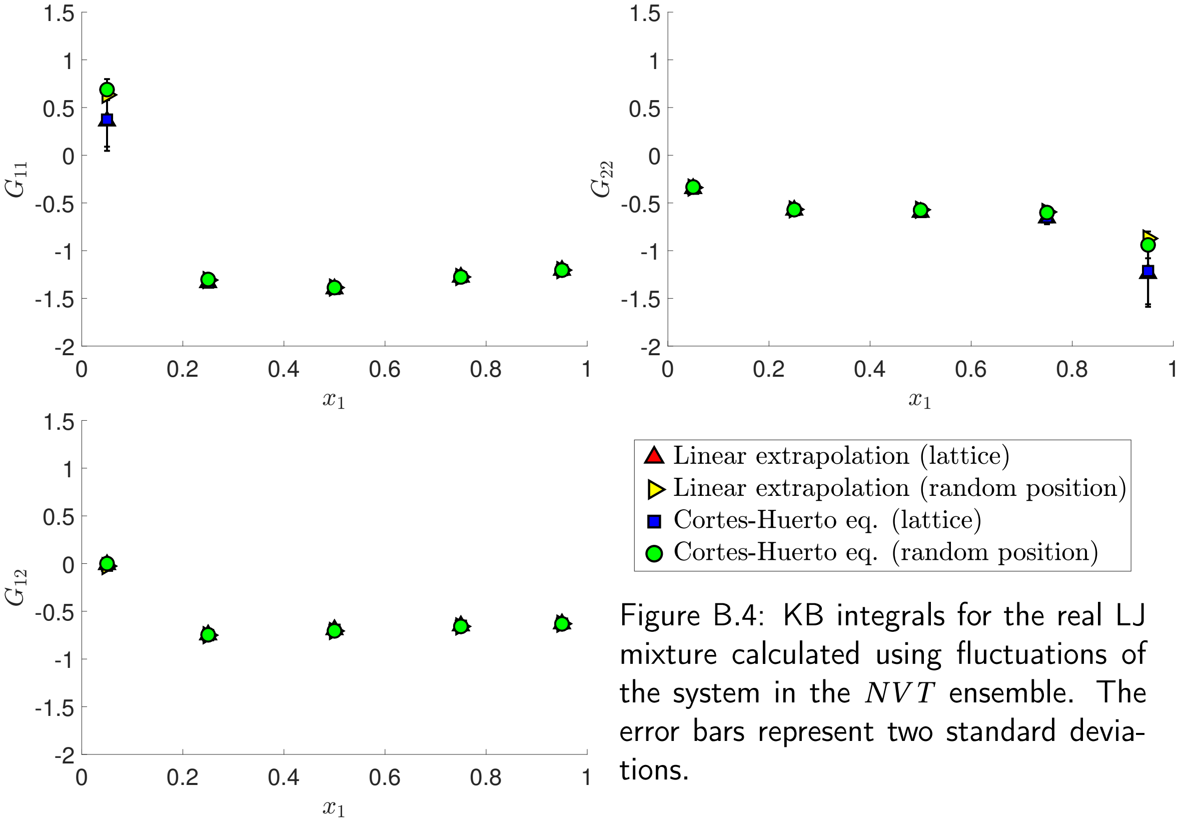

Figure B.4: KB integrals for the real LJ mixture calculated using fluctuations of the system in the NVT ensemble. The error bars represent two standard deviations.

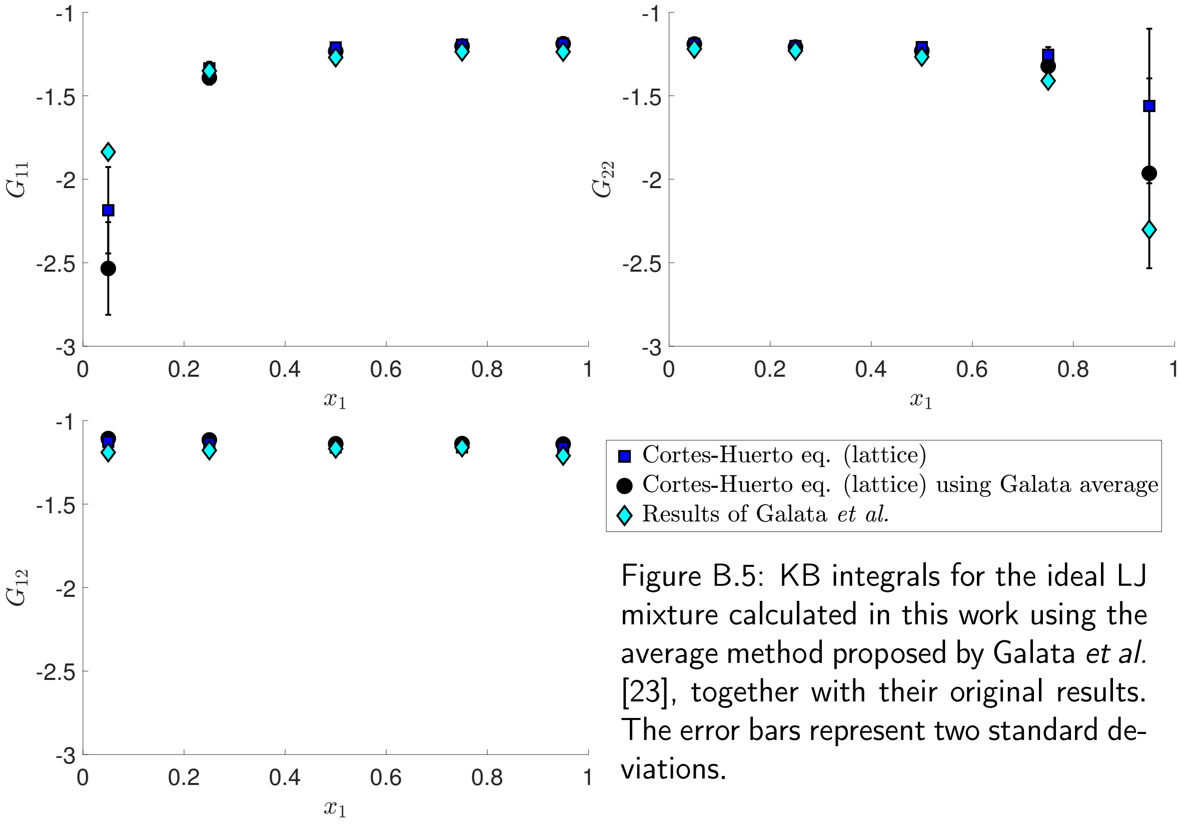

Figure B.5: KB integrals for the ideal LJ mixture calculated in this work using the average method proposed by Galata *et al.* [23], together with their original results. The error bars represent two standard deviations.

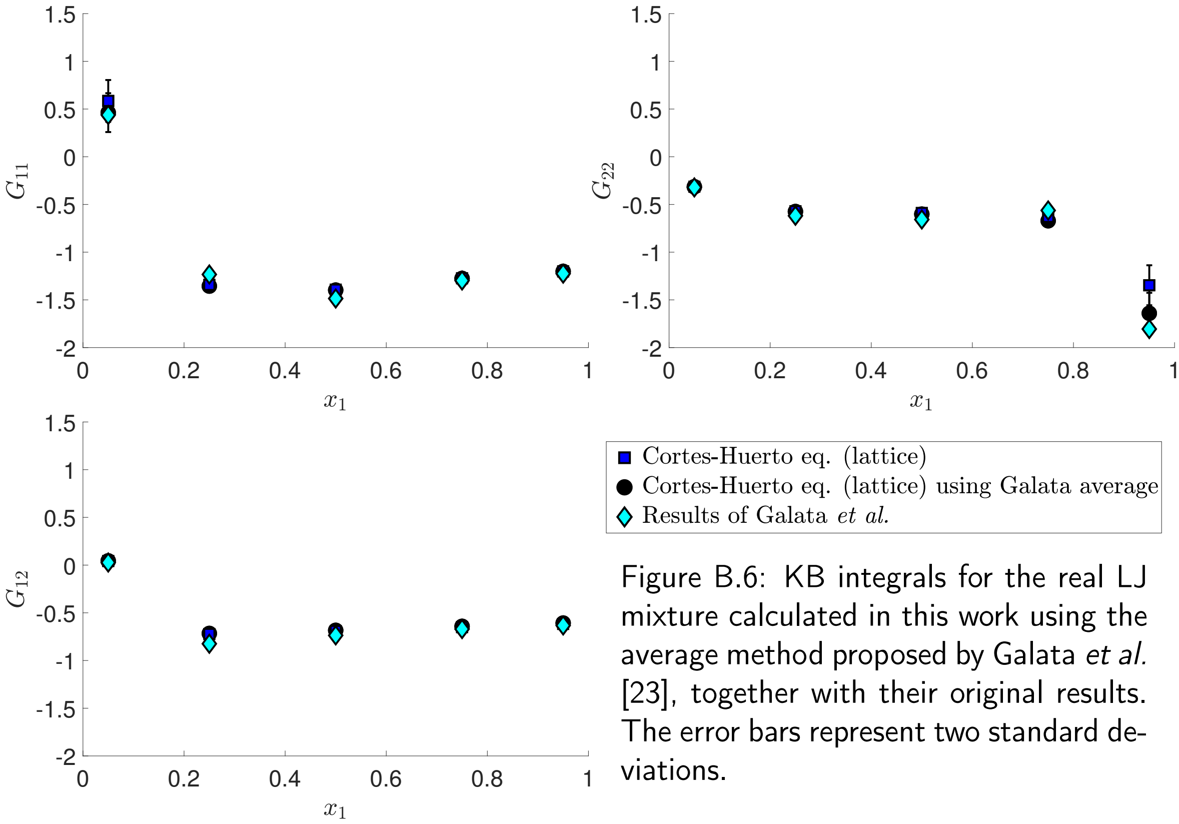

Figure B.6: KB integrals for the real LJ mixture calculated in this work using the average method proposed by Galata *et al.* [23], together with their original results. The error bars represent two standard deviations.

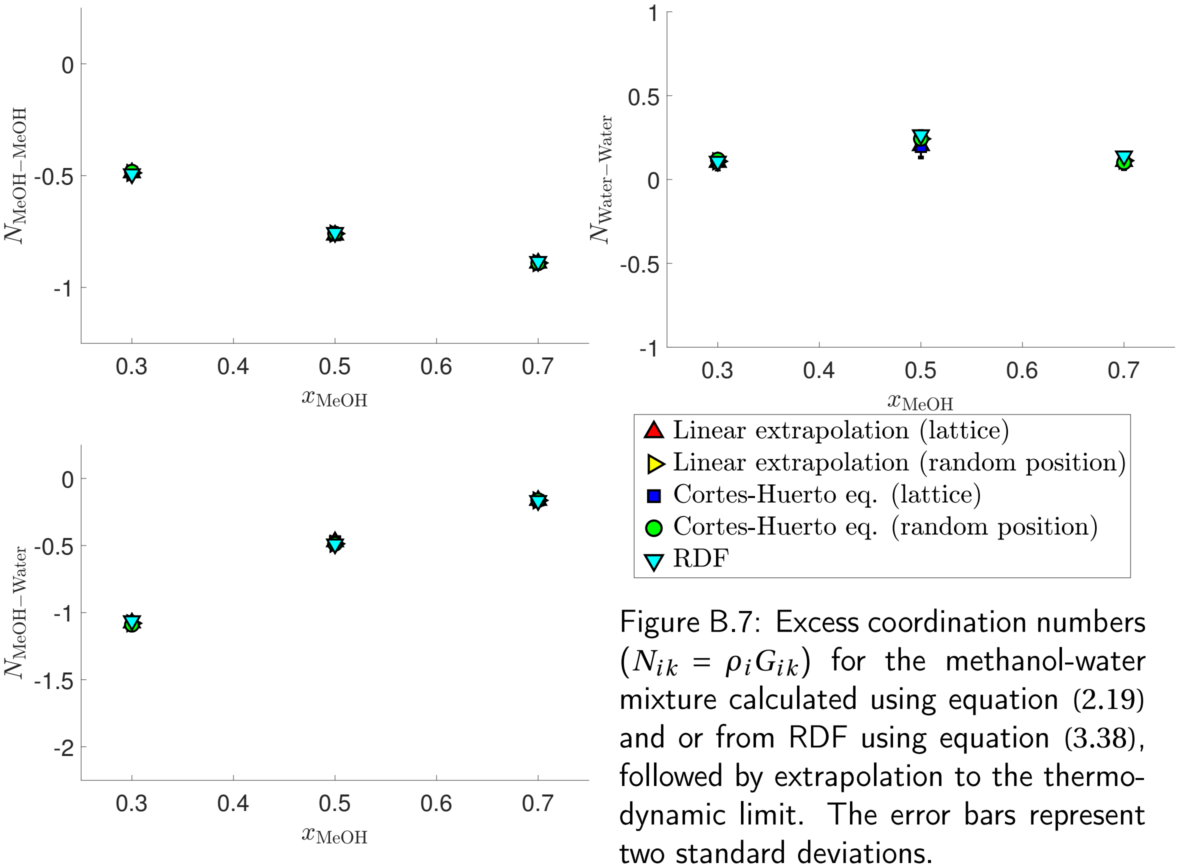

Figure B.7: Excess coordination numbers ($N_{ik} = \rho_i G_{ik}$) for the methanol-water mixture calculated using equation (2.19) and or from RDF using equation (3.38), followed by extrapolation to the thermodynamic limit. The error bars represent two standard deviations.

C Python scripts

The following sections contain Python scripts used for fluctuation calculations and post-processing of the results. This includes the script used to create initial configurations in section C.1, the integration of the RDF in section C.2, the fluctuation calculations of LJ mixtures using random subvolumes in section C.3, the fluctuation calculations of LJ mixtures using lattice partitioning C.4 and the fluctuation calculations of the methanol-water mixture including both random positioned subvolumes and lattice partitioning in section C.5.

C.1 Creating Initial Configurations

```
# Master Project
# Vilde Braten
# 15.02.2018

import numpy as np
import os
import sys
from random import randint

def change_id(data,x1,N):
    """
    Changes composition of pure mixture by editing write_data file
    from LAMMPS.
    """

    N1 = 0
    N2 = 0
    pID = np.zeros((N,1))
    pType = np.zeros((N,1))
    counter = -1

    for i in data[:(N+21)]:
        i = i.split()

        if (len(i) == 10):
            counter += 1
            pType[counter] = float(i[2])
            pID[counter] = float(i[0])

            if (pID[counter] > N*x1):
                pType[counter] = 2
    # checking correct number of atoms
    for j in pType:
        if (j == 1):
            N1 += 1
        elif (j == 2):
            N2 += 1

    print(N1,N2)

    return pType


if __name__ == "__main__":

    INPUT = "initconfig.org"
    OUTPUT = "initconfig"
    inputFile = open(INPUT, "r")
    data = [x for x in inputFile]
    inputFile.close()
    x1 = 0
    N = 20000
    res = change_id(data,x1,N)
    part = -1

    # printing
    outputFile = open(OUTPUT, "w")
    for i in data:
        i = i.split()

        if (len(i) == 10):
            part += 1
            outputFile.write(str(i[0])+" "+str(i[1]) \
                             +" "+str(int(res[part]))+" ")
            for j in i[3:]:
                outputFile.write(str(j)+" ")
            outputFile.write("\n")

        else :
            for j in i:
                outputFile.write(str(j)+" ")
            outputFile.write("\n")
```

C.2 Integrating RDF

```
# Master Project
# Vilde Braten
# vildebr@stud.ntnu.no
# 19.02.2018

import numpy as np
import scipy.integrate
import os
import sys

def corr_rdf(g,r,N,L):
    """
    Correcting finite size effects of RDF. Using Ganguly
    correction.
    """

    if (g.shape[0] != r.shape[0]):
        print("Vectors are different length. Exiting")
        sys.exit()

    inters = g.shape[1]
    bins = g.shape[0]

    vol = np.zeros_like(r)
    deltaN = np.zeros_like(g)
    gVDV = np.zeros_like(g)

    totvol = L*L*L
    meanvol = meanL*meanL*meanL
    hbox = L/2.0
    rho = np.array([N[0], N[1], N[1]])/meanvol

    for j in range(bins):
        if (r[j]<hbox):
            vol[j] = (4.0/3.0)*np.pi*r[j]**3
        else:
            print("Cut-off is too long. Exiting")
            sys.exit()

    for j in range(1,bins+1):
        deltaN[j-1,0] = scipy.integrate.trapz((g[:j,0]-1.0)*r[:j]**2 \
                                              *rho[0],r[:j])
        deltaN[j-1,1] = scipy.integrate.trapz((g[:j,1]-1.0)*r[:j]**2 \
                                              *rho[1],r[:j])
        deltaN[j-1,2] = scipy.integrate.trapz((g[:j,2]-1.0)*r[:j]**2 \
                                              *rho[1],r[:j])

    deltaN = deltaN*4.0*np.pi

    for j in range(bins):
        gGanguly[j,0] = g[j,0]*((N[0]*(1.0-vol[j]/totvol))/ \
                            (N[0]*(1.0-vol[j]/totvol)-deltaN[j,0]-1))
        gGanguly[j,1] = g[j,1]*((N[1]*(1.0-vol[j]/totvol))/ \
                            (N[1]*(1.0-vol[j]/totvol)-deltaN[j,1]-1))
        gGanguly[j,2] = g[j,2]*((N[1]*(1.0-vol[j]/totvol))/ \
                            (N[1]*(1.0-vol[j]/totvol)-deltaN[j,2]))

    return gGanguly


def KBint_rdf(g,r):
    """
    Integrate the pair-correlation function for all interactions.
    Using analytical expression for weighting parameter for 3D sphere.
    """

    if (g.shape[0] != r.shape[0]):
        print("Vectors are different length. Exiting")
        sys.exit()

    inters = g.shape[1]
    bins = g.shape[0]

    h = g - 1.0
    G = np.zeros_like(g)

    for i in range(inters):
        for j in range(1,bins+1):
            G[j-1,i] = scipy.integrate.trapz(h[:j,i]*r[:j]**2 \
                        *(1.0-1.5*(r[:j]/r[j-1])+0.5*(r[:j]/ \
                          r[j-1])**3),r[:j])

    G = G*4.0*np.pi

    return G


if __name__ == "__main__":
```

```
numPar = 5

inputParam = "inputRDF.txt"
inputData = np.loadtxt(inputParam)
N = [int(inputData[0]), int(inputData[1])]
bins = int(inputData[2])
filename = "rdf%s.out" %(bins)

inputSize = "size.out"
sizeData = np.loadtxt(inputSize)
L = float(sizeData[np.argmin(sizeData[:,0]),0])
L = float(mean(sizeData[:,0]))

gGangulyTOT = []
KBintGangulyTOT = []
KBintTOT = []

for i in range(numPar):
    data = np.genfromtxt(filename,skip_header=(4+(bins+1)*i), \
                         max_rows=bins)
    r = data[:,1]
    r += r[0]/2.0
    g = data[:,2::2]
    gGanguly = corr_rdf(g,r,N,L)
    KBint = KBint_rdf(g,r)
    KBintGanguly = KBint_rdf(gGanguly,r)

    gGangulyTOT.append(gGanguly)
    KBintGangulyTOT.append(KBintGanguly)
    KBintTOT.append(KBint)
```

C.3 Fluctuations SSM - LJ mixtures

```
# Master Project
# Vilde Braten
# 21.03.2018

import numpy as np
import random
import os

def fluct_calc(inputFile):
    """
    Reads LAMMPS dump file and calculates density fluctuations.
    """

    delta = (maxR-minR)/bins
    radial = [(minR + delta*(m+1)) for m in range(bins)]

    n1 = np.zeros(bins)
    n2 = np.zeros(bins)
    n1n1 = np.zeros(bins)
    n2n2 = np.zeros(bins)
    n1n2 = np.zeros(bins)

    random.seed(2873698)

    counter = 0
    for snapshot in range(numSnap):

        # read N
        _ = inputFile.readline()
        _ = inputFile.readline()
        _ = inputFile.readline()
        line = inputFile.readline()
        N = int(line)

        # read box dimensions
        _ = inputFile.readline()
        line = inputFile.readline()
        _boxlo, _boxhi = line.split()
        boxlo, boxhi = float(_boxlo), float(_boxhi)

        boxL = boxhi-boxlo
        hbox = boxL/2
```

```
# move forward by three lines
_ = inputFile.readline()
_ = inputFile.readline()
_ = inputFile.readline()

snapshots = []
for i in range(N):
    line = inputFile.readline()
    _, ptype, dx, dy, dz = line.split()
    snapshots.append([float(ptype), float(dx), float(dy), \
                      float(dz)])

for k in range(rndSmp):
    posx = boxlo + boxL*random.uniform(0,1)
    posy = boxlo + boxL*random.uniform(0,1)
    posz = boxlo + boxL*random.uniform(0,1)

    n_1 = np.zeros(bins)
    n_2 = np.zeros(bins)

    for snapshot in snapshots:
        ptype, dx, dy, dz = snapshot
        dx -= posx
        dy -= posy
        dz -= posz

        if (dx > hbox):
            dx -= boxL
        elif (dx < -hbox):
            dx += boxL

        if (dy > hbox):
            dy -= boxL
        elif (dy < -hbox):
            dy += boxL

        if (dz > hbox):
            dz -= boxL
        elif (dz < -hbox):
            dz += boxL

        dr = np.sqrt(dx*dx + dy*dy + dz*dz)

        for r in range(bins):
            if (dr < radial[r]):
                if (ptype == 1):
```

```
                            n_1[r] += 1.0
                        elif (ptype == 2):
                            n_2[r] += 1.0

            for g in range(bins):
                n1[g] += n_1[g]
                n2[g] += n_2[g]
                n1n1[g] += n_1[g]*n_1[g]
                n2n2[g] += n_2[g]*n_2[g]
                n1n2[g] += n_1[g]*n_2[g]

            counter += 1.0

    n1 /= counter
    n2 /= counter
    n1n1 /= counter
    n2n2 /= counter
    n1n2 /= counter

    return radial,n1,n2,n1n1,n2n2,n1n2


if __name__ == "__main__":

    minR = 1.0
    maxR = 14.0
    bins = 100
    rndSmp = 10
    numSnap = 40000


    with open("SSM.dump", "r") as inputFile:
        radial,n1,n2,n1n1,n2n2,n1n2 = fluct_calc(inputFile)
```

C.4 Fluctuations Lattice - LJ Mixtures

```
# Master Project
# Vilde Braten
# vildebr@stud.ntnu.no
# 23.02.2018

import numpy as np
import os
import sys

def search_files(dirName):
    """
    Searches for all files with names 'latticeSSMX.out' and returns
    array of X values sorted by increasing value and array of input
    file names sorted by increasing X values.
    """

    tmp = os.popen("ls "+dirName).read().split()
    fileNames = []
    sortNames = []
    num = []

    for i in tmp:
        if (i[:10] == "latticeSSM"):
            fileNames.append(i)

    for j in fileNames:
        if (len(j) == 15):
            num.append(int(j[10]))
        elif (len(j) == 16):
            num.append(int(j[10:12]))

    num.sort()

    for k in range(len(num)):
        for p in fileNames:

            if (len(p) == 15):
                if (int(p[10]) == num[k]):
                    sortNames.append(p)

            elif (len(p) == 16):
                if (int(p[10:12]) == num[k]):
                    sortNames.append(p)
    return sortNames, num



def time_avg(dirName,sizeName,INPUT,part,numSnap,eqStep,numPar,sampRat):
    """
    Calculates time average of density fluctuations and Kirkwood-Buff
    integrals as an average over all subvolumes.
    """

    inputSize = open(sizeName, "r")
    sizeData = np.loadtxt(inputSize)
    inputSize.close()
    L = float(sizeData[np.argmin(sizeData[:,0]),0])
    boxlo = float(sizeData[np.argmin(sizeData[:,0]),1])
    boxhi = float(sizeData[np.argmin(sizeData[:,0]),2])

    lattice = [int(x**3) for x in part]
    steps = numSnap/numPar
    limPar = [eqStep+steps*(y*sampRat) for y in range(1,numPar+1)]
    print(limPar)

    Gavg = []
    fluct = []

    if (len(INPUT) != len(part)):
        print("Vectors are of different length. Exiting")
        sys.exit()

    for i in range(len(INPUT)):

        fullfile = os.path.join(dirName,INPUT[i])
        inputFile = open(fullfile, "r")
        totData = [z for z in inputFile]
        data = totData[3:]
        inputFile.close()

        timeAvg = np.zeros((lattice[i]*numPar,5))
        tmpFluct = np.zeros((numPar,6))
        meanG = np.zeros((numPar,3))

        vol = (L/float(part[i]))**3

        box = -1
```

```
par = 0

for k in data:
    k = k.split()
    j = [float(l) for l in k]

    if (len(j) == 2 and j[0] == 20000000):
        break

    elif (len(j) == 2 and j[0] in limPar):
        par  += 1

    elif (len(j) == 6):
        if (j[3] > boxlo and j[3] < boxhi and j[4] > boxlo and \
            j[4] < boxhi and j[5] > boxlo and j[5] < boxhi):

            if (box == (par*lattice[i]-1)):
                box = lattice[i]*(par-1)
            elif (box < (par*lattice[i]-1)):
                box += 1

            n1 = float(j[1])
            n2 = float(j[2])
            n1n1 = float(j[1])*float(j[1])
            n2n2 = float(j[2])*float(j[2])
            n1n2 = float(j[1])*float(j[2])
            timeAvg[box] += [n1,n2,n1n1,n2n2,n1n2]

timeAvg = timeAvg/steps
tmpFluct[:,0] = L/part[i]
for n in range(numPar):
    tmpFluct[n,1:] = np.mean(timeAvg[n*lattice[i]: \
                                     (n+1)*lattice[i]], axis=0)

for m in range(numPar):
    for t in timeAvg[m*lattice[i]:(m+1)*lattice[i]]:
        G11 = vol*((t[2]-t[0]*t[0])/(t[0]*t[0]) - 1.0/t[0])
        G22 = vol*((t[3]-t[1]*t[1])/(t[1]*t[1]) - 1.0/t[1])
        G12 = vol*((t[4]-t[0]*t[1])/(t[0]*t[1]))

        meanG[m] += [G11,G22,G12]

meanG = meanG/lattice[i]
Gavg.append(meanG)
fluct.append(tmpFluct)
```

```
    return fluct, Gavg


if __name__ == "__main__":

    numPar = 5
    numSnap = 500000
    sampRat = 100
    eqStep = 0

    INPUT,part = search_files("dir_lattice")
    fluctRes, GavgRes  = time_avg("dir_lattice","size.out", \
                          INPUT,part,numSnap,eqStep,numPar,sampRat)
```

C.5 Fluctuations - Methanol-Water

```
# Master Project 2018
# Vilde Braten
# vildebr@stud.ntnu.no
# 14.04.2018

import numpy as np
import scipy.integrate
import os
import sys
import random


def calc_fluct_mol():
    """
    Reads LAMMS dump file consisting of methanol and water molecules
    and calculates density fluctuations.
    """

    delta = (maxR-minR)/bins
    radial = [minR + delta*(m+1) for m in range(bins)]

    com = np.zeros((Nmol,4))
    n1 = np.zeros(bins)
    n2 = np.zeros(bins)
    n1n1 = np.zeros(bins)
    n2n2 = np.zeros(bins)
    n1n2 = np.zeros(bins)

    random.seed(861924)
    counter = 0

    boxAvg = np.zeros((pts,6))

    # read dump file
    for snapshot in range(numSnap):

        # read N
        _ = inputFile.readline()
        _ = inputFile.readline()
        _ = inputFile.readline()
        line = inputFile.readline()
        N = int(line)
```

```
# read box dimensions
_ = inputFile.readline()
line = inputFile.readline()
_boxlo, _boxhi = line.split()
boxlo, boxhi = float(_boxlo), float(_boxhi)

boxL = boxhi-boxlo
hbox = boxL/2.0

# move forward by three lines
_ = inputFile.readline()
_ = inputFile.readline()
_ = inputFile.readline()

snapshot = []
for i in range(N):
    line = inputFile.readline()
    mol_id, atom_id, atom_type, pos_x, pos_y, pos_z \
            = line.split()
    snapshot.append([float(mol_id), float(atom_type), \
              float(pos_x), float(pos_y), float(pos_z)])

# calculating com of molecules
for l in range(8):
     for j in range(l*(Nm+Nw),l*(Nm+Nw)+Nm):
         CH3 = snapshot[j*3][2:]
         Om = snapshot[1+j*3][2:]
         Hm = snapshot[2+j*3][2:]

         com[j,0] = 1

         for k in range(3):
             if (Om[k]-CH3[k] > hbox):
                 CH3[k] += boxL
             elif (Om[k]-CH3[k] < -hbox):
                 CH3[k] -= boxL
             if (Om[k]-Hm[k] > hbox):
                 Hm[k] += boxL
             elif (Om[k]-Hm[k] < -hbox):
                 Hm[k] -= boxL

             com[j,k+1] = (Omass*Om[k] + CH3mass*CH3[k] \
                           + Hmass*Hm[k])/Mmass

             if (com[j,k+1] > boxhi):
                 com[j,k+1] -= boxL
             if (com[j,k+1] < boxlo):
                 com[j,k+1] += boxL

     for p in range(l*(Nm+Nw)+Nm,l*(Nm+Nw)+2*Nw):
         H1 = snapshot[p*3][2:]
         Ow = snapshot[1+p*3][2:]
         H2 = snapshot[2+p*3][2:]

         com[p,0] = 2

         for n in range(3):
             if (Ow[n]-H1[n] > hbox):
                 H1[n] += boxL
             elif (Ow[n]-H1[n] < -hbox):
                 H1[n] -= boxL
             if (Ow[n]-H2[n] > hbox):
                 H2[n] += boxL
             elif (Ow[n]-H2[n] < -hbox):
                 H2[n] -= boxL

             com[p,n+1] = (Omass*Ow[n] +  \
                           Hmass*(H1[n]+H2[n]))/Wmass

             if (com[p,n+1] > boxhi):
                 com[p,n+1] -= boxL
             if (com[p,n+1] < boxlo):
                 com[p,n+1] += boxL


# SSM
for m in range(rndSmp):
    rndx = boxlo + boxL*random.uniform(0,1)
    rndy = boxlo + boxL*random.uniform(0,1)
    rndz = boxlo + boxL*random.uniform(0,1)

    n_1 = np.zeros(bins)
    n_2 = np.zeros(bins)

    for molpos in com:
        moltype, dx, dy, dz = molpos
        dx -= rndx
        dy -= rndy
        dz -= rndz

        if (dx > hbox):
            dx -= boxL
        elif (dx < -hbox):
            dx += boxL

        if (dy > hbox):
            dy -= boxL
        elif (dy < -hbox):
            dy += boxL

        if (dz > hbox):
            dz -= boxL
        elif (dz < -hbox):
            dz += boxL

        dr = np.sqrt(dx*dx + dy*dy + dz*dz)

        for r in range(bins):
            if (dr < radial[r]):
                if (moltype == 1):
                    n_1[r] += 1.0
                elif (moltype == 2):
                    n_2[r] += 1.0

    for g in range(bins):
        n1[g] += n_1[g]
        n2[g] += n_2[g]
        n1n1[g] += n_1[g]*n_1[g]
        n2n2[g] += n_2[g]*n_2[g]
        n1n2[g] += n_1[g]*n_2[g]

    counter += 1.0


# LATTICE
for part in range(1,pts+1):
    numBox = part*part*part
    subBoxL = boxL/part
    vol = (boxL*boxL*boxL)/numBox
    c = [boxlo+subBoxL*w for w in range(part+1)]

    N_1 = np.zeros(numBox)
    N_2 = np.zeros(numBox)
    box_avg = np.zeros(6)

    for molpos in com:
        moltype, posx, posy, posz = molpos
        subBox = -1
```

```
                for x in range((len(c)-1)):
                    xlo = c[x]
                    xhi = c[x+1]
                    for y in range((len(c)-1)):
                        ylo = c[y]
                        yhi = c[y+1]
                        for z in range((len(c)-1)):
                            zlo = c[z]
                            zhi = c[z+1]
                            subBox += 1
                            if (posx < xhi and posx > xlo and \
                                posy < yhi and posy > ylo and \
                                posz < zhi and posz > zlo):

                                if (moltype == 1):
                                    N_1[subBox] += 1
                                elif (moltype == 2):
                                    N_2[subBox] += 1

            for b in range(numBox):
                N1 = N_1[b]
                N2 = N_2[b]
                N1N1 = N_1[b]*N_1[b]
                N2N2 = N_2[b]*N_2[b]
                N1N2 = N_1[b]*N_2[b]

                box_avg += [subBoxL,N1,N2,N1N1,N2N2,N1N2]

            box_avg /= numBox
            boxAvg[part-1,:] += box_avg


    # normalizing
    n1 /= counter
    n2 /= counter
    n1n1 /= counter
    n2n2 /= counter
    n1n2 /= counter

    boxAvg /= numSnap

    return radial,n1,n2,n1n1,n2n2,n1n2,boxAvg


if __name__ == "__main__":
```

```
minR = 1.0
maxR = 20.0
bins = 200
pts = 8
rndSmp = 10
numSnap = 100000

Nmol = 8000
Nm = 500
Nw = 500

CH3mass = 15.0347
Hmass = 1.008
Omass = 15.9994
Wmass = Omass+2*Hmass
Mmass = CH3mass+Hmass+Omass

with open("dump.out", "r") as inputFile:
    radial,n1,n2,n1n1,n2n2,n1n2,boxAvg = calc_fluct_mol()
```

D LAMMPS Scripts

The following sections contain LAMMPS input scrips used for simulations and geometry files for the methanol and water molecules. For the LJ mixtures, only the scripts used for the real LJ mixture are presented as examples. This includes the initialization scrips in section D.1, the main runscript in section D.2, the rerunscript in section D.3 and the script used to generate data at different pressures used for computation of isothermal compressibility in section D.4. The scrips given for the methanol-water mixture are the initialization scrips in section D.5, the main run script in section D.6, the rerun script in section D.7 and the script used to generate data at different pressures used for computation of isothermal compressibility in section D.8. In addition, the geometry file for the methanol molecules is presented in section D.9, while the geometry file for the water molecules is presented in section D.10.

D.1 Initialization Script - LJ Mixtures

```
# Master Project 2018
# Vilde Bråten
# vildebr@stud.ntnu.no
# 26.01.2018

variable       L equal 500.0

units          lj
atom_style     full

region         box block 0.0 ${L} 0.0 ${L} 0.0 ${L} units box

create_box     2 box

create_atoms   1 random 20000 749302 NULL units box

mass           * 1.0

group          part1 type 1
group          part2 type 2

velocity       all create 3.0 87287

pair_style     lj/cut 3.0
pair_coeff     1 1 1.0 1.0
pair_coeff     1 2 0.7726 0.8838
pair_coeff     2 2 0.5968 0.7677
pair_modify    tail yes
```

```
minimize        0.0 0.0 1000000 1000000

neighbor        0.5 bin
neigh_modify    every 10 delay 0 check yes
timestep        0.002

fix             ens all npt temp 0.92825 0.92835 0.2 iso 0.4721 0.4721 2

thermo_style    custom step time ke pe etotal press lx xlo xhi density
thermo          10000

run             10000000

write_data      config/initconf
```

D.2 Main Runscript - LJ Mixture

```
# Master Project 2018
# Vilde Bråten
# vildebr@stud.ntnu.no
# 26.01.2018

units           lj
atom_style      full

pair_style      lj/cut 3.0

read_data       config/eqconfig

variable        Lx equal lx
variable        x0 equal xlo
variable        x1 equal xhi
variable        y0 equal ylo
variable        y1 equal yhi
variable        z0 equal zlo
variable        z1 equal zhi
variable        Nevery equal 100
variable        Nrepeat equal 100000
variable        Nfreq equal 10000000
variable        parStep equal 10000000

mass            * 1.0

group           part1 type 1
group           part2 type 2

pair_coeff      1 1 1.0 1.0
pair_coeff      1 2 0.7726 0.8838
pair_coeff      2 2 0.5968 0.7677
pair_modify     tail yes

neighbor        0.5 bin
neigh_modify    every 10 delay 0 check yes
timestep        0.0036

fix             ens all npt temp 0.92835 0.92835 0.36
                iso 0.4721 0.4721 3.6

fix             size all print 100 "${Lx} ${x0} ${x1} ${y0} ${y1}
                ${z0} ${z1} " file postprocessing/size.out screen no
```

```
compute         temp all temp
compute         pres all pressure temp
compute         ke all ke
compute         pe all pe

fix             calcH all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                c_ke c_pe c_pres c_temp file postprocessing/pres_en.out

fix             volH all ave/time ${Nevery} ${Nrepeat}  ${Nfreq}
                v_Lx ave running file postprocessing/volH.out

thermo_style    custom step time ke pe etotal press lx density
thermo          10000

dump            totSSM all custom 100 dump_files/SSM.dump id type x y z
dump            gSSM all custom 10000 dump_files/gSSM.dump id type x y z

dump            SSMdump1 all custom 100 dump_files/SSM1.dump
                id type x y z
run             ${parStep}
undump          SSMdump1

dump            SSMdump2 all custom 100 dump_files/SSM1.dump
                id type x y z
run             ${parStep}
undump          SSMdump2

dump            SSMdump3 all custom 100 dump_files/SSM1.dump
                id type x y z
run             ${parStep}
undump          SSMdump3

dump            SSMdump4 all custom 100 dump_files/SSM1.dump
                id type x y z
run             ${parStep}
undump          SSMdump4

dump            SSMdump5 all custom 100 dump_files/SSM1.dump
                id type x y z
run             ${parStep}
undump          SSMdump5

write_data      config/endconfig
```

D.3 Rerunscript - LJ Mixtures

```
# Master Project 2018
# Vilde Bråten
# vildebr@stud.ntnu.no
# 26.01.2018

units          lj
atom_style     full

pair_style     lj/cut 3.0

read_data      config/eqconfig

variable       Lx equal 29.3427531845
variable       x0 equal 235.3286234077
variable       x1 equal 264.6713765922
variable       L2 equal ${Lx}/2.0
variable       L3 equal ${Lx}/3.0
variable       L4 equal ${Lx}/4.0
variable       L5 equal ${Lx}/5.0
variable       L6 equal ${Lx}/6.0
variable       L7 equal ${Lx}/7.0
variable       L8 equal ${Lx}/8.0

mass           * 1.0

pair_style     zero ${L2}
pair_coeff     * *

group          part1 type 1
group          part2 type 2

neighbor       0.5 bin
neigh_modify   every 10 delay 0 check yes page 110000 one 11000

compute        cube1 all chunk/atom bin/3d
               x ${x0} ${Lx} y ${x0} ${Lx} z ${x0} ${Lx}
               bound x ${x0} ${x1} bound y ${x0} ${x1}
               bound z ${x0} ${x1} units box
compute        1count1 part1 property/chunk cube1 count
compute        2count1 part2 property/chunk cube1 count
compute        size1 all property/chunk cube1 coord1 coord2 coord3
fix            totcount1 all ave/time 100 1 100
               c_1count1[*] c_2count1[*] c_size1[*]
               file postprocessing/dir_lattice/latticeSSM1.out
               mode vector

compute        cube2 all chunk/atom bin/3d
               x ${x0} ${L2} y ${x0} ${L2} z ${x0} ${L2}
               bound x ${x0} ${x1} bound y ${x0} ${x1}
               bound z ${x0} ${x1} units box
compute        1count2 part1 property/chunk cube2 count
compute        2count2 part2 property/chunk cube2 count
compute        size2 all property/chunk cube2 coord1 coord2 coord3
fix            totcount2 all ave/time 100 1 100
               c_1count2[*] c_2count2[*] c_size2[*]
               file postprocessing/dir_lattice/latticeSSM2.out
               mode vector

compute        cube3 all chunk/atom bin/3d
               x ${x0} ${L3} y ${x0} ${L3} z ${x0} ${L3}
               bound x ${x0} ${x1} bound y ${x0} ${x1}
               bound z ${x0} ${x1} units box
compute        1count3 part1 property/chunk cube3 count
compute        2count3 part2 property/chunk cube3 count
compute        size3 all property/chunk cube1 coord1 coord2 coord3
fix            totcount3 all ave/time 100 1 100
               c_1count3[*] c_2count3[*] c_size3[*]
               file postprocessing/dir_lattice/latticeSSM3.out
               mode vector

compute        cube4 all chunk/atom bin/3d
               x ${x0} ${L4} y ${x0} ${L4} z ${x0} ${L4}
               bound x ${x0} ${x1} bound y ${x0} ${x1}
               bound z ${x0} ${x1} units box
compute        1count4 part1 property/chunk cube4 count
compute        2count4 part2 property/chunk cube4 count
compute        size4 all property/chunk cube1 coord1 coord2 coord3
fix            totcount4 all ave/time 100 1 100
               c_1count4[*] c_2count4[*] c_size4[*]
               file postprocessing/dir_lattice/latticeSSM4.out
               mode vector

compute        cube5 all chunk/atom bin/3d
               x ${x0} ${L5} y ${x0} ${L5} z ${x0} ${L5}
               bound x ${x0} ${x1} bound y ${x0} ${x1}
               bound z ${x0} ${x1} units box
compute        1count5 part1 property/chunk cube5 count
compute        2count5 part2 property/chunk cube5 count
compute        size5 all property/chunk cube1 coord1 coord2 coord3
```

```
fix             totcount5 all ave/time 100 1 100
                c_1count5[*] c_2count5[*] c_size5[*]
                file postprocessing/dir_lattice/latticeSSM5.out
                mode vector

compute         cube6 all chunk/atom bin/3d
                x ${x0} ${L6} y ${x0} ${L6} z ${x0} ${L6}
                bound x ${x0} ${x1} bound y ${x0} ${x1}
                bound z ${x0} ${x1} units box
compute         1count6 part1 property/chunk cube6 count
compute         2count6 part2 property/chunk cube6 count
compute         size6 all property/chunk cube1 coord1 coord2 coord3
fix             totcount6 all ave/time 100 1 100
                c_1count6[*] c_2count6[*] c_size6[*]
                file postprocessing/dir_lattice/latticeSSM6.out
                mode vector

compute         cube7 all chunk/atom bin/3d
                x ${x0} ${L7} y ${x0} ${L7} z ${x0} ${L7}
                bound x ${x0} ${x1} bound y ${x0} ${x1}
                bound z ${x0} ${x1} units box
compute         1count7 part1 property/chunk cube7 count
compute         2count7 part2 property/chunk cube7 count
compute         size7 all property/chunk cube1 coord1 coord2 coord3
fix             totcount7 all ave/time 100 1 100
                c_1count7[*] c_2count7[*] c_size7[*]
                file postprocessing/dir_lattice/latticeSSM7.out
                mode vector

compute         cube8 all chunk/atom bin/3d
                x ${x0} ${L8} y ${x0} ${L8} z ${x0} ${L8}
                bound x ${x0} ${x1} bound y ${x0} ${x1}
                bound z ${x0} ${x1} units box
compute         1count8 part1 property/chunk cube8 count
compute         2count8 part2 property/chunk cube8 count
compute         size8 all property/chunk cube8 coord1 coord2 coord3
fix             totcount8 all ave/time 100 1 100
                c_1count8[*] c_2count8[*] c_size8[*]
                file postprocessing/dir_lattice/latticeSSM8.out
                mode vector

compute         rdf150 all rdf 150 1 1 2 2 1 2
compute         rdf1000 all rdf 1000 1 1 2 2 1 2
compute         rdf2000 all rdf 2000 1 1 2 2 1 2

fix             rdfcomp150 all ave/time 100 100000 10000000
```

```
               c_rdf150[*] file postprocessing/rdf150/rdf150.out
               mode vector

fix            rdfcomp1000 all ave/time 100 100000 10000000
               c_rdf1000[*] file postprocessing/rdf1000/rdf1000.out
               mode vector

fix            rdfcomp2000 all ave/time 100 100000 10000000
               c_rdf2000[*] file postprocessing/rdf2000/rdf2000.out
               mode vector

thermo         100000

rerun          dump_files/SSM.dump first 0 every 1 dump x y z
               box yes scaled yes
```

D.4 Script Creating Different Pressure Data - LJ Mixtures

```
# Master Project 2018
# Vilde Bråten
# vildebr@stud.ntnu.no
# 03.04.2018

units          lj
atom_style     full

pair_style     lj/cut 3.0

read_data      config/endconfig

variable       eqStep equal 1000000
variable       parStep equal 2000000
variable       start67 equal ${eqStep}+1*${parStep}
variable       start87 equal ${eqStep}+2*${parStep}
variable       start107 equal ${eqStep}+3*${parStep}
variable       Nevery equal 100
variable       Nrepeat equal 2000
variable       Nfreq equal 200000
variable       Lx equal lx

mass           * 1.0

group          part1 type 1
group          part2 type 2

pair_coeff     1 1 1.0 1.0
pair_coeff     1 2 0.7726 0.8838
pair_coeff     2 2 0.5968 0.7677
pair_modify    tail yes

neighbor       0.5 bin
neigh_modify   every 10 delay 0 check yes
timestep       0.0036

thermo_style   custom step time ke pe etotal press lx density
thermo         10000

compute        temp all temp
compute        pres all pressure temp

fix            pres47 all npt temp 0.92835 0.92835 0.36
```

```
               iso 0.4721 0.4721 3.6
fix            pt47 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               c_pres c_temp start ${eqStep} file compress/p47.out
fix            size47 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               v_Lx start ${eqStep} ave running
               file compress/size47.out

run            ${parStep}
unfix          pres47
unfix          pt47
unfix          size47

fix            pres67 all npt temp 0.92835 0.92835 0.36
               iso 0.6721 0.6721 3.6
fix            pt67 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               c_pres c_temp start ${start67} file compress/p67.out
fix            size67 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               v_Lx start ${start67} ave running
               file compress/size67.out

run            ${parStep}
unfix          pres67
unfix          pt67
unfix          size67

fix            pres87 all npt temp 0.92835 0.92835 0.36
               iso 0.8721 0.8721 3.6
fix            pt87 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               c_pres c_temp start ${start87} file compress/p87.out
fix            size87 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               v_Lx start ${start87} ave running
               file compress/size87.out

run            ${parStep}
unfix          pres87
unfix          pt87
unfix          size87

fix            pres107 all npt temp 0.92835 0.92835 0.36
               iso 1.0721 1.0721 3.6
fix            pt107 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               c_pres c_temp start ${start107} file compress/p107.out
fix            size107 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               v_Lx start ${start107} ave running
               file compress/size107.out
```

```
run             ${parStep}
unfix           pres107
unfix           pt107
unfix           size107

write_data      config/compressconfig
```

D.5 Initialization Script - Methanol-Water Mixture

```
# Master Project
# Vilde Bråten
# 05.04.2018

atom_style      full
units           real

lattice         fcc 20
region          box block 0 1000 0 1000 0 1000 units box

create_box      5 box bond/types 3 angle/types 2 extra/bond/per/atom 2
                extra/angle/per/atom 1 extra/special/per/atom 2

mass            1 15.0347
mass            2 15.9994
mass            3 1.008
mass            4 1.008
mass            5 15.9994

pair_style      lj/cut/coul/long 15
pair_coeff      1 1 0.2073 3.748
pair_coeff      2 2 0.1555 3.192
pair_coeff      3 3 0.0210 1.580
pair_coeff      4 4 0.0    0.0
pair_coeff      5 5 0.1555 3.166

pair_coeff      1 2 0.1795 3.459
pair_coeff      1 3 0.0660 2.434
pair_coeff      1 4 0.0    0.0
pair_coeff      1 5 0.1795 3.445

pair_coeff      2 3 0.0572 2.246
pair_coeff      2 4 0.0    0.0
pair_coeff      2 5 0.1555 3.179

pair_coeff      3 4 0.0    0.0
pair_coeff      3 5 0.0572 2.237

pair_coeff      4 5 0.0    0.0

bond_style      harmonic
bond_coeff      1 0.0 1.430
bond_coeff      2 0.0 0.945
bond_coeff      3 0.0 1.0

angle_style     harmonic
angle_coeff     1 0.0 108.49
angle_coeff     2 0.0 109.47

kspace_style    ewald 1.0E-5
pair_modify     tail yes

neighbor        2.0 bin
neigh_modify    every 1 delay 0 check yes

molecule        methanol in.methanol
molecule        water in.water
create_atoms    0 random 400 638543 NULL mol methanol 648376 units box
create_atoms    0 random 600 297027 NULL mol water 193756 units box

set             type 1 charge 0.30
set             type 2 charge -0.82
set             type 3 charge 0.52
set             type 4 charge 0.4238
set             type 5 charge -0.8476

velocity        all create 300.0 648376

fix             constrain all shake 1.0e-4 100 0 b 1 2 3 a 1 2
fix             6 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
fix             7 all momentum 100 linear 1 1 1

thermo_style    custom step density temp etotal lx press
thermo          100000

run             10000000

write_data      config.molmix
```

D.6 Main Runscript - Methanol-Water Mixture

```
# Master Project
# Vilde Braten
# 05.04.2018

atom_style      full
units           real

pair_style      lj/cut/coul/long 15
bond_style      harmonic
angle_style     harmonic
read_data       repconfig.molmix

variable        x0 equal xlo
variable        x1 equal xhi
variable        Lx equal lx
variable        L2 equal ${Lx}/2.0
variable        Nevery equal 100
variable        Nrepeat equal 100000
variable        Nfreq equal 10000000

mass            1 15.0347
mass            2 15.9994
mass            3 1.008
mass            4 1.008
mass            5 15.9994

pair_coeff      1 1 0.2073 3.748
pair_coeff      2 2 0.1555 3.192
pair_coeff      3 3 0.0210 1.580
pair_coeff      4 4 0.0    0.0
pair_coeff      5 5 0.1555 3.166

pair_coeff      1 2 0.1795 3.459
pair_coeff      1 3 0.0660 2.434
pair_coeff      1 4 0.0    0.0
pair_coeff      1 5 0.1795 3.445

pair_coeff      2 3 0.0572 2.246
pair_coeff      2 4 0.0    0.0
pair_coeff      2 5 0.1555 3.179

pair_coeff      3 4 0.0    0.0
pair_coeff      3 5 0.0572 2.237
pair_coeff      4 5 0.0    0.0

bond_style      harmonic
bond_coeff      1 0.0 1.430
bond_coeff      2 0.0 0.945
bond_coeff      3 0.0 1.0

angle_style     harmonic
angle_coeff     1 0.0 108.49
angle_coeff     2 0.0 109.47

kspace_style    ewald 1.0E-5
pair_modify     tail yes

neighbor        2.0 bin
neigh_modify    every 1 delay 0 check yes

set             type 1 charge 0.30
set             type 2 charge -0.82
set             type 3 charge 0.52
set             type 4 charge 0.4238
set             type 5 charge -0.8476

fix             constrain all shake 1.0e-4 100 0 b 1 2 3 a 1 2
fix             6 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
fix             7 all momentum 100 linear 1 1 1

compute         temp all temp
compute         pres all pressure temp
compute         ke all ke
compute         pe all pe

fix             par all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                c_pres c_temp c_ke c_pe v_Lx file parameters.out

dump            alldump all custom 100 dump.out mol id type x y z
dump_modify     alldump sort id

thermo_style    custom step temp etotal lx press density
thermo          100000

run             ${Nfreq}

write_data      endconfig.molmix
```

D.7 Rerunscript - Methanol-Water mixture

```
# Master Project 2018
# Vilde Braten
# vildebr@stud.ntnu.no
# 26.01.2018


atom_style      full
units           real

pair_style      lj/cut/coul/long 15
bond_style      harmonic
angle_style     harmonic
read_data       eqconfig.molmix

variable        x0 equal xlo
variable        x1 equal xhi
variable        Lx equal lx
variable        hbox equal ${Lx}/2.0

mass            1 15.0347
mass            2 15.9994
mass            3 1.008
mass            4 1.008
mass            5 15.9994

pair_style      zero ${hbox}
pair_coeff      * *

neighbor        2.0 bin
neigh_modify    every 10 delay 0 check yes page 1100000 one 110000

compute         rdf all rdf 3000 2 2 5 5 2 5
fix             rdfcomp all ave/time 100 100000 10000000 c_rdf[*]
                file rdf.out mode vector

thermo          10000

rerun           dump.out first 100 every 100 dump x y z box yes
                scaled yes
```

D.8 Script Creating Different Pressure Data - Methanol-Water Mixture

```
# Master Project
# Vilde Braten
# 05.04.2018

atom_style      full
units           real

pair_style      lj/cut/coul/long 15
bond_style      harmonic
angle_style     harmonic
read_data       eqconfig.molmix

variable        eqStep equal 1000000
variable        parStep equal 2000000
variable        start67 equal ${eqStep}+1*${parStep}
variable        start87 equal ${eqStep}+2*${parStep}
variable        start107 equal ${eqStep}+3*${parStep}
variable        Nevery equal 100
variable        Nrepeat equal 2000
variable        Nfreq equal 200000
variable        Lx equal lx

mass            1 15.0347
mass            2 15.9994
mass            3 1.008
mass            4 1.008
mass            5 15.9994

pair_coeff      1 1 0.2073 3.748
pair_coeff      2 2 0.1555 3.192
pair_coeff      3 3 0.0210 1.580
pair_coeff      4 4 0.0    0.0
pair_coeff      5 5 0.1555 3.166

pair_coeff      1 2 0.1795 3.459
pair_coeff      1 3 0.0660 2.434
pair_coeff      1 4 0.0    0.0
pair_coeff      1 5 0.1795 3.445

pair_coeff      2 3 0.0572 2.246
pair_coeff      2 4 0.0    0.0
pair_coeff      2 5 0.1555 3.179
pair_coeff      3 4 0.0    0.0
pair_coeff      3 5 0.0572 2.237

pair_coeff      4 5 0.0    0.0

bond_style      harmonic
bond_coeff      1 0.0 1.430
bond_coeff      2 0.0 0.945
bond_coeff      3 0.0 1.0

angle_style     harmonic
angle_coeff     1 0.0 108.49
angle_coeff     2 0.0 109.47

kspace_style    ewald 1.0E-5
pair_modify     tail yes

neighbor        2.0 bin
neigh_modify    every 1 delay 0 check yes

set             type 1 charge 0.30
set             type 2 charge -0.82
set             type 3 charge 0.52
set             type 4 charge 0.4238
set             type 5 charge -0.8476

fix             constrain all shake 1.0e-4 100 0 b 1 2 3 a 1 2
fix             6 all npt temp 300.0 300.0 100 iso 1.0 1.0 1000
fix             7 all momentum 100 linear 1 1 1

compute        temp all temp
compute        pres all pressure temp

fix            pres1 all npt temp 300.0 300.0 100
               iso 1.0 1.0 1000
fix            pt1 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               c_pres c_temp start ${eqStep} file compress/p1.out
fix            size1 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
               v_Lx start ${eqStep} ave running
               file compress/size1.out

run            ${parStep}
unfix          pres1
unfix          pt1
unfix          size1
```

```
fix             pres5 all npt temp 300.0 300.0 100
                iso 5.0 5.0 1000
fix             pt5 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                c_pres c_temp start ${start5} file compress/p5.out
fix             size5 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                v_Lx start ${start5} ave running
                file compress/size5.out

run             ${parStep}
unfix           pres5
unfix           pt5
unfix           size5

fix             pres10 all npt temp 300.0 300.0 100
                iso 10.0 10.0 1000
fix             pt10 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                c_pres c_temp start ${start10} file compress/p10.out
fix             size10 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                v_Lx start ${start10} ave running
                file compress/size10.out

run             ${parStep}
unfix           pres10
unfix           pt10
unfix           size10

fix             pres15 all npt temp 300.0 300.0 100
                iso 15.0 15.0 1000
fix             pt15 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                c_pres c_temp start ${start15} file compress/p15.out
fix             size15 all ave/time ${Nevery} ${Nrepeat} ${Nfreq}
                v_Lx start ${start15} ave running
                file compress/size15.out

run             ${parStep}
unfix           pres15
unfix           pt15
unfix           size15

thermo_style     custom step temp etotal lx press density
thermo           100000
run              ${Nfreq}
```

D.9 Geometry File - Methanol

```
# LAMMPS

3 atoms
2 bonds
1 angles

Coords
1 1.55000 0.27954 2.15638
2 1.55000 1.55000 1.50000
3 1.55000 2.22754 2.15638

Types
1 1
2 2
3 3

Charges
1 0.30
2 -0.82
3 0.52

Bonds
1 1 1 2
2 2 2 3

Angles
1 1 1 2 3

Special Bond Counts
1 1 1 0
2 2 0 0
3 1 1 0

Special Bonds
1 2 3
2 1 3
3 2 1

Shake Flags
1 2
2 1
3 2
```

```
Shake Atoms
1 1 2
2 2 1 3
3 2 3

Shake Bond Types
1 1
2 1 2 1
3 2
```

D.10 Geometry File - Water

```
# LAMMPS

3 atoms
2 bonds
1 angles

Coords
1 1.55000 2.36649 2.07736
2 1.55000 1.55000 1.50000
3 1.55000 0.73351 2.07736

Types
1 4
2 5
3 4

Charges
1 0.4238
2 -0.8476
3 0.4238

Bonds
1 3 1 2
2 3 2 3

Angles
1 2 1 2 3

Special Bond Counts
1 1 1 0
2 2 0 0
3 1 1 0

Special Bonds
1 2 3
2 1 3
3 2 1

Shake Flags
1 2
2 1
3 2
```

```
Shake Atoms
1 1 2
2 2 1 3
3 2 3

Shake Bond Types
1 1
2 1 1 1
3 1
```