MOLECULAR SIMULATION OF SIMPLE FLUIDS AND POLYMERS IN NANOCONFINEMENT

By

CHRISTOPHER JOHN RASMUSSEN

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Chemical and Biochemical Engineering

written under the direction of

Alexander V. Neimark

and approved by

____

____

____

____

New Brunswick, New Jersey

October 2012

ABSTRACT OF THE DISSERTATION

Molecular Simulation of Simple Fluids and Polymers in Nanoconfinement

By CHRISTOPHER JOHN RASMUSSEN

Dissertation Director:

Alexander V. Neimark

Prediction of phase behavior and transport properties of simple fluids and polymers confined to nanoscale pores is important to a wide range of chemical and biochemical engineering processes. A practical approach to investigate nanoscale systems is molecular simulation, specifically Monte Carlo (MC) methods. One of the most challenging problems is the need to calculate chemical potentials in simulated phases. Through the seminal work of Widom, practitioners have a powerful method for calculating chemical potentials. Yet, this method fails for dense and inhomogeneous systems, as well as for complex molecules such as polymers. In this dissertation, the gauge cell MC method, which had previously been successfully applied to confined simple fluids, was employed and extended to investigate nanoscale fluids in several key areas. Firstly, the process of cavitation (the formation and growth of bubbles) during desorption of fluids from nanopores was investigated. The dependence of cavitation pressure on pore size was determined with gauge cell MC calculations of the nucleation barriers correlated with experimental data. Additional computational studies elucidated the role of surface defects and pore connectivity in the formation of cavitation bubbles. Secondly, the gauge cell method was extended to polymers. The method was verified against the literature results and found significantly more efficient. It was used to examine adsorption of polymers in nanopores. These results were applied to model the dynamics of translocation, the act of a polymer threading through a small opening, which is implicated in drug packaging and delivery, and DNA sequencing. Translocation dynamics was studied as diffusion along the free energy landscape. Thirdly, we show how computer simulation of polymer adsorption could shed light on the specifics of polymer chromatography, which is a key tool for the analysis and purification of polymers. The quality of separation depends on the physico-chemical mechanisms of polymer/pore interaction. We considered liquid chromatography at critical conditions, and calculated the dependence of the partition coefficient on chain length. Finally, solvent-gradient chromatography was modeled using a statistical model of polymer adsorption. A model for predicting separation of complex polymers (with functional groups or copolymers) was developed for practical use in chromatographic separations.

Acknowledgements

This thesis would not have been possible without the help of many people. First and foremost, I thank my academic advisor, Dr. Alexander V. Neimark, for his constant support, limitless patience, and guidance. Thanks to Dr. Aleksey Vishnyakov, who taught me all of the simulation skills required for this work, Dr. Yefim Brun, an expert in polymer chromatography who graciously hosted me in his lab at DuPont for a summer, Dr. Stephen Garofalini for serving on my committee and for valuable advice during my thesis proposal, and Dr. Yee Chiew, whose excellent teaching made graduate school a possibility at all.

The work presented here is of course a collaborative effort, and in no way could be accomplished on my own. I give my utmost thanks to my co-authors: For the first cavitation work [1], Dr. Matthias Thommes from Quantachrome Instruments in Boyton Beach, Florida; Dr. Bernd Smarsly from Justus-Liebig University in Giessen, Germany; Dr. Freddy Kleitz from Université Laval in Quebec, Canada. For the second [2] and third [3] cavitation works: Dr. Gennady Gor, currently at Princeton University in Princeton, New Jersey. Finally, thanks to Brian McCauley at DuPont Corporate Center for Analytical Sciences, who measured all chromatography samples in this work, and in our paper [4].

I thank continued support from Department of Chemical and Biochemical Engineering, which started my first year of undergraduate studies and continued to this day. I especially thank Debbie Moon, Lynn DeCaprio, Ursula Wolf, Charanjeet Kaur, Joseline Williamson, Aruna Chadda, and Kirk Tarabokia.

Thanks to my fellow group mates, past and present: Dr. John Landers, Dr. Gennady Gor, Ming-Tsung Lee, Dr. Shuang Yang, Dr. Kan Yang, and Rich Cimino.

Thanks to my friends here at Rutgers, who were a constant source of support: Kapil Deshpande, Dr. Carolyn Waite Federici, Dr. Eric Jayjock, Dr. Athanas Koynov, Sara Koynov, Dr. Keirnan LaMarche, Dr. Matt Metzger, Dr. Frank Romanski, and Dr. Alice Vasilenko.

Finally, I thank my wife Phyllis, my parents John and Carolyn, and my high school chemistry teacher Mrs. Anne McCormick, who set me on the path towards chemical engineering, rather than biology, or worst of all, law.

Table of contents

List of tables

List of figures

Chapter 1

Introduction

1.1 Motivation

Fluids confined to nanoscale pores exhibit very different physics and thermodynamics than their bulk counterpart. This is due in part to strong adsorptive forces (which give rise to the capillary effect [5, 6]), and high surface to volume aspect ratio [7]. Understanding how these factors affect macroscale properties of materials is crucial for their successful design, characterization, and production. Because of the characteristic scale of such systems (≤ 1 nm to 50 nm is typical), is difficult to observe the relevant mechanics that are necessary to fully describe and understand such systems. Atomistic simulation can help fill this gap of knowledge. Methods such as molecular dynamics (MD) and Monte Carlo (MC) simulations model atoms or groups of atoms as particles that obey classical mechanics, and interact with each other by empirical potentials. These methods give immense detail on the mechanics and thermodynamics of nanoscale systems.

A primary goal of molecular simulation is the calculation of a component's chemical potential and free energy. The free energy is a thermodynamic state function, and its minima define stable and metastable states within the system. Thus, its prediction is crucial when dealing with vapor-liquid phase equilibria, nucleation, protein folding, crystal topography, and polymer chromatography and other physio-chemical separation techniques. Widom [8] formulated a powerful method to "measure" the chemical potential in molecular simulation. However, this method fails at high densities and for extremely inhomogeneous phases, as well as for complex molecules such as polymers. One method developed to overcome this drawback is the mesocanonical ensemble, also known as the gauge cell method [9, 10]. It is the application and extension of the gauge cell method that is the key to this dissertation work.

The first part of this dissertation focuses on the behavior of fluids in nanopores, specifically relating to the formation of cavitation bubbles and their role in the desorption of fluid from pores. Cavitation is the spontaneous formation and growth of bubbles in a liquid. It is observed when a confined liquid is significantly blocked upon desorption, as occurs in pores with very narrow connecting necks; as the external pressure is reduced, the confined liquid becomes metastable and eventually reaches its fracture point. This is characterized by a specific desorption pressure on the isotherm. In this work, a combined experimental and simulation approach studies the relationship of pore size on the pressure at which cavitation occurs. Gauge cell MC studies determine the nucleation barrier of the critical cavitation bubble. Additional computational studies examine the role of heterogeneous defects of the pore wall on the cavitation pressure, and how overlapping spherical pores affect the mechanism of desorption and pressure of cavitation.

The second part of this dissertation extends the gauge cell to polymer chains. Because of the general complexity of a polymer molecule, the Widom method fails completely. Many ingenious approaches have been suggested to overcome this, but they each have drawbacks. We extend the gauge cell method by utilizing the incremental approach of Kumar et al. [11], which calculates the chemical potential in increments of single monomers. Kumar's approach enables calculation of the chemical potential for chain molecules, but with the same caveats as Widom's method, i.e. it fails for high density and highly structured fluids. The gauge cell helps overcome these problems. The new method, called the incremental gauge cell, is derived, implemented, and validated, and found to be an order of magnitude more efficient. We use the new method to study the adsorption of chains in nanopores. This study is further extended to investigate translocation, or the threading of a polymer chain through a very small hole. Translocation is implicated in many important biological processes, such as DNA/RNA transport, movement of proteins and polypeptides through membranes, the mechanism of viral attack on cells, advanced drug packaging and delivery, and DNA and protein sequencing. We study translocation dynamics by modeling it as one-dimensional diffusion along the free energy landscape, which is calculated by the incremental gauge cell method.

The third part of this dissertation deals with the practical problem of polymer separation. A common technique for polymer purification and separation is polymer chromatography [12]. Traditionally, separation is achieved by steric interaction of dissolved polymers with a porous substrate. Thus, polymers are separated according to their hydrodynamic volume. Interaction chromatography, which separates by their adsorptive interaction with the substrate, is limited to relatively short chains, as the interaction scales strongly with length. When these two phenomena (steric repulsion of large molecules in pores, and enthalpic attraction to the porous substrate) are balanced, the critical point of adsorption (CPA) is reached. At CPA, molecules are not separated by size (i.e. length), but by more subtle structural motifs, such as topology, the presence of functional groups, and monomer composition. Operating a chromatography column at such conditions (called liquid chromatography at critical conditions, or LCCC) is difficult, and there are no general models capable of predicting a polymer's CPA given solvent, substrate, and temperature. To this end, we show how the incremental gauge cell method might be used to predict CPA for a given set of conditions. A practical alternative to LCCC is the so-called gradient-elution at critical point of adsorption (GE-CPA) chromatography [13]. In this approach, the solvent quality is varied at a know rate, so that the dissolved polymers pass through the critical point. Using this method, one can separate complex polymers by structure and not weight. However, because a detailed theory of the separation mechanism does not exist, interpreting results is ambiguous, and designing experiments is a difficult, trial-by-error process. For these reasons, an extension of Brun's GE-CPA model [14, 15] was developed for complex polymers, and was tested against experimental results for functionalized polymers.

The rest of this dissertation is organized as follows. A short, general background on adsorption phenomena, polymer science, and molecular simulation is given in Section 1.2. Chapter 2 is a review of previous methods used to calculate the chemical potential of polymers in MC simulation, and their respective strengths and weaknesses. Chapter 3 contains three works on cavitation in nanopores: Section 3.1 details a combined experimental and simulation approach that determined the dependence of nitrogen cavitation pressures on the size of spherical silica pores, while gauge cell MC studies estimated the corresponding nucleation barrier. Section 3.2 uses a heterogeneous adsorption potential to model a surface defect, and the effect of the defect's size on cavitation pressure is discussed. Section 3.3 connects two adjacent spherical pores, and quantifies how the degree of their overlap affects cavitation as the desorption mechanism. Chapter 4 focuses on a new method that was developed to calculate the chemical potential of polymers, the incremental gauge cell method, and its practical applications. Section 4.1 gives a detailed derivation and discusses its foundation in the gauge cell method, and initial results of polymer adsorption in spherical pores. Section 4.2 examines polymer translocation into an adsorbing pore using the new method. Section 4.3 uses a statical model for the adsorption of complex polymers to model gradient-elution chromatography. Conclusions and the potential for future work is discussed in Chapter 5. Finally, the appendices give additional, detailed derivations of the incremental gauge cell method (A), as well as the code used for most MC work (B) and for the gradient-elution work (C).

1.2 Background

1.2.1 Behavior of fluids in nanoconfinement

The physics of nanoconfined fluids has been a long and well studied problem [16–20]. This work centers around the physical process known as *adsorption*, the accumulation of fluid particles from one phase onto a solid that is in contact with said phase. There are many practical consequences of this process, such as separations, purification, catalysis, nanofluidics and drug delivery. The primary application, and the motivation for much of the scientific research on the topic, is material characterization. When a fluid is in contact with a most solids, the fluid is adsorbs onto the solid's surface. A volume where the adsorbed fluid (*adsorbate*) is surrounded by at least two solid walls is considered a pore (the material containing pores is the *adsorbent*). Adsorption in pores is more energetically favorable, as the surface area of attractive pore walls increases. As the pore volume decreases (e.g. the two walls are brought closer to each other), the more strongly the fluid adsorbs, for the same reason. The interest in nanopores (where the distance between pore walls, or the *pore diameter*, is ∼0.5–100 nm) is due to the fact that at this scale, the number of molecules at the solid-fluid interface is on the same order as total molecules in the system, and thus its thermodynamic properties can be expected to deviate markedly from the bulk properties. Because these properties are a product of the solid-fluid interface, the pore's geometry and size can be expected to influence the final state of the system as well.

The amount of fluid adsorbed depends on the compositions of the adsorbate and adsorbent, as well as the temperature and partial pressure of the external bulk fluid which is in contact with the porous material. Generally speaking, as gas molecules are adsorbed, a liquid film forms on the pore walls. This is indicated on an isotherm by a step up in density at low pressures. This layer grows thicker as external pressure increases. Though the layer is considered a liquid, thermodynamically speaking, the fluid in the pore is considered as a "vapor-like" state or a low-density state. At some point, the fluid condenses, and a sharp step is observed on the isotherm. The fluid in the pore is then in a "liquid-like" state (see Fig. 1.2 and 1.5).

Because of the attractive pore walls, the condensation of the adsorbate will occur below the saturation pressure of the bulk fluid and is called 'capillary condensation.' The shifted pressure of condensation (p_c) is a hallmark of finite volume systems. This pressure p_c occurs higher than the vapor-liquid equilibrium pressure (p_e), but lower than the vapor phase limit of stability (vapor spinodal point, p_{SV}). In this region of $p_e < p < p_{SV}$, the vapor is considered in a metastable state. Once the fluid has condensed and filled the pore, we can begin to lower the external pressure. Before the fluid evaporates back to a vapor state (*desorption*) at some pressure p_d, we observe a liquid metastable state. For an illustration of these regimes, see Figures 1.2 and 1.5, and a further description in §1.2.4. Together, these phenomena (capillary condensation and delayed desorption) produce a reproducible hysteresis in phase transition pressures that is characteristic of porous materials [21, 22]. That is to say, these metastable states delay condensation and evaporation from the true vapor-liquid equilibrium (VLE). A metastable liquid is essentially overstretched, and is stabilized (on the time scale of experiments) by the strong attractive potential of the pore wall. The extent of hysteretic behavior, as well as other isotherm features, can be used to characterize the porous solid [23].

1.2.2 Porous materials

Porous solids encompass a huge cross section of materials. These range from natural materials such as coal, zeolites, and pumice stone to highly ordered synthetic materials derived from template chemistry. Any solid material with a large fraction of void space can be considered a porous solid. These materials have uses that are too numerous to count. This work will focus on two applications that are fundamental to both theory and practice: porous material characterization and chromatographic stationary phase.

A single sample of porous material can contain a wide distribution of pore sizes, as well as differing geometries. Structures can vary from amorphous crystalline materials where the pore volume is highly irregular voids between crystals, to symmetric, uniform pore size distributions possible through precursor chemical templates [24, 25]. The latter class of materials is of high interest. This is because materials with a very narrow pore size distribution come close to the theoretically ideal case of a single pore in simple geometry. Thus, with the discovery of such uniform structures, theoretical and computational methods can be tested and verified experimentally.

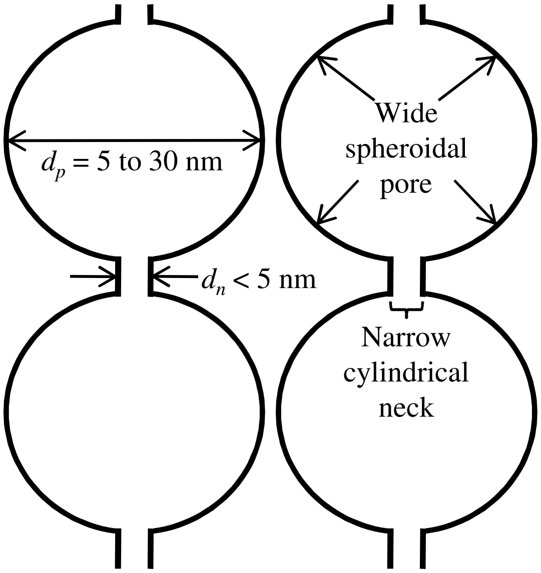


Figure 1.1: A schematic of a pore network with ink-bottle type pores where cavitation occurs.

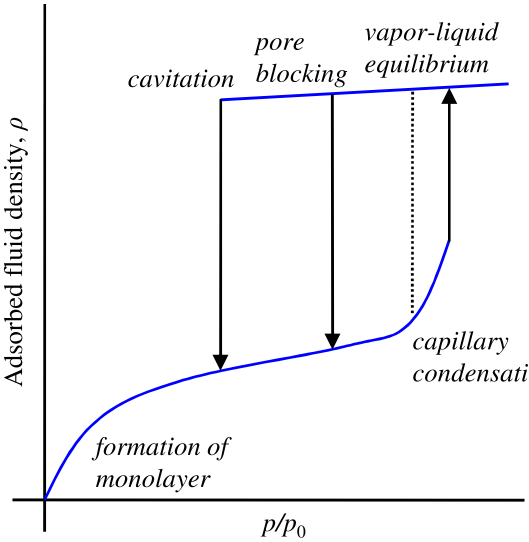


Figure 1.2: Sketch of an isotherm with approximate pressures of condensation, equilibrium and evaporation.

A tool central to the characterization of such materials is adsorption measurement. A small amount of a porous material is placed in an enclosed glass sample vial. The sample is then heated to de-gas it, and flushed with an inert gas such as helium, then placed in a heat bath. Each point on the isotherm is measured by introducing a known amount of adsorbate gas and measuring the pressure in the sample vial after an equilibration period. The resulting isotherms can reveal a great deal of information about the materials' porous structure. Early methods used to interpret these isotherms, such as BET surface area [26], were based on classical thermodynamic theory and idealized Langmuir isotherms. Recently, more accurate methods based on non-local density functional theory (NLDFT) have been introduced [23, 27–29] and have enjoyed widespread use, as well as a recent inclusion to the ISO standards for porous materials characterization [30]. This method involves creation of a kernel of isotherms of many pore sizes, for a given solid and geometry using NLDFT. The pore size distribution (PSD) can be back-calculated by comparing an experimental isotherm to the NLDFT kernel. Porous materials are classified by their nominal pore diameter; according to IUPAC standards [31], a *micropore* is defined where $d_p < 2$ nm, and *mesopore* is where 2 nm $< d_p <$ 50 nm, and a *macropore* is where $d_p >$ 50 nm.

The mechanism by which desorption occurs depends mainly on geometry of the pore. In long, cylindrical pores open to the bulk, desorption occurs at p_e via a receding meniscus. In materials with so-called "ink-bottle pores," the pore structure is usually a complicated network of small channels ("necks") connecting larger cavities to each other (Fig. 1.1). In such a system, desorption primarily occurs by two mechanisms: pore blocking percolation or cavitation. The evaporation of the condensate from a network of ink-bottle pores is hindered by the interconnected pore passages. In this case, desorption from the pore body may occur only after emptying of its neck. In other words, desorption from the neck triggers evaporation in the blocked pore. Thus, the pressure p_d of desorption from the pore body depends on the neck size and network connectivity. The onset of evaporation from the pore network is associated with the percolation threshold and the formation of a continuous cluster of pores open to the external surface [32–34]. The percolation mechanism is observed in pore networks with sufficiently large necks. Theoretical and experimental studies [35, 36] have revealed that if the neck diameter is smaller than a certain critical size (estimated to be ca. 5 nm for nitrogen at 77.4 K), the mechanism of desorption from the pore body involves cavitation, the spontaneous nucleation and growth of gas bubbles in the metastable fluid condensed in the pore, while the neck remains filled. In this case, the desorption transition pressure p_d does not depend on the size of pore necks and is determined by the properties of the adsorbate, and the size and shape of the pore body. Both desorption mechanisms, percolation and cavitation, originate from geometrical pore blocking; however, the physical mechanisms and the pore size dependence are quite distinct. The pore blocking effects cause a "delay" in desorption (Fig. 1.2), i.e., desorption occurs at vapor pressure below that of the equilibrium vapor-liquid transition in the given pore ($p_d < p_e$). Thus, in the course of desorption, one observes long-living metastable states of condensed fluid that is overstretched due to the action of capillary pressure.

1.2.3 Polymer physics

Polymer physics encompasses several fields and many decades of work. However, in light of this thesis, we are mainly interested in the statistical mechanical branch of polymer theory, and its application to the adsorption of polymers, i.e. in chromatography.

Freely-jointed chains

A freely-jointed ideal, or Gaussian, chain is the simplest model one can use when describing the statistical nature of chain molecules [37]. It consists of n monomer units connected with rigid bonds of the same length l. The polymer is assumed to behave as a random walk. No restrictions are placed on bond angles or torsion angles. Monomers are non-interacting, and can occupy the same volume. Although simple, this model is crucial to the field of statistical polymer physics, and several important results follow from the model in the thermodynamic limit. The most relevant to this work is relation of a chain's end-to-end distance and its radius of gyration to its chain length n,

$$\left\langle R_E^2 \right\rangle = nl^2 \tag{1.1}$$

$$\left\langle R_G^2 \right\rangle = \left\langle R_E^2 \right\rangle /6 \tag{1.2}$$

where end-to-end distance and radius of gyration are defined respectively as

$$R_E^2 = (\mathbf{r}_1 - \mathbf{r}_n)^2 \tag{1.3}$$

$$R_G^2 = \frac{1}{2n^2} \sum_{i,j} (\mathbf{r}_i - \mathbf{r}_j)^2 \tag{1.4}$$

The freely-jointed ideal chain was mainly derived as statistical model that characterizes polymers at certain length scales, not give specific information regarding any particular polymer species. A modification of the freely-joined chain is to assign potentials to monomers, such as hard core, square well or Lennard-Jones. This work will use the Lennard-Jones variant, since the goal is to produce realistic molecular models.

Theta transition

Long polymer chains exist freely in two states: the globule and the random coil [38]. A globular state occurs when intra-molecular (monomer-monomer) attraction is greater than the entropic penalty for a collapse to a denser state. Conversely, a random coil is observed when thermal motion overcomes the attraction between monomers. A random coil can be modeled as a self avoiding walk in the athermal limit [39]. A globular polymer is the chain molecule analogy to a simple fluid condensing into a droplet, while the random coil would be gaseous if the monomers were not tethered together. A state exists between these two extremes where attractive interactions are balanced by entropic repulsions; it is called the *theta (θ) point*. The state a chain assumes is determined by its temperature, and if in solution, by its solvent.

Theta transitions can be characterized by how a chain's radius of gyration (Eq. 1.4) scales with increasing chain length n [37]. In higher temperature systems, chains are well above the θ transition, which means that the chain accepts a self-avoiding random coil configuration manifested by the $R_G \propto (n-1)^{0.59}$ dependence between the radius of gyration R_G and the chain length n. In other words, we can say that the polymer is dissolved in a good solvent. For chains below the θ point, which means that polymer-polymer interactions are more favorable than polymer-solvent ones; this effective attraction between the monomers prevails over the entropic penalty for collapse, and the polymer comprises into a globule with $R_G \propto (n-1)^{1/3}$. At the θ temperature, the entropic contribution (favoring an expansion into a self-avoiding coil) and enthalpic contribution (favoring a contraction into a globule) approximately cancel each other, and the polymer effectively behaves like a Gaussian chain (one with no intermolecular potential), $R_G \propto (n-1)^{1/2}$. The θ temperature for a freely-jointed LJ polymer is estimated as $T^* = 3.18$ when considering short range interactions ($r_c < 2.5\sigma$) [39].

Adsorption and chromatography

Polymer adsorption is observed in a wide range of fields, for example in hydrocarbon separations, biological systems (protein and DNA translocation), colloidal stabilization and surfactants, just to name a few. However, a main interest of this thesis is separations via chromatography. Chromatography is a technique based on the differing interactions of certain components in a mixture with some stationary phase. A very general description of a separation using chromatography follows: the target mixture (called the solute) is added to a solvent (the *eluent*). The eluent is then injected into a chromatographic column containing the *stationary phase.* This is can be some adsorbent material such as silica particles or highly cross-linked polymers, and is selected based on operating parameters and known chemistry of the solute and solvent. A pressure gradient is then applied to the column. Each component of the original mixture is *eluted* (expelled from the column) at a different rate depending on its interaction with the stationary phase. The time each component spent in the column is called the *retention time.* Chromatography is an important tool in biochemical and analytical chemistry, as well as in the petrochemical industry a unit process.

Many different varieties of chromatography are in use, depending on experimental setup and substance under analysis. Separation of polymers by molecular weight is performed using size-exclusion chromatography (SEC). This method does not depend on adsorption interactions between the mobile and stationary phase, but only the entropic replusion of confining a polymer in pores (found in the stationary phase). If the desired separation mode is not molecular weight, one must utilize adsorptive chromatagraphy. In this method, the solute does interact with the stationary phase, and this interaction can be tailored using several parameters such as type and concentration of solvent [14, 15] (see Fig. 1.3).

The transition from SEC (dominate repulsive forces) to adsorptive chromaography (dominate adsorption forces) is called the critical point of adsorption (CPA) [12]. CPA is defined as the point where steric, entropic forces are balanced by attractive, enthalpic attractive forces. With liquid chromatography at the critical conditions (LCCC), retention times do not depend on the molecular weight of the solutes. Thus, the process can be tailored to

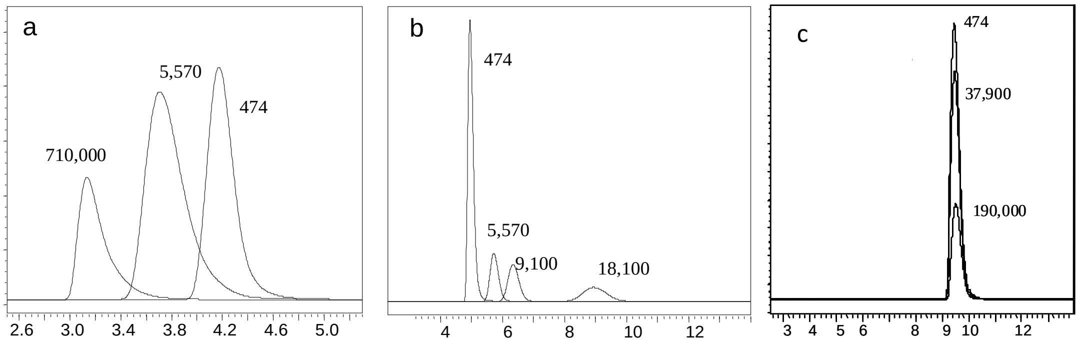


Figure 1.3: Effects of solvent on polymer chromatography. Changing solvent in the mobile phase can result in (a) the largest MW to be eluted first, (b) the smallest MW to be eluted first, or (c) no dependance on MW. Data from Ref. [13]

.

separate components based on chemical composition and other structural properties. A major obsticle for widespread implementation of CPA based chromatography is the inablity to predict under which conditions it will occur.

1.2.4 Computational methods of study

A nanopore (of say $d_p \approx 10$ nm) filled with an adsorbing fluid such as nitrogen or argon would contain up to 10,000 molecules. The tools relevant to the scale of such a problem would be molecular dynamics (MD) or Monte Carlo (MC) simulations. These methods employ an either an atomistic or united-atom model. For adsorption studies, MD is often of little use. This is because MD can only sample a finite time frame, and one that is usually very short (on the order of picoseconds). To reach an equilibrium point on an adsorption isotherm, long experimental times are required (at least several minutes). Thus, we choose MC to simulate the adsorbate/pore system. In general, an adsorbate is modeled as a Lennard-Jones (LJ) sphere, and the pore body is modeled as a single pore with a simple geometry (sphere, cylinder, etc.) that interacts with the adsorbate molecules via a mean-field type, integrated LJ potential [27, 40].

Because our goal is a detailed understanding of equilibrium phase of confined fluid, knowledge of the system's free energy is crucial. Free energy and chemical potential are considered *thermal* quantities, rather than *mechanical* quantities [41]. In general, mechanical properties such as pressure and internal energy are relatively simple to calculate in a computer program. Thermal quantities such as entropy, free energy and chemical potential, are much more difficult as they are usually functions of the partition function and thus of the system's phase space. Although they cannot be directly calculated, many innovative methods are available for the practical calculation of free energy or chemical potential. Several basic ones are described below.

Thermodynamic integration

Perhaps the most straight-forward approach to the calculation of thermal quantities is thermodynamic integration. To calculate free energy by thermodynamic integration, one must construct a reversible path from a state with known free energy (the reference state), and integrate along it. It is useful to see the derivation. In an adsorption system, the fluid confined in the pore is (i) in equilibrium with a fixed bulk fluid phase (constant μ), (ii) held at a constant temperature as an isotherm is measured (constant T) (iii) in a pore where the walls are assumed to be rigid (constant V). This is considered a grand canonical ensemble (see Fig. 1.4, top, for a schematic), and the applicable thermal quantity (the value that will be minimized to obtain an equilibrium state) is the grand potential Ω. Written as a Fundamental Equation,

$$\Omega = U - TS - \mu N = F - \mu N \tag{1.5}$$

This can be written in the differential form as

$$d\Omega = -SdT - pdV - Nd\mu \tag{1.6}$$

Applying the conditions mentioned above (constant V, T) results in the Gibbs adsorption equation:

$$N = -\left(\frac{\partial \Omega}{\partial \mu}\right)_{V,T} \tag{1.7}$$

Thus, if an isotherm $N(\mu)$ has a continuous path from some known reference state r to some other point of interest a, we can easily calculate the grand potential along this path,

$$\Omega(\mu_a) - \Omega(\mu_r) = -\int_{\mu_r}^{\mu_a} N(\mu) d\mu \tag{1.8}$$

A similar approach is used to determine the phase coexistence. Thermodynamic integration along the continuous isotherm $N(\mu)$, and the vapor-liquid equilibrium (line $B-F$ in Figure 1.5) is determined Maxwell's rule of equal areas,

$$\oint_{\mu_e} N(\mu) = 0 \tag{1.9}$$

In other words, if line $B{-}F$ is the equilibrium point, areas $D{-}SL{-}F{-}D$ and $B{-}SV{-}D{-}B$ of Fig. 1.5 must be equal. Maxwell's rule has been proven to be legitimate for bulk phase equilibrium [42].

Chemical potential in Monte Carlo

Above, it is shown that the ability to calculate a continuous pathway from a reference to a target state allows for the determination of the free energy at that point. For the case we're interested in, it is most useful to calculate the Gibbs isotherm (Eq. 1.7). One of the first and simplest methods for meansuring chemical potential in MC simulations was introduced by Widom in 1963 [8]. It involves inserting a *trial* or *test* particle that interacts with the system, but not allowing that test particle to influence the system. The chemical potential potential is then related to the average potential energy interaction 'felt' by the test particles. As this method is basis for the gauge cell method, a short derivation follows:

To derive the expression for chemical potential for a simple fluid MC system, we begin with the canonical (constant NVT) partition function,

$$Q(N, V, T) = \frac{1}{\Lambda^{3N} N!} \int_{(V)} d\mathbf{r}^N \exp\left[-\beta\Phi(\mathbf{r}^N)\right], \tag{1.10}$$

where N is the number of indistinguishable molecules, V and T are the system's volume and temperature respectfully, $\beta = 1/k_\mathrm{B}T$ is the inverse temperature, Λ is the thermal de Broglie wavelength, Φ is the sum of the various potential energy contributions (such as intermolecular [LJ] and external [attractive pore walls]), and $\mathbf{r}$ is the set of position vectors for the system in phase space. The so called "bridge" equation links the microscopic partition function to the macroscopic quantity Helmholtz free energy (the thermal quantity relevant to this ensemble),

$$F(N,V,T) = -k_{\mathrm{B}}T \ln Q(N,V,T). \tag{1.11}$$

The applicable Fundamental Equation in differential form for this ensemble is

$$dF = -SdT - pdV + \mu dN \tag{1.12}$$

Separating chemical potential from the Fundamental Equation gives us

$$\mu(N) = \left(\frac{\partial F}{\partial N}\right)_{V,T} \tag{1.13}$$

at constant V, T and in the thermodynamic limit $N \to \infty$. For large N, this value can be approximated by a finite difference:

$$\mu(N) = F(N+1,V,T) - F(N,V,T). \tag{1.14}$$

Note that either a forward or reverse difference is technically correct, however this choice has serious practical consequences. Substituting Eq. 1.11 into Eq. 1.14 gives

$$\mu(N) = -k_{\mathrm{B}}T \ln[Q(N+1,V,T)/Q(N,V,T)] \tag{1.15}$$

Substituting the partition function (Eq. 1.10) into Eq. 1.15 and rearranging gives us

$$\begin{aligned} \mu &= -k_{\mathrm{B}}T \ln\left(\frac{V/\Lambda^3}{N+1}\right) - k_{\mathrm{B}}T \ln\left[\frac{1}{V}\frac{\int d\mathbf{r}^{N+1} \exp\left(-\beta\Phi(\mathbf{r}^{N+1})\right)}{\int d\mathbf{r}^{N} \exp\left(-\beta\Phi(\mathbf{r}^{N})\right)}\right] && (1.16) \\ &= \mu_{id} + \mu_{ex} && (1.17) \end{aligned}$$

where the first part of Eq. 1.17 is the ideal chemical potential and the second is the excess chemical potential, which is calculated by Widom's method. If we define $\Delta\phi = \Phi(\mathbf{r}^{N+1}) -$

$\Phi(\mathbf{r}^N)$, or the interaction energy of the $N+1$ particle, we can finally rewrite Eq. 1.16 as

$$\begin{aligned} \mu_{ex} &= -k_{\mathrm{B}}T \ln \left[\frac{1}{V} \int_V d\mathbf{r}_{N+1} \left\langle \exp(-\beta\Delta\phi) \right\rangle_N \right] && (1.18) \\ &= -k_{\mathrm{B}}T \ln \left\langle \exp(-\beta\Delta\phi) \right\rangle_N && (1.19) \end{aligned}$$

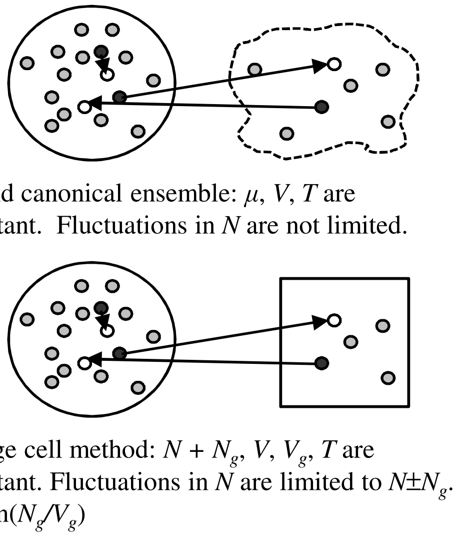
where $\langle\cdots\rangle$ represents a canonical average, and the subscript $_N$ signifies constant N (in that the test $N+1$ particle never influences the system). The integral in Eq. 1.18 means we must compute the average over all particle positions for $N+1$ in volume V. Equation 1.19 is easily calculated in the course of an MC simulation. The main disadvantage of Widom's particle insertion method is its failure at high molecular densities and with inhomogeneous systems. At high densities, trial insertions are much more likely to overlap with another molecule, thus contributing zero to the running average (as $\Delta\phi$ is increasingly large, $\exp(-\beta\Delta\phi)$ goes to zero).

Many methods that expanded on Widom's have been proposed, such as particle deletions (the reverse difference of Eq. 1.14), multi-staged particle deletions [43], the bicanonical ensemble [44], the overlapping distribution method [45], the multiple histogram method [46], the acceptance ratio method [45], umbrella sampling [47], the method of expanded ensembles [48], and others. Although Widom's method is technically correct for dense or inhomogeneous systems, it is practically unfeasible, as simulation times grow too large and still result in large statistical uncertainties.

The gauge cell method

The mesocanonical ensemble, also known as the gauge cell method [9, 10, 49], models equilibrium between the fluid in a pore sample system and a finite size reservoir of a limited capacity; the so-called gauge cell. The two cells of constant volume are considered in thermal and chemical equilibrium, maintained using particle exchange between them, somewhat similar to grand canonical MC, where the system is allowed to exchange particles with an infinite bulk at a constant chemical potential (Fig. 1.4). In the gauge cell, the total number of particles, $N_\Sigma = N_{pore} + N_{gauge}$, is conserved. The gauge cell MC setup is referred to as the mesocanonical ensemble; it approaches the canonical ensemble when the gauge cell is infinitely small ($V_{gauge} = 0$) and the grand ensemble at the infinitely large gauge cell ($V_{gauge} \to \infty$). The gauge cell fulfills two purposes: its limited size restricts density fluctuations in the system, and it serves as a reference for finding the chemical potential of the pore fluid (since the condition of equilibrium between the pore and the gauge is the equality of chemical potentials). The suppression of fluctuations allows for simulations of metastable and labile states, resulting in a continuous van der Waals's type isotherm that can be thermodynamically integrated. The backwards trajectory of unstable states corresponds to unstable bubbles, which would spontaneously evaporate in an open system.

The gauge cell found numerous applications in simulation studies of phase transitions in strongly heterogeneous systems. It has been employed in studies of capillary condensation in various geometries [50–52], liquid bridge [53] and droplet nucleation [54], bubble cavitation in metastable liquid [55], adsorption deformation [56] and the adsorption of surfactants [57]. The gauge cell has been extended from its original formulation [9] to extremely small confinements with the ideal gas gauge cell (IGGC) method [10], multi-component gauge cell method for mixtures [49], coupling with Widom insertions [58], configurational bias [59, 60], and more [61].



Grand canonical ensemble: μ, V, T are constant. Fluctuations in N are not limited.

Gauge cell method: $N + N_g$, V, V_g, T are constant. Fluctuations in N are limited to $N \pm N_g$. $\mu \sim \ln(N_g/V_g)$

Figure 1.4: Diagram comparing grand canonical and gauge cell Monte Carlo.

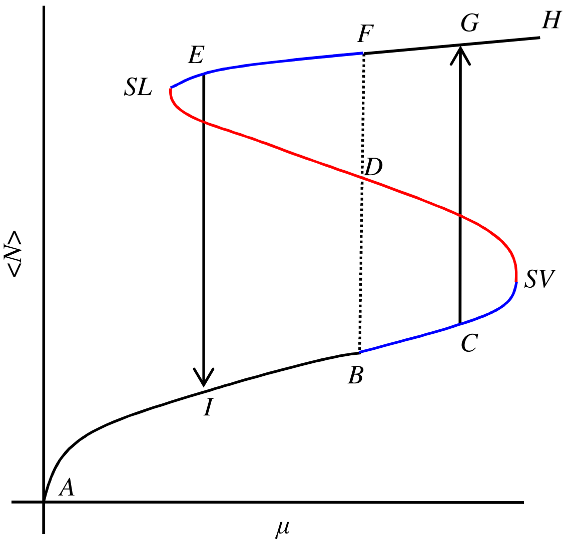


Figure 1.5: Sketch of a canonical isotherm, which can be calculated using the gauge cell method.

The main advantage of the gauge cell is relatively efficient calculation of the canonical isotherm. Recall from §1.2.4 with the isotherm $N(\mu)$ one can calculate the free energy difference between two states with relative ease. Figure 1.5 shows a sketch of a canonical isotherm that summarizes the preceding sections. At low μ, the pore is empty (point A). Increasing μ (by increasing the external bulk fluid's pressure) forms the monolayer of adsorbed fluid ($A - I$). Adsorbate density continues increasing with μ. When $\mu_e < \mu < \mu_{SV}$, the fluid is in a metastable vapor-like phase (line $B - SV$). In other words, in the region of chemical potential greater than the vapor-liquid equilibrium, but before capillary condensation, the adsorbate is in a metastable state. Capillary condensation occurs at μ_c, or line $C - G$, in grand canonical ensemble MC (GCMC). By using the gauge cell, the metastable branch can be extended to the limit of stability, the spinodal, where the condensation barrier is zero. From the vapor spinodal, $N(\mu)$ follows a backward trajectory of unstable, or labile, states (line $SV - SL$). Any of these states would immediately transition to the vapor or liquid branch in an open system. From the liquid spidonal SL to the equilibrium F is a liquid metastable state, followed by the stable liquid state ($F - H$). The spontaneous desorption μ_d observed in GCMC is represented by line $E - I$.

Chapter 2

Methods of Calculating Chemical Potential of Polymers

2.1 Introduction

Atomic-scale simulation of macromolecules has been a challenge spanning decades of work. A particular difficulty is the calculation of a chain molecule's chemical potential, and thus the system's free energy. The free energy is a crucial thermodynamic quantity that determines the system's stability. Determining the free energy is necessary for many current problems involving structured polymer systems. Examples of problems that require determination of polymeric chemical potential are numerous and wide-reaching. The prediction of vapor-liquid equilibrium [62, 63] and thus development of an equations of state is a classic example. Separation and purification of polymer is usually achieved by chromatography, the theory of which is based on the free energy difference of dissolved and adsorbed polymers [64]. Knowledge of the free energy is crucial when predicting the morphology of nanostructured systems such as micelles, polymer brushes, and systems where self assembly is key. Translocation, or the process by which a polymer moves through a small opening, is implicated in many important biological phenomena such as DNA transport, viral injection mechanisms, trans-membrane protein transport, and DNA sequencing. Theoretical studies require the free energy landscape to model translocation [65]. The field of protein folding and denaturing depends on highly accurate calculations to probe large free energy landscape and find likely structures [66]. Advanced drugs and drug delivery and implicated as well; biological drugs are proteinous in nature, and their stability in processing, storage, and *in vivo* is crucial to their efficacy and marketability. Clearly, simulation of polymers and calculation of their free energy encompasses many of today's challenging problems.

One inherent difficulty encountered when modeling a polymeric system is large range of relaxation times. For a typical polymer chain in noncritical conditions, pertinent relaxation times range from 10^{-15} s for bond vibrations to 10^{-4} s for conformation change [67]. For this reason, Metropolis algorithm [68] Monte Carlo (MC) simulations are well-suited to study polymers. The stochastic nature of MC allows convergence to equilibrium state without visiting every relaxation state along the way. Careful implementation allows for study of dynamics as well [67].

In many cases of practical problems, a relatively detailed continuum model is required is capture relevant properties. As complexity increases, obtaining the chemical potential and free energy from simulations becomes more challenging. The chemical potential of a thermodynamic system is a thermal quantity, and cannot be measured directly (unlike mechanical properties, such as pressure, volume, and particle count). Obtaining chemical potential from simulation thus requires a diligent approach on part of the researcher. Due to the large number of conformations a single molecule can attain, standard MC tools such as Widom trial insertions are insufficient due to poor sampling. The need for sophisticated equilibration and relative flexible that MC permits has led to a wealth of clever algorithms in the literature over the last 30 years.

The goal of this work is to provide the reader with all available techniques for the calculation (or equilibration) of chemical potential of polymer molecules in off-lattice MC simulations. An emphasis is placed on standard models, i.e. pseudo-atomistic force fields such as OPLS [62] or TraPPE [63], and for polymers in the melt or dilute solution limit. The focus will be on standard ensembles: canonical (constant $N,\ V,\ T$), grand canonical (constant $\mu,\ V,\ T$), and the Gibbs ensemble [69] for vapor-liquid equilibrium. This paper is not a comprehensive history of polymer simulation, which is far too rich and varied for a single paper, nor is it a guide for the successful simulation and equilibration of polymers. For that, the reader is referred to Refs [41, 67, 70, 71] as a practical starting point.

The rest of this chapter is organized as follows: Section 2.2 presents a short review of calculating the chemical potential of simple fluids in MC simulations. This serves as a basis for most of the methods that follow. Section 2.3 details all of the different approaches to calculating the chemical potential of polymers, organized by the type of method. A summary and outlook is presented in Section 2.4.

2.2 Chemical Potential in Computer Simulation

2.2.1 Widom's Insertion Method

he chemical potential of a given species is defined as the change of free energy with respect to the number of particles of that species. It can be obtained from the fundamental equation of thermodynamics, displayed here in terms of the Helmholtz free energy F,

$$dF = -SdT - pdV + \sum_i \mu_i dN_i \tag{2.1}$$

For the canonical variables N, V, T, the representative free energy quantity is the Helmholtz free energy. Chemical potential is a thermodynamic potential in that a system tends to minimize its chemical potential. Fixing the canonical ensemble variables $N_{j\neq i}$, V, T, an expression for the chemical potential of a given species i is obtained, as the partial derivative of F with respect the number of particles of species i,

$$\mu_i = \left.\frac{\partial F}{\partial N_i}\right|_{T,V,N_{j\neq i}} \tag{2.2}$$

Equation (2.2) is the basis for most computational algorithms designed to calculate μ.

The first and most common approach to calculating chemical potential in MC is Widom's method [8]. In this approach, Equation (2.2) is considered a finite difference in the thermodynamic limit,

$$\mu(N) = F(N+1) - F(N), \tag{2.3}$$

where the subscript i is dropped to focus on single species systems. Note the use of a forward or backward difference makes practical difference, and will be illustrated below. The classical partition function for a canonical system (fixed N, V, T) is given by [41],

$$Q(N,V,T) = \frac{1}{\Lambda^{3N}N!}\int_{(V^N)} dr^N \exp\left[-\Phi\left(r^N\right)/k_\mathrm{B}T\right], \tag{2.4}$$

where $\mathbf{r}^N$ is the position vector of all N particles and $\Phi(\mathbf{r}^N)$ is their potential energy. From statistical thermodynamics, we know that the Helmholtz free energy is related to the partition function,

$$F(N,V,T) = -k_\mathrm{B}T\ln Q(N,V,T) \tag{2.5}$$

Substituting Equation (2.4) into (2.5) gives the free energy in terms of the Boltzmann function,

$$F(N,V,T) {=} -k_{\mathrm{B}}T\ln\left(\frac{V^N}{\Lambda^{3N}N!}\right) - k_{\mathrm{B}}T\ln\left(\frac{1}{V^N}\int dr^N \exp\left[-\Phi\left(r^N\right)/k_{\mathrm{B}}T\right]\right) \tag{2.6a}$$

$$= F_{\mathrm{id}}(N,V,T) + F_{\mathrm{ex}}(N,V,T). \tag{2.6b}$$

The free energy is represented as a combination of two terms, the ideal and excess free energy (2.6b). The ideal free energy serves as a reference state is the free energy of an ideal gas at N, V, T. The excess free energy can be calculated from MC averaging. Combining Equations (2.3) and (2.5), we obtain an expression for chemical potential in terms of the partition function,

$$\mu(N) = -k_{\mathrm{B}}T\ln\left[\frac{Q(N+1,V,T)}{Q(N,V,T)}\right] \tag{2.7}$$

It follows then that,

$$\frac{\mu(N)}{k_{\mathrm{B}}T} {=} -\ln\left[\frac{V}{\Lambda^3(N+1)}\right] - \ln\left[\frac{\int dr^{N+1}\exp\left[-\Phi\left(r^{N+1}\right)/k_{\mathrm{B}}T\right]}{\int dr^N \exp\left[-\Phi\left(r^N\right)/k_{\mathrm{B}}T\right]}\right] \tag{2.8a}$$

$$= -\ln\left[\frac{V}{\Lambda^3(N+1)}\right] - \ln\left[\frac{\int dr_{N+1}\exp\left[-\phi\left(r_{N+1}\right)/k_{\mathrm{B}}T\right] dr^N \exp\left[-\Phi\left(r^N\right)/k_{\mathrm{B}}T\right]}{\int dr^N \exp\left[-\Phi\left(r^N\right)/k_{\mathrm{B}}T\right]}\right] \tag{2.8b}$$

$$= \frac{1}{k_{\mathrm{B}}T}\left(\mu_{\mathrm{id}}(N) + \mu_{\mathrm{ex}}(N)\right) \tag{2.8c}$$

where the subscript $N{+}1$ refers to a single particle, and ϕ is its interaction energy with the remaining N particles. Realizing the second term in Equation (2.8b) is the definition of a canonical average, we obtain an expression applicable to a computer simulation,

$$\mu_{\mathrm{ex}}(N) = -k_{\mathrm{B}}T\ln\left\langle \exp\left[-\phi(r_{N+1})/k_{\mathrm{B}}T\right]\right\rangle_N \tag{2.9}$$

where the angle brackets refer to a canonical average, and the subscript N denotes the average is taken over N particles. In other words, the Boltzmann factor of the $N{+}1$ particle is averaged over the ensemble of N particles, at constant volume and temperature. Crucially, the system of N particles does not interact with the $N{+}1$ particle, hence its designation as the “trial,” “probe,” or “ghost” particle. In practice, one simply inserts a molecule at a uniformly random position in the system during the simulation run. The Boltzmann factor of these insertions is then averaged over the insertion volume. Equation (2.9) is the most common and straightforward approach to calculating the chemical potential of simple molecular fluids. However, it suffers significantly at high density, especially in crowded and confined systems. As the likelihood of overlap between the monomers or with the confining boundaries increases, the chance of a statistically significant insertion grows small. Thus for dense fluids, prohibitively long simulations are required to obtain reasonable results.

2.2.2 Extension to Polymers

Extension of Equation (2.9) to a polymer chain is straightforward. For ease, we will focus of a fluid of monodisperse fluid of N polymers of length n (although it is relatively simple to consider a polydisperse fluid instead [1]). The partition function for such a case is

$$Q(N, n, V, T) = \frac{1}{\Lambda^{3Nn} N!} \int_{(V^{Nn})} dr^{Nn} \exp\left[-\Phi\left(r^{Nn}\right)/k_{\mathrm{B}}T\right] \tag{2.10}$$

Note that no restrictions for connectivity are made. Again using a finite difference for Equation (2.2), the chemical polymer is the increase in free energy from the addition of another n-mer,

$$\mu(N) = F(N+1, n, V, T) - F(N, V, T) \tag{2.11}$$

Following the same approach as Widom's method above, we can obtain the equivalent expression for a fluid of chains,

$$\frac{\mu(N)}{k_{\mathrm{B}}T} = -\ln\left[\frac{Q(N+1, n, V, T)}{Q(N, n, V, T)}\right] \tag{2.12a}$$

$$= -\ln\left[\frac{V}{\Lambda^{3n}(N+1)}\right] - \ln\left(\left\langle \exp\left[-\phi\left(r_{N+1}\right)/k_{\mathrm{B}}T\right]\right\rangle_{Nn}\right) \tag{2.12b}$$

$$= \frac{1}{k_{\mathrm{B}}T}\left(\mu_{\mathrm{id}}(N) + \mu_{\mathrm{ex}}(N)\right). \tag{2.12c}$$

The primary differences between Equations (2.9) and (2.12) is the reference state is a gas of ideal n-mers, and $\mathbf{r}_{N+1}$ is a vector of n coordinates for the inserted test chain.

However, for all but the shortest chains (length $n \leq 5$), a random insertion would most likely encounter significant overlap with itself or neighboring fluid particles of confining boundaries. This results in a Boltzmann factor of zero, and no contribution to the average. A similar effect is found in grand canonical simulations; successful insertions of chains are extremely rare, and thus $\langle N \rangle$ is poorly sampled. The likelihood of overlap increases exponentially with chain length n, and with the density of fluid. Thus, a simulation, even with very long length, has extremely poor statistics. The main goal of the works mentioned below is to alleviate this problem. The great flexibility of MC techniques has allowed for many ingenious approaches.

A typical molecular model for a polymer consists of internal and external potentials. Internal, or intramolecular, potentials impart connectivity and rigidity to a chain. Atoms and/or pseudoatoms are connected with rod or harmonic bonds. Rigidity can be imbued by harmonic angle bonds, 1—3 harmonic bonds, and torsion potentials (typically a cosine series). A chain's external, or intermolecular, potential usually refers to all non-bonded contributions to its potential energy. This can include monomer-monomer interactions such as the Lennard-Jones (LJ), electrostatic, or adsorption potentials. LJ or other repulsive potentials account for excluded volume of real chains. The internal potential energy is another important distinction from simple fluids, whose internal energy is zero or fixed for most MC simulations. This also complicates the reference state of a chain fluid. It is typical to use a fluid of non-interacting chains as a reference state; that is, a chain with its internal interactions only. Without monomer-monomer interactions, overlap of particles is allowed and the chain has no excluded volume. In the case of a flexible chain (no rigidity) with stiff, fixed length bonds, the reference state is a fluid of *ideal*, or *Gaussian* chains [37]. If the bonds were modeled instead as a harmonic well, the reference state would be a fluid of *Rouse* chains [38]. For clarity, variable references are standardized to the above notation whenever possible: N is the number of chains in a system, n is their length in effective monomers, and Φ their potential energy.

2.3 Methods of Calculating Chemical Potential

We will now present a collection of approaches available in the literature that circumvent the poor insertion probabilities mentioned above. The sections below are roughly ordered by time of introduction. Note that their effectiveness and usefulness is highly dependent on the specific application.

2.3.1 Simple Insertions

For the reasons mentioned above, simple Widom insertions (2.9) are not feasible for polymers and even most oligomers. Kumar et al. [11] used simple insertions for chains $n \leq 10$ inserted into free space, at low (globular) and high (random-coil) temperatures. They found poor statistics for the low temperature case at $n \geq 5$, and at $n \geq 8$ for the high temperature case. This is displayed in Figure 2.2 as filled triangles and circles for low and high temperatures, respectfully. They found at lengths longer than 10 monomers, the simple insertions were too inefficient to use at all. Similarly, Frenkel et al. [72] found that simple insertions failed completely for $n > 3$.

Although some earlier studies made use of simple insertions [73–75], they focused mainly on lower dimensions, and hard-core models.

2.3.2 Thermodynamic integration

The thermodynamic integration approach is perhaps the simplest idea to overcome insertion overlap. It is based on the idea that the chemical potential is equivalent to the reversible work of inserting a molecule into a fluid. There is no requirement that this insertion occur in one physical step, only a continuous, reversible path. If a molecule can be inserted gradually, the total work of insertion would be the integral over the path taken. A simple method was proposed by Mon and Griffiths [76]. Although their implementation was on a 2D lattice, their approach is easily transferable to continuum studies. A set of “weakened” increments is introduced,

$$\Psi_\lambda = \sum_{i=1}^{N} \Phi_\lambda \left(|\mathbf{r}_{N+1} - \mathbf{r}_i| \right) \tag{2.13}$$

where Φ_λ is the weakened interaction energy of an $N+1$ molecule inserted into a system at constant N, V, T, and $\mathbf{r}_i$ is the position of particle i in the system. It varies as $\Phi_{\lambda=0} = 0$ to $\Phi_{\lambda=1} = \Phi$. The average of an observable A in an ensemble with the $N+1$ particle interacting through a weakened potential is

$$\langle A \rangle = \frac{\int d\mathbf{r}_1 \cdots d\mathbf{r}_{N+1} A \exp\left[-\beta(U_N + \Psi_\lambda)\right]}{\int d\mathbf{r}_1 \cdots d\mathbf{r}_{N+1} \exp\left[-\beta(U_N + \Psi_\lambda)\right]} \tag{2.14}$$

Note that $\lambda = 0$ and $\lambda = 1$ refers to ensembles of N and $N+1$, respectively. If we let λ vary continuously, it can be shown that the chemical potential is

$$\mu = \int_0^1 \left\langle \frac{d\Psi_\lambda}{d\lambda} \right\rangle_{N+\lambda} d\lambda \tag{2.15}$$

There is flexibility in how λ is implemented. Initially, the authors [76] used λ as a proportionality coefficient, such that

$$\Phi(r) = \epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] \tag{2.16}$$

and

$$\Phi_\lambda = \lambda\Phi \tag{2.17}$$

This worked poorly, as overlap causes Equation (2.16) increases rapidly, even at very small values of λ. To alleviate this, they implemented a step-wise potential, shown in Figure 2.1. As λ increases, the repulsive part of the LJ potential is increased in discreet ‘jumps.’

Using Equation (2.15), Mon and Griffiths [76] calculated the chemical potentials of LJ 15- and 35-mers on a 2D lattice. Their approach is a straight-forward and robust method for calculating chemical potential of polymers.

A similar approach was followed Müller and Paul [77] chose the excluded volume interaction as the path variable. This is achieved by gradual inclusion of the intermolecular potential. In this way, a full chain can be inserted (initially), experiencing no resistance from the fluid molecules. In their work [77], the excess chemical potential was calculated for bond fluctuation model [78], which is a type of lattice model. The inserted chain is called a

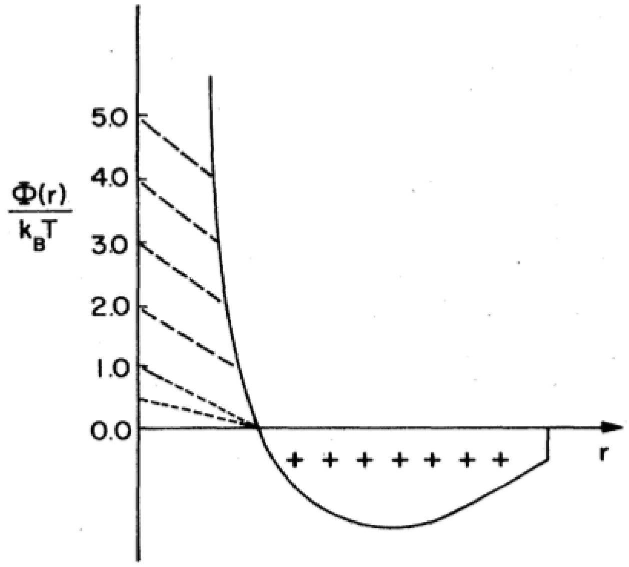


Figure 2.1: Diagram of progressively weakened potentials for insertion. The solid line is the LJ pair potential, and the dashed lines are successively weaker potentials Φ_λ. From Ref. [76].

"ghost-polymer," and obeys the internal bond restrictions, as well as hard-sphere repulsion of its constituent monomers. Note that latter is not necessary, but aids in efficiency of the algorithm; the ghost-polymer starts as self-avoiding before the excluded volume interaction is increased in subsequent simulations. The authors used this method to calculate μ_{ex} of bond-fluctuation polymer melts with chains of length n = 20, 50, 80 on a 40^3 lattice, at densities of 0.1 to 0.5. The ghost-polymer method was later extended by Wilding and Müller [79] to the method of expanded ensembles. This allows calculation of μ_{ex} in a single simulation.

The method of thermodynamic integration over 'partially' inserted chains provides an accurate, chain length-independent approach to calculating chemical potential. It is also quite flexible; varying excluded volume interaction or intermolecular potential are only two choices for the reversible path. The obvious disadvantage is the need for many simulations to obtain one state point. This is confounded by the non-physicality of the intermediate states; they provide no useful information in and of themselves.

2.3.3 Incremental Insertion

Rather than insert an entire non-interacting chain, the approach of Kumar et al. [11] obtained chemical potential by incrementing an existing chain with a trial particle and measuring its interaction with the rest of the system. A similar method was described by Mon and Griffiths [76], where increments where included gradually by varying their interactional potential, but only applied to 2D fluid. Kumar el al. called their approach the "modified Widom method." In a sense, it can be thought of as thermodynamic integration over the chain length coordinate. By insertion of only a single monomer, the statistics become as reliable as traditional Widom insertions. Of course, this scheme does not calculate the chemical potential, but the *incremental chemical potential*, that is, the difference in chemical potential between an i-mer and an $(i{+}1)$-mer in a solution of N chains of length n,

$$\mu_{\text{inc}}(n+1) \equiv \mu(\{N-1,n\},\{1,n+1\}) - \mu(N,n) \tag{2.18a}$$

$$= F(\{N-1,n\},\{1,n+1\}) - F(N,n) \tag{2.18b}$$

Equation (2.18) describes the difference between a system of N n-mers and an identical system where one n-mer is incremented by one. Note the chemical potentials in Equation (2.18a) are calculated with a backwards finite difference. With the definition presented in Equation (2.18b), parallels with the Widom test are obvious. The modified Widom method then calculates the incremental chemical potential by

$$\mu_{\text{inc}} = -k_{\text{B}}T\ln\left(\frac{V_{ins}}{\Lambda^3}\right) - k_{\text{B}}T\ln\left(\langle\exp\left[-\phi\left(r_{ins}\right)/k_{\text{B}}T\right]\rangle_{Nn}\right) \tag{2.19}$$

where $\mathbf{r}_{ins}$ are the coordinates a monomer inserted into a volume V_{ins} as the end of an n-mer. The chemical potential of the n-mer is then sum of its incremental values. The incremental chemical potential is well-defined thermodynamic quantity, with no restricting assumptions used in its derivation [18]. This can be illustrated by expanding terms in the definition of the chemical potential of a chain (Equation (2.11)),

$$\exp\left(-\frac{\mu}{k_{\text{B}}T}\right) = \frac{Q(N+1,n)}{Q(\{N,n\},\{1,n-1\})}\frac{Q(\{N,n\},\{1,n-1\})}{Q(\{N,n\},\{1,n-2\})}\cdots\frac{Q(\{N,n\},\{1,1\})}{Q(N,n)} \tag{2.20}$$

Applying the definition of the incremental chemical potential (2.18b), and (2.5),

$$\mu = \mu_{\text{inc}}(n) + \mu_{\text{inc}}(n-1) + \ldots + \mu_0 \tag{2.21}$$

Here, μ_0 refers to the chemical potential of a monomer (i.e. the first bead of the chain) which is inserted into the system.

The advantage of the modified Widom method is that it returns the effectiveness of sampling statistics back to that of Widom insertions into a simple fluid. There is no length restriction on incremental chemical potential, only the same density limitations that Widom's method suffers. A significant disadvantage is the fact that one must construct n simulations for the rigorous determinations of the chemical potential of an n-mer. This is somewhat alleviated by the fact that the shorter chain systems are physically relevant systems; chain length dependence of the incremental potential can be predicted in certain situations [80], and smart interpolation lessens the computational burden.

Kumar et al. used the modified Widom method to calculate the chemical potential and pressure for melts of 20-mers at varying density and temperature [11]. In addition, incremental chemical potential was calculated as a function of chain length (Figure 2.2). The authors asserted that the incremental chemical potential did not depend on chain length for long enough chains. They proposed that the chemical potential of a polymer could be calculated in a single simulation, with several small "short-chain" corrections for increased accuracy,

$$\mu = \sum_{i}^{n_{\text{short}}} \mu_{\text{inc}}^{\text{short}}(n) + (n - n_{\text{short}})\mu_{\text{inc}} \tag{2.22}$$

In this way, the required number of simulations is greatly reduced. This assumption of chain length independence became known as the "chain increment ansatz" [81] and was sharply contested [82, 83]. Kumar later clarifies [80] the ansatz to include only chains over their θ temperature, that is, synonymously "swollen," "random coil," or "good-solvent" chains. This condition occurs when monomer-monomer interactions are less favorable than monomer-solvent (or in this case, when thermal motion supersedes attractive monomer interactions). Above the θ temperature, a polymer chain behaves as a self-avoiding random walk. Kumar reported the threshold reduced temperature $T^* \geq 3$, however this is likely too low for longer chains. Literature values range from 2.0 [84] to 4.3 [85], for flexible LJ chains. The collapse of θ chains is a widely studied topic (see [86] and references therein). In a study of ultra-long chains (up to $n = 10^6$) using the PERM method (see Section 2.3.5 below), Grassberger [85] calculated the θ temperature to be at least $T^* = 4.3$. Spyriouni et al. [81] performed a detailed work testing the accuracy of the modified Widom method with configurational bias insertions for realistic n-alkanes up to $n = 16$. They confirmed the increment ansatz was valid for the temperature tested (which was in the good-solvent regime).

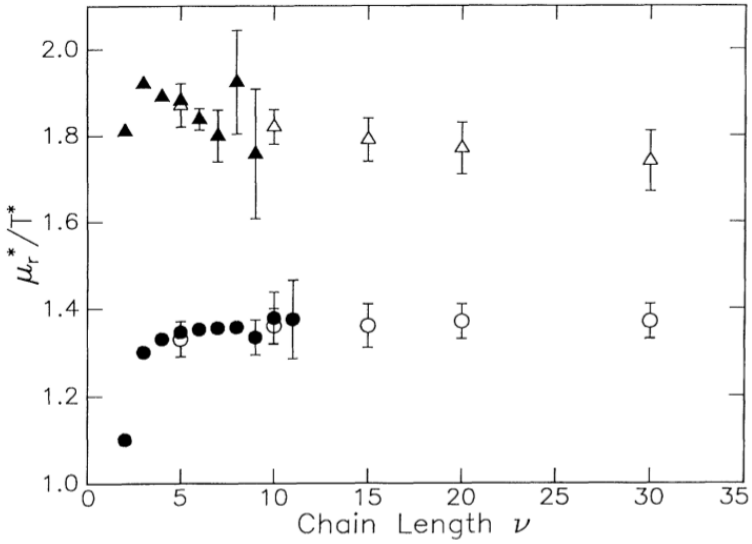


Figure 2.2: The excess incremental chemical potential of free chains at $T^* = 2$ (triangles) and $T^* = 8$, as function of chain length. From Ref. [11].

The modified Widom method suffers from three primary disadvantages. First is the aforementioned need for n independent simulations to rigorously calculate an n-mer. Second is poor sampling at high density, due to Widom-type trial insertions. Finally, the method is limited to linear chains, as complex branching morphologies would be CPU prohibitive using an incrementing strategy.

The modified Widom method is widely used, especially for conditions where the chain length ansatz is valid. Spyriouni et al. [81] calculated the chemical potential n-alkanes from 6 to 16 in length in n-hexadecane, in both vapor and liquid states. Vega et al. [87]

calculated isotherms of polymers confined in very narrow slit pores. The method has been extended to linear chains with monomers of arbitrary complexity, as well as a molecular dynamics implementation [88]. This was done by using configurational bias insertions of trial-monomers. Sheng et al. studied the vapor-liquid equilibrium of 20, 50, and 100-mers by extending the modified Widom method to the Gibbs ensemble [89].

2.3.4 Rosenbluth Insertion

The primary drawback to the previously mentioned methods is the requirement of multiple simulations to determine the chemical potential of a given state. This can be addressed, to a degree, by schemes that allow for insertion of the entire chain. To achieve this, knowledge of the system's current configuration is used to insert the molecule step-wise, with a bias to guide the chain to low-energy conformation. When used as a sampling scheme, the bias is then removed with an appropriate MC acceptance rule. This method is collectively termed *configurational bias* [72, 90, 91], and is quite useful, and popular, for effectively equilibrating a polymeric system in MC. Trial insertions based on this method sample the Rosenbluth distribution [92], and are thus termed *Rosenbluth insertions*, and with careful implementation, can accurately determine the chemical potential of oligomers and moderate-length chain molecules.

The basis of configurational bias is the Rosenbluth and Rosenbluth [92] algorithm for construction of random-walk self-avoiding polymer chains. If sampled randomly, the failure to grow a self-avoiding random-walk increases exponentially with the length of the walk. Rosenbluth and Rosenbluth used a bias sampling to strongly reduce the attrition rate of random-walks. The bias was then corrected by associating a weight to each generated configuration. This basis was extended to calculated the chemical potential of lattice polymers in a dense monolayer by Siepmann [93]. It was extended again to off-lattice polymers soon after by Frenkel et al. [72, 82] and de Pablo et al. [90]. Configurational bias generates the trial polymer step-by-step, recording the bias weights for each monomer. These weights, called Rosenbluth weights, are related to the excess chemical potential of the fully inserted chain. As with Widom methods, the generated chain is never accepted into the system configuration. To generate a trial chain [41], first a random monomer is inserted into the system, and serves as one terminal end of the trial chain. Its Rosenbluth weight is recorded as

$$w_1^{ext} = k \exp\left[-\beta\phi_{ext}^{(1)}(1)\right], \tag{2.23}$$

where *ext* refers to the external potential energy (i.e. non-bonded contributions from the fluid or external fields), as opposed to internal potential energy (*int*, the contributions from bonds and stiffness), ϕ_{ext} is the potential energy interaction of the inserted monomer, and k is the number of trials per subsequent monomer. The next step is to generate k trial positions for following monomer. These trials should be distributed by their internal Boltzmann factor, $\exp(\beta\phi_{int})$, to improve sampling. Next the external Boltzmann factor is calculated for all k trials, and its sum is the monomer's Rosenbluth weight,

$$w_i^{ext} = \sum_{j=1}^{k} \exp\left[-\beta\phi_{ext}^{(i)}(j)\right] \tag{2.24}$$

One of the trials is selected with the probability

$$p^{(i)}(m) = \frac{\exp\left[-\beta\phi_{ext}^{(i)}(m)\right]}{w_i^{ext}} \tag{2.25}$$

The selected segment m is then added to the trial chain. The sequence is repeated until the entire n-mer is generated. The scheme is illustrated in Figure 2.3. Once the entire length of chain is generated, its normalized Rosenbluth factor is given by

$$W^{ext} = \prod_{i=1}^{n} \frac{w_i^{ext}}{k} \tag{2.26}$$

The excess chemical potential can be calculated by

$$\mu_{\mathrm{ex}} = -k_{\mathrm{B}}T \ln \frac{\left\langle W^{ext}\right\rangle}{\left\langle W_{\mathrm{id}}^{ext}\right\rangle}, \tag{2.27}$$

where W_{id}^{ext} is the normalized Rosenbluth factor (2.26) of an isolated chain with its intramolecular non-bonded potential. This must be calculated from a separation simulation, usually at little computational cost.

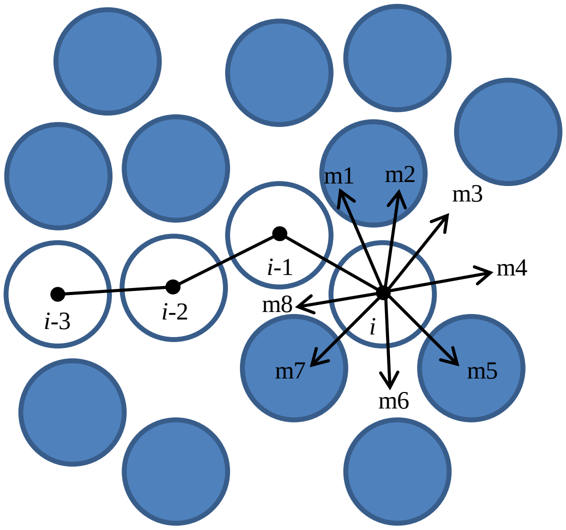


Figure 2.3: Schematic of a growing chain by configurational bias insertion. Adapted from Ref. [72].

The number of positions k is a tunable parameter; using more trials increases the likelihood of “finding” a low energy state, yet also increases computation by $O(kNn)$. There is a practical limit to increasing the number of trial moves. For a chain of 20 units, Frenkel et al. calculated an “acceptance” rate (i.e. a non-zero Rosenbluth factor) of 2.0% using 100 trials per monomer. It is generally accepted that the method is inefficient for chains longer than this.

One important consideration is that generated conformations are always “accepted,” however unlikely (that is, a chain is always generated to completion). This does not sample the Boltzmann distribution [94, 95]. Only approach true Boltzmann sampling in the limit of an infinitely long simulation. Results are accurate only if distributions overlap; long chains/short simulations will have systematic error. This was illustrated clearly in the work of Batoulis and Kremer [36]; see Figure 2.4 for details.

An obvious advantages of Rosenbluth insertions is that the chemical potential of the entire molecule can be calculated in one step. This is because the entire chain inserted into the system, rather than only incrementally as in the modified Widom approach, or over a range of parameters as with thermodynamic integration and staged particle deletion. For this reason, Rosenbluth insertions are very efficient for chains of moderate length.

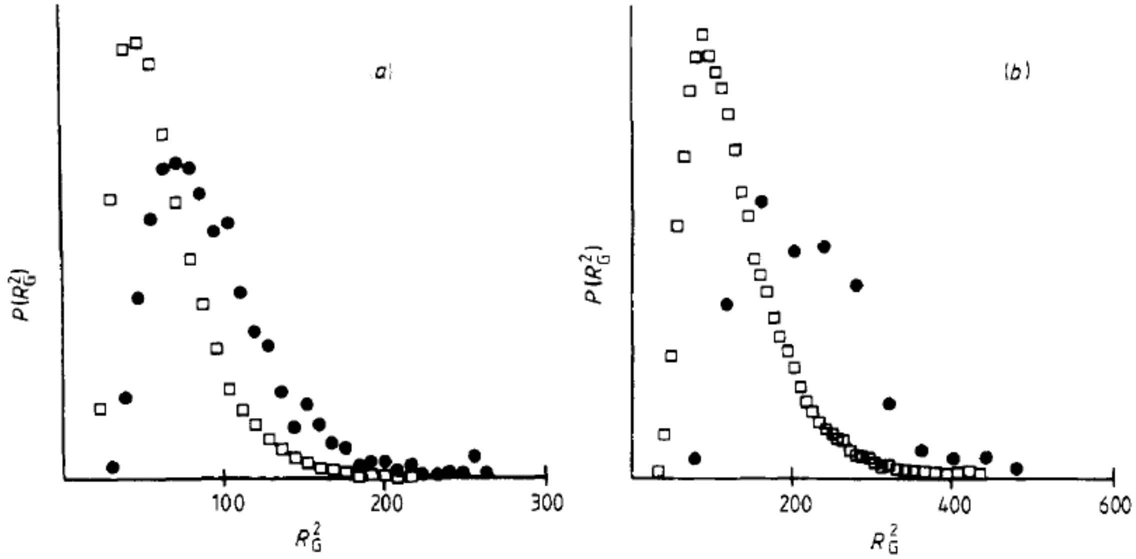


Figure 2.4: Comparison of Boltzmann (square) and Rosenbluth (circle) distributions, obtained by sampling the radius of gyration for (a) chains of length $n = 120$ and (b) $n = 240$. From Ref. [95].

However, Rosenbluth insertions are still limited by chain length. The longer an inserted chain is, the highly the probability it will 'grow' into a dead end or intersect itself. Frenkel calculated the maximum length for this method as $n = 20$, and this required a large number of trials. This is obviously affected negatively by increasing fluid densities. Another disadvantage is the fact that the scheme does not sample the Boltzmann distribution, but rather the Rosenbluth distribution. Figure 2.4 shows the difference between the distributions when sampling a chain's radius of gyration R_{G}. The correct chemical potential is only recovered when the two distributions have significant overlap.

Several improvements to the configurational bias scheme have been suggested: overlapping distribution Rosenbluth method [96] extends histogram reweighing to obtain more accurate estimation of the chemical potential. Recoil growth [97, 98] provides a framework for "restarting" a failed insertion by recoiling to previous successfully inserted monomers. Finally, early rejection [41] reduces computational time by abandoning insertions with zero weight before the full chain is generated.

There are numerous extensions of configurational bias to other methods and ensembles. Grand canonical MC [99–101] simulations was used to calculate the adsorption isotherms of alkanes and other chain molecules. Jiang et al. and Mota et al. [59–61] extended the gauge cell to include configurationally bias insertions from the gauge cell to the system cell. Similarly, Gibbs ensemble MC was extended [102] to include biased insertions between vapor and liquid phases. Finally, the modified Widom method was extended [88] to include biased insertions of arbitrarily complex monomers, i.e. monomers with side chains.

Configuration bias Rosenbluth insertion has found widespread use. Some useful examples of its application include: adsorption of alkanes in slit pores [103], polymer films [104, 105], vapor-liquid equilibria of alkanes [63, 100, 106], with many more examples available in the literature.

Recently, Garberoglio et al. introduced a Boltzmann bias [107] scheme. This method involves insertion of pre-equilibrated images. It was applied to H_2 in strong confinement, and is general suited for cases where the Boltzmann distribution is very narrow and standard sampling methods fail, such as very low temperatures. Although it has not been generalized to polymers, the authors suggest it is straightforward and advantageous to do so [108].

2.3.5 Recursive Methods

A significant drawback to the Rosenbluth-Rosenbluth algorithm [92] is the distinct possibility that a partially inserted chain is discarded. In other words, there is no foresight when growing a chain; very often it may grow into a location where no further trials are permissible, i.e. a “dead-end.” Although there are direct ways to alleviate this (e.g. the scanning method [97], mentioned above), the computational expense increases exponentially the further ahead one probes. The recursive methods of this section offer a different approach; they sample *populations* of chains, rather than follow a Markovian chain through phase-space. This allows for enrichment of probable (non dead-end) configurations. Improbable chain configurations “die,” while chains that are more probable (and thus contribute more to the Boltzmann average) take their place.

The idea of using enrichment to counter attrition due to discarded Rosenbluth chains is an old one, first suggested and implemented by Wall and Erpenbeck in 1959 [109]. In their method, partially grown chains are duplicated as starting configurations for more samples. Although similar algorithms have been proposed (e.g., Grishman [110], Garel and

Orland [111], Higgs and Orland [112], and Velikson et al. [113]), we will focus on those proposed by Hegger and Grassberger [114, 115] and Grassberger [116] for several reasons. These algorithms search populations of chains *depth* first, rather than *breadth* first. This results the ability to call the main function recursively, and thus is very efficient. Also, depth first searching requires the storage of only one conformation at a time, enabling calculation of very long chains. Finally, these methods are well established in the literature, with several extensions and many practical applications reported.

This method of recursive sampling by depth for off-lattice polymers was proposed by Grassberger and Hegger [115]. Various earlier works detail lattice models, see Refs. [86, 86, 116, 117], and references therein. The goal of this method is to calculate a Monte Carlo estimate of the partial partition function. The configurational part of the canonical partition function (2.3) is

$$Z_N = \int dr_1 \ldots dr_N \exp\left[-\Phi\left(r^N\right)/k_{\mathrm{B}}T\right] \tag{2.28}$$

As above, $\mathbf{r}$ refers to either a vector or matrix of coordinates, with a subscript referring to the coordinates for a given particle, and the superscript denoting a collection particle coordinates, and Φ is the potential energy of the system. To write the partition function recursively, we consider a partial partition function that describes a system of Ni particles interacting with a fixed background of the first $i\ +1$ particles,

$$Z_{N-i|i}(r_1, \ldots, r_i) = \int dr_{i+1} \ldots dr_N \exp\left[-\sum_{j=i+1}^{N} \phi_j(r_1 \ldots r_j) \Big/ k_{\mathrm{B}}T\right] \tag{2.29}$$

This can be written recursively as

$$Z_{N-i+1|i-1}(r_1, \ldots, r_{i-1}) = \int dr_i \exp\left[-\phi_i(r_1 \ldots r_i)/k_{\mathrm{B}}T\right] Z_{N-i|i}(r_1, \ldots, r_i) \tag{2.30}$$

Note that $Z_{0|N}(r_1, \ldots, r_N) \equiv 1$ and $Z_{N|0}(r_1, \ldots, r_N) \equiv Z_N$. Equation (2.30) is the basis of recursive sampling. The goal is to calculate estimates of the partial partition function using this relation. This is done by averaging states at the end of the chain first, while working back towards start. The partial partition function is sampled after decomposing the potential energy ϕ_i into ideal $\phi^{(0)}$ and excess $\Delta\phi_i$ contributions, where the ideal partition function $Z^{(0)}$ can be determined analytically (or at least independent of the current sampling scheme),

$$\begin{aligned} &Z_{N-i+1|i-1}(r_1,\ldots,r_{i-1}) = \\ &\quad Z^{(0)}_{N-i+1|i-1} \lim_{M\to\infty} \frac{1}{M}\sum_{k=1}^{M} \exp\left[-\Delta\phi_i(r_1\ldots r_{i-1},\xi_k)/k_{\mathrm{B}}T\right] \frac{Z_{N-i|i}(r_1,\ldots,r_{i-1},\xi_k)}{Z^{(0)}_{N-i|i}(r_1,\ldots,r_{i-1},\xi_k)} \end{aligned} \tag{2.31}$$

where ξ_k is a randomly generated trial distributed according the internal Boltzmann weights. To compute (2.31) in an MC scheme, a weight w_i should be assigned to ξ_k to remove the bias from its selection. Alternatively, one could replace ξ_k with $p_i w_i$ copies of itself (labeled α below), each counted with a unit weight. p_i is an arbitrary parameter that controls the resulting size of the population. Note that p_i is independent of $\mathbf{r}_i$ and ξ_k, and need not be constant during the simulation. Under this scheme, Equation (2.31) reduces to

$$Z_{N-i+1|i-1}(r_1,\ldots,r_{i-1}) \approx \frac{1}{p_i}\sum_{k=1}^{M}\sum_{\text{replicas}\alpha} Z^{[\alpha]}_{N-i|i-1}(r_1,\ldots,r_{i-1},\xi_k) \tag{2.32}$$

In this way, the resulting population of chains will be correctly weighed relative to the partition function. Thermodynamic properties can be obtained during growth, or from the statistics of surviving chains. Selection of the parameter p_i is critical; too large and the population size will grow quickly, and too small will lead to poor sample sizes. As a heuristic, the authors [115] suggest the following for selecting the population parameter: set $p_i w_i(\xi_k) = k + \eta$, where k is an integer and $\eta \in [0, 1)$, the algorithm should be called recursively k times, with one additional call with the probability η. In this way, improbable configurations with low weights w_i will 'die', and probable configurations will replicate.

In practice, the recursive enrichment method is implemented by a single recursive function, which is called with a position and current position along the chain i. It generates a new position at i+1, and is called with this position approximately k times, or dies, with weights as described above. The chemical potential is obtained from the average number of surviving chains at the end of the n-th step,

$$\mu_{\mathrm{ex}}(n) = -k_{\mathrm{B}}T \ln \left(\frac{\langle N_n \rangle}{\prod_{i=0}^{n} p_i} \right) \tag{2.33}$$

Here, N_n is the average number of surviving chains after $i = n$ steps. The recursive algorithm is quite efficient; computation scales with chain length n as $O(n^2)$ [115], and the recursive structure only requires that the chain's configuration up to the current point i is stored. Memory requirements are minimal and thus very long chains can be calculated. Finally, the algorithm generates a population of chains by their Boltzmann weight, and not their Rosenbluth weight as with configurational bias insertions.

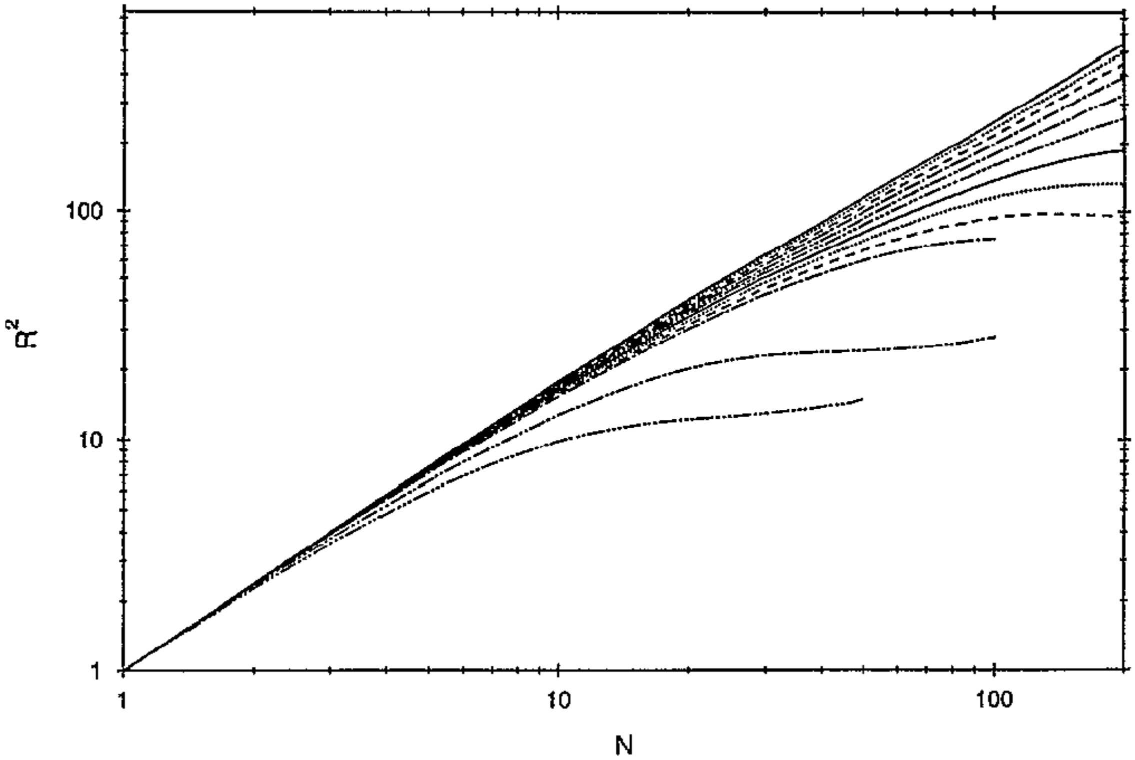


Figure 2.5: Averaged squared end-to-end distances of free chains calculated using the recursive enrichment algorithm, for inverse temperatures $\beta = 0.175$ to 0.375 (with $\Delta\beta = 0.025$), and 0.4, 0.6, and 0.833. Theta collapse is evident at $\beta \geq 0.25$ ($T^* = 4$). From Ref. [115].

A significant extension to the recursive enrichment algorithm is call pruned-enriched Rosenbluth method, or PERM [85, 118]. PERM is a strategy for managing the population of chains so that they optimally cover phase space. It does so by adding additional tuning

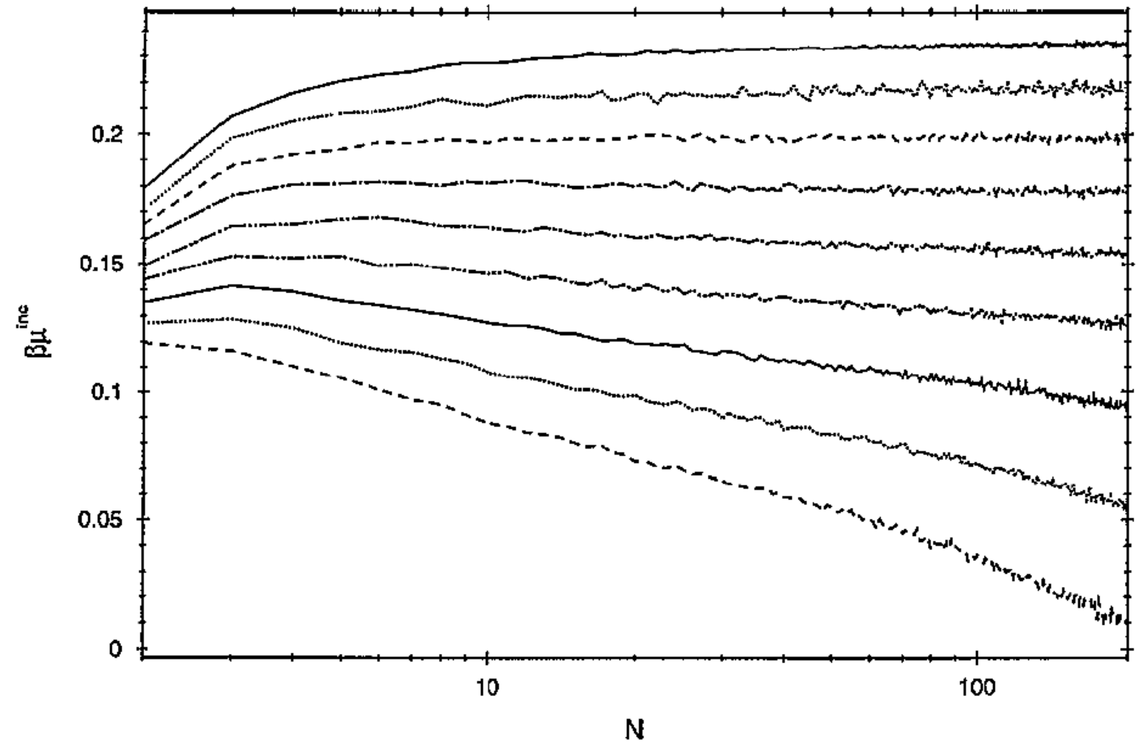


Figure 2.6: Incremental chemical potential of free chains calculated using the recursive enrichment algorithm, for inverse temperatures $\beta = 0.175$ to 0.375 (with $\Delta\beta = 0.025$). The chain increment ansatz is found to be valid for $\beta \geq 0.225$ ($T^* \geq 4.44$). From Ref. [115].

parameters to the recursive scheme that favor high-Boltzmann weight chains and having low-weight chain have a high probability of dying. This is done without adding bias: favorable chains are "enriched" by adding multiple copies and reducing their weight proportionally, while unfavorable chains are "pruned" by discarding half, while doubling the remaining chain's weight. The tuning parameters that PERM introduces are a high limit for the weight of a current replica, a corresponding low limit, and the number of copies that the enrichment portion should create.

The primary drawbacks of this approach is the fact it is a static MC algorithm. This means the chain cannot interact with another (non-fixed) atoms or molecules. In other words, only isolated chains with fixed external interactions can be calculating using recursive/PERM methods. The obvious benefit is that the algorithm generates uncorrelated configurations for each sample, as opposed to importance sampling, where successive samples are highly correlated. Attempts to extend PERM to traditional dynamic Markov chain MC [119, 120] showed no benefit over a comparable configurational bias scheme. Another disadvantage is the need for selection of tuning parameters, although several methods of dynamic generation are suggested by the authors [115]. Finally, the method is designed for linear chains, and an extension to branched polymers is not straightforward.

The recursive-enrichment algorithms have been applied to several important problems, beginning with extensive studies of the θ chain collapse [85, 86] (see Figure 2.5 for example of off-lattice collapse). Recent work includes extending to simple models of protein folding [66, 121]. For a recent review of work using PERM, see Ref. [122].

2.3.6 Expanded Ensembles

Attempting to combine the advantages of several of the schemes above, Escobedo and de Pablo introduced the expanded variable-length chain method (EVALENCH) [123]. This approach utilizes the expanded ensemble ideology introduced by Lyubartsev et al. earlier [124], which allows for the accurate determination of a full chains chemical potential in a single simulation. The basic methodology of EVALENCH is the insertion and deletion of subchains using a configurational bias technique onto a tagged chain. A preselected number of component subchains of the tagged chain are allowed, which are then defined as different states of the expanded ensemble and can be combined in a rigorous fashion. Thus the system configuration and test chain can change in course of simulation. In this fashion, incremental [11], configurational bias [72, 91], and expanded ensemble [124] methods are combined to calculate the chemical potential of arbitrary polymers. A brief overview of expanded ensembles and implementation of EVALENCH follows [123].

An expanded ensemble is used to calculate the ratio of partition functions. A convenient case is the ratio of $Q(N+1)$ and $Q(N)$, which is of course the definition of the chemical potential (2.8a), (2.12a). The expanded canonical ensemble can be written as

$$\Omega = \sum_{m=1}^{M} Q(N, V, \gamma_m) \exp(\psi_m) \tag{2.34}$$

where γ_m is an adjustable parameter that characterizes the m-h state, and ψ_m is a positive weight factor. The canonical ensemble partition function is recovered when summation of states $m = 1$ to M is along a continuous path in m. To sample along this path, transitions between neighboring states are allowed. The probability of observing a given state is

$$p(m) = p(\gamma_m) = \frac{Q(N, V, \gamma_m)}{\Omega} \exp(\psi_m) \tag{2.35}$$

Note that this quantity can be easily tabulated during simulation making a histogram of transitions between states. The ratio between two states m_1 and m_2 is given by

$$\frac{Q(N, V, \gamma_{m_1})}{Q(N, V, \gamma_{m_2})} = \frac{p(m_1) \exp(-\psi_{m_1})}{p(m_2) \exp(-\psi_{m_2})} \tag{2.36}$$

The above ratio of partition functions can be used to determine chemical potential if m_2 characterizes a system of N particles at constant V, T, and m_1 an otherwise identical system with $N+1$ particles. The expanded ensemble is realized if γ allows these two states to be connected in a smooth fashion. Thus, the expression for chemical potential from an expanded ensemble is,

$$\mu_{\text{ex}} = -k_{\text{B}} T \ln \left[\frac{Q(N, V, T, \gamma_M)}{Q(N, V, T, \gamma_1)} \right] = k_{\text{B}} T \ln \left[\frac{p(1)}{p(M)} \right] + \psi_M - \psi_1 \tag{2.37}$$

The above expression can be directly applied to a polymeric system by constructing a system with a gradually inserted chain, where γ_1 represents a system of N polymer chains with 1 tagged non-interacting system, and γ_M a system with the fully interacting tagged chain, i.e. $N+1$, similar to the thermodynamic integration method described above. This method was applied by Wilding and Müller for the lattice-based bond-fluctuation model [79]. Note that the preweights ψ_i must be specified before the simulation starts.

The EVALENCH method extends the expanded ensemble approach to off-lattice chains with arbitrary connectivity by utilizing the incrementing strategy of Kumar [11] and the efficient insertions of short chains with configurational bias [72, 91]. This is accomplished by first rewriting Equation (2.34) in terms of an incremental chain,

$$\Omega = \sum_{k=1}^{M} Q(N, V, m_k) \exp(\psi_k) \tag{2.38}$$

where m is the current length of an additional tagged n-mer in an N, V, T system; $m_1 = 0$ and $m_M = n$. Note that the incremented amount need not be a single monomer. Transitions between neighboring states (e.g. $m_k \rightarrow m_k + 1$ and $m_k \rightarrow m_k 1$) are permitted. Insertions are selected from a set of N_{samp} according to configurational bias rules,

$$W_i = \frac{w_i}{\sum_{j=1}^{N_{\text{samp}}} w_j} \tag{2.39}$$

where w_j is the Boltzmann factor of the j-th trial configuration (as in Equation (2.24)). The probability of transitions between neighboring states is determined by the detail balance,

$$P_{\text{acc}}(m_y \rightarrow m_x) = \min\left[1, \frac{T(x \rightarrow y)}{T(y \rightarrow x)} \frac{p(x)}{p(y)}\right] \tag{2.40}$$

where $p(x)$ is the probability of observing a system at state x (2.35), and $T(x \rightarrow y)$ is the probability of initiating a transition from state x to y. The ratio $p(x)/p(y)$ is

$$\frac{p(x)}{p(y)} = \exp\left[-\frac{\phi(m_x) - \phi(m_y)}{k_{\text{B}}T}\right] \exp\left(\psi_x - \psi_y\right) \tag{2.41}$$

where ϕ is the interaction energy of the tagged chain with the rest of the system. Finally, the general expressions for the transition probabilities T is given by

$$T(b \to a) = \frac{1}{K_b} \prod_{j=m_b}^{m_a} N_{\mathrm{samp}} W_j \tag{2.42}$$

$$T(a \to b) = \frac{1}{K_a} \tag{2.43}$$

given $m_a > m_b$ so that (2.42) represents an incremental insertion, and (2.43) a similar deletion. The variable K_i describes the connectivity, the i-th section having K_i number of neighboring states (a linear homopolymer would also have $K_i = 2$, except for states 1 and M, the beginning and end of the chain). The incremental chemical potential is then given by

$$\mu_{\mathrm{ex}}^{\mathrm{inc}}(m_\alpha \to m_\omega) = k_{\mathrm{B}} T \ln \left[\frac{p(m_\alpha)}{p(m_\omega)} \right] + \psi_\omega - \psi_\alpha \tag{2.44}$$

The chemical potential of the entire chain is determined by setting $\alpha = 1$ and $\omega = M$.

The primary advantage of EVALENCH is that the chemical potential of an n-mer can be calculated in a single simulation. Also, as formulated [123], there is no restriction to linear chains. However, EVALENCH does require a preliminary calculation to obtain the preweight parameters ψ_i. Careful selection of the preweights is necessary. Improperly chosen weights would negatively affect sampling, for example, by not allowing a chain to grow to its finished state. Assuming a uniform distribution of $p(m)$, it follows from Equation (2.44) that $\psi_\omega - \psi_\alpha = \mu_{\mathrm{ex}}^{\omega}(m_\alpha \to m_\omega)$. Thus a single short simulation can usually determine adequate preweight values.

EVALENCH was used to study the purely-repulsive case of hard core chain fluids [123], for lengths of 4, 8, 16 and 32 in packing fractions up to 0.4. It was extended to open systems in the grand canonical and Gibbs ensemble [125]. Recently, it was rigorously optimized in closed and open systems [126, 127] and extended to transition matrix MC [128].

2.3.7 Scanning Methods

The scanning method was introduced by Meirovitch and calculates the entropy directly by recording the transition probabilities to a future (finished) system [129]. In this way, the probability P of a given configuration is known (as it is the product of the transition probabilities), and thus its entropy is known (since $S \sim \ln P$). Pressure and chemical potential can then be easily calculated from standard thermodynamic relationships. This method was initially applied to a self-avoiding polymer melt on a lattice [129]. However, application of the scanning method is limited to static system that can easily constructed. To overcome this, Meirovitch developed the hypothetical scanning (HS) method to couple entropy prediction of the scanning method with the flexibility of the Metropolis Monte Carlo method [130]. HS assumes a large system in equilibrium and produces an estimate of P by sampling the MC transition probabilities. The HS method was later extended to continuum chain models using configurational bias with an incremental approach [131] for efficient simulation of chain molecules.

The hypothetical scanning method was rigorously extended to LJ MC systems [129, 130] and MD systems [132]. The primary advantage of scanning methods is the wealth of detailed thermodynamic information a single simulation can produce. Quantitative knowledge of absolute entropy allows for exact values of pressure and chemical potential. A significant disadvantage is the requirement of new formulation of HS for different MC moves. New, complex moves that are needed to equilibrate polymer systems would require independent implementation and validation of the HS method.

2.3.8 Staged Particle Deletion

The standard implementation of the Widom method is based on a forward-difference definition of the chemical potential (2.3). As discussed above, it is progressively difficult to sample dense fluids with some Widom trial insertions. The “reverse” or “inverse” Widom method is mathematically identical (in the thermodynamic limit) [133]. Instead of inserting a trial particle to measure $\Delta F(N+1)$, a random particle is “removed” to calculate $\Delta F(N-1)$. Of course, this particle is not actually removed. The inverse Widom method suffers from poor statistics as the removed particle is necessarily in local equilibrium with its surroundings; thus it is difficult to fully sample phase space. In other words, the removal of a particle as a trial creates a volume bias (the “hole” the trial particle was occupying) in the system during sampling. Boulougouris et al. developed a rigorous approach to account for this bias called the “staged particle deletion” method [43]. This is done not by comparing the N and $N-1$ systems, but including an intermediate system with $N-1$ particles and a hard sphere of radius r_{core} in place of the removed particle. The free energy difference between the N and $N-1$ systems is then the calculated by the free energy differences between N and $N-1$ with the intermediate system (see Figure 2.7). Their method was tested successfully for a system of hard spheres (for which the Widom method as derived above would fail), and an LJ fluid [43], and found to be efficient and accurate.

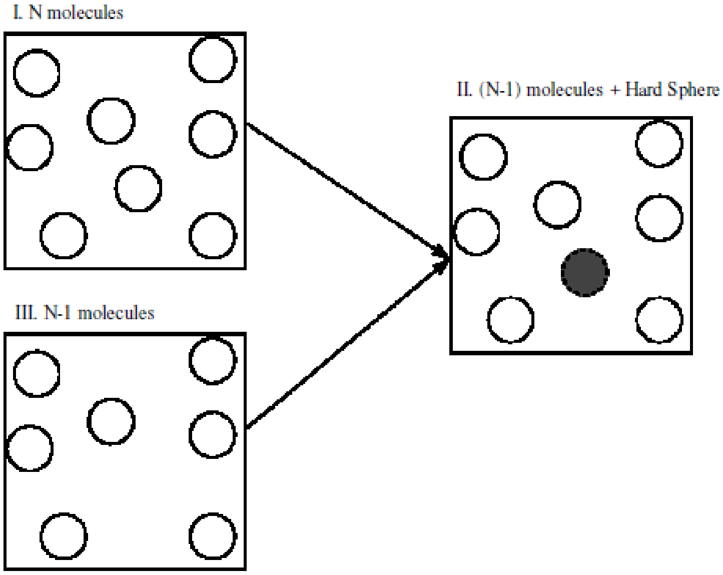


Figure 2.7: Schematic of the staged particle deletion method that calculates the chemical potential of dense fluids by the inverse Widom method. From Ref. [43].

The staged particle deletion method was extended to chain molecules by Boulougouris et al. [134]. Their implementation included two additional methods to improve efficiency: excluded volume map sampling method [86], and analytical calculation of the accessible volume [135]. The former increases the likelihood of efficient insertions, while the latter quickly calculates the free energy of cavity formation, i.e. the contribution from the intermediate system (panel II in Figure 2.7). The extension to chain molecules is straightforward; the volume contribution from a chain removal is sampled by the generation of a hard sphere chain. See Ref. [134] for a thorough derivation in the isothermal-isobaric ensemble. The authors used their method to calculate the equation of state for LJ dimers, and found it quite efficient, using about a tenth of CPU time that a similar Widom method would take.

The staged particle deletion method is most applicable to dense fluids and in fact suffers at low densities when sampling the volume contribution takes an appreciable amount of time. It is best suited to chains of modest length in melt conditions. Recently application of the method was found in the study of competitive adsorption of carbon monoxide, carbon dioxide, and methane on activated carbon [136].

2.4 Summary

The calculation of polymer chemical potential and free energy is a practically important problem, whose significance is continually rising. In light of current biology problems, and the requirement of complex phase predictions of nano-structured systems, the need for a clear understanding of suitable methods is obvious. Many approaches to the calculation of chemical potential of polymer chains have been suggested. However, each method has its respective advantages and disadvantages. It is the goal of the MC practitioner to successful apply the correct method to their specific problem.

Chapter 3

Monte Carlo Studies of Cavitation in Nanopores

3.1 Cavitation in Metastable Liquid Nitrogen Confined to Nanoscale Pores

3.1.1 Introduction

Behavior of fluids confined to nanoscale pores has been attracting considerable interest among several generations of chemists, physicists, and materials scientists [17–20, 137, 138]. Starting from the seminal works of Zigmondi in the beginning of 20th century [16], this interest was triggered mainly by the use of physical adsorption and capillary condensation phenomena for characterization of surface area and porosity of adsorbents and catalysts. Recent discoveries in nanotechnology and biomedicine have significantly expanded the area of practical applications of confined fluids to the nanomaterials synthesis, nanofluidics, and drug delivery. At the same time, the availability of novel materials with controlled pore structures has opened up new opportunities for detailed experimental studies of the mechanisms of phase transformations in confined fluids.

Major progress has been achieved in the understanding of adsorption, capillary condensation and desorption phenomena in highly ordered mesoporous materials with simple pore geometries, such as MCM and SBA mesoporous crystals [25, 139, 140]. However, there are still many open questions concerning adsorption-desorption mechanisms in more complex porous systems. Fluids adsorbed in hierarchically structured micro-mesoporous materials exhibit great a variety of hys-teretic adsorption-desorption isotherms with multiple steps related to phase transformations in adsorbed phases. Adsorption-desorption processes involve a combination of physical mechanisms, such as delayed condensation, advanced condensation, cavitation induced evaporation, pore blocking, and percolation, which are reflected in characteristic types of the hysteresis loops formed by adsorption and desorption isotherms [21, 35, 141]. The complexity of hysteresis loops causes a considerable complication for the pore stucture characterization; but if interpreted correctly, they provide important information about the pore network morphology, which is crucial for discriminating physical mechanisms of phase transformation [36, 142].

On the pore level, capillary condensation hysteresis can be regarded as an intrinsic property of vapor-liquid phase transitions in finite volume systems [20, 21]. A classical scenario of capillary condensation implies that the vapor-liquid transition occures at a vapor pressure p_{cc} that exceeds the pressure p_{e} of vapor-liquid equilibrium in the pore, but is below the limit of stability for the vapor-like states (a.k.a. vapor-like spinodal p_{SV}). At $p_{\mathrm{e}} < p < p_{\mathrm{SV}}$, vapor-like states are metastable, and condensation is associated with a certain nucleation barrier, which cannot be overcome within the timeframe of experiment. In relatively narrow pores, where the nucleation barrier is low, $p_{\mathrm{cond}} \approx p_{\mathrm{e}}$, while in wider pores with high barriers, condensation occures close to the spinodal pressure $p \approx p_{\mathrm{SV}}$. According to the classification [21], the former regime is called the regime of reversible condensation, and the latter, the regime of developed hysteresis. The hysteresis type depends on the pore size and shape. In particular, for nitrogen adsorption at at 77.4 K, reversible condensation occurs in cylindrical pores smaller than $\sim$4 nm, and developed hysteresis occurs in cylindrical pores wider than $\sim$5 nm. In intermediate pores, the width of the hysteresis loop gradually increases and the position of condensation shifts from p_{e} to p_{SV}. This regime is known as the regime of developing hysteresis. In open uniform cylindrical or slit-like pores, desorption occurs without nucleation at $p_{\mathrm{c}} = p_{\mathrm{e}}$ via the meniscus receding from an open pore end, and therefore, no metastability is observed. This scenario is not valid for more complex pore structures, such as SBA-16, KIT-5 silicas, etc., where wider pores are connected by narrower pores or "necks" [35, 36].

Two basic mechanisms of desorption in pore networks are distinquished as pore blocking percolation and cavitation. The former mechanism was introduced in the early studies of capillary hysteresis, and is referred to as "ink-bottle" or "classical pore blocking" mechanism [20, 137]. It is well understood that evaporation of the capillary condensate from a network of ink-bottle pores is hindered by the pore constrictions. In this case, desorption from the pore body may occur only after emptying of its neck. In other words, desorption from the neck triggers evaporation in the blocked pore. Thus, the vapor pressure $_{\mathrm{d}}$ of desorption from the pore body depends on the neck size and network connectivity. The onset of evaporation from the pore network is associated with the percolation threshhold and the formation of a continuous cluster of pores open to the external surface [32, 33, 141, 143, 144]. The percolation mechanism is observed in pore networks with sufficiently large necks.

Theoretical and experimental studies [35, 36] have revealed that if the neck diameter is smaller than a certain critical size (estimated to be ca. 5 nm for nitrogen at 77.4 K), the mechanism of desorption from the pore body involves cavitation — the spontaneous nucleation and growth of gas bubbles in the metastable fluid condensed in the pore — while the neck remains filled. In this case, the desorption transition pressure p_d does not depend on the size of pore necks and is determined by the properties of the adsorbate, and as we show below by the size and shape of the pore body. Both desorption mechanisms, percolation and cavitation, originate from geometrical pore blocking; however, the physical mechanisms and the pore size dependence are quite distinct. The pore blocking effects cause a "delay" in desorption, i.e., desorption occures at vapor pressure below that of the equilibrium vapor-liquid transition in the given pore ($p_\mathrm{d} < p_\mathrm{e}$). Thus, in the course of desorption, one observes long-living metastable states of condensed fluid that is overstretched due to the action of capillary pressure.

In this section, we focus on the cavitation mechanism of desorption. Cavitation is broadly defined as spontaneous formation and activity of bubbles in metastable liquids [145]. In the process of desorption, a critically sized bubble serves as a nucleus for forming the vapor-like phase in a metastable (overstretched) liquid. In this light, cavitation is a nucleation phenomenon. In other words, evaporation must be preceded by the formation of a critical cavity (bubble) [55]. Creation of a critical cavity is associated with a free energy barrier to overcome, i.e., the nucleation barrier. Cavities smaller than a certain critical size collapse back into the liquid state. Cavities that reach the critical size grow spontaneously and initiate a transition from liquid state to vapor state, since the latter is thermodynamically more stable. This process is similar to cavitation processes in bulk liquids, which are widespread in nature, physiology, and technology. Historically, cavitation has been viewed as a negative effect of bubble formation in flow fields, particularly concerning the design of impellers. It is also the culprit of "diver's sickness", caused by formation of nitrogen bubbles in the blood during decompression. Nowadays, cavitation is widely employed in technology and medicine. Stable, non-transient cavitation via a lithotripter is used to break kidney and gall bladder stones [146]. Recently, it has been suggested that a similar method can be used to non-invasively treat solid tumors [147]. Sonophoresis, a technique that increases permeability of lipid bilayer membranes to drugs, is also based on cavitation [148]. A better understanding of the formation of critical nuclei that precede cavitation is vitally important for the design and practical implementation of these and other cavitation-based technologies.

The study of nucleation in bulk liquids is notoriously difficult; although a metastable liquid phase can be achieved via superheating, measurements of the energetics and the rate of formation of critical nuclei is challenging due to natural fluctuations and impurities. Capillary evaporation of fluids from mesoporous materials offers a unique opportunity to monitor the very onset of cavitation. Firstly, a confining solid matrix limits temperature and pressure fluctuations that are always present in macroscopic systems. Secondly, the adsorbing pore walls are covered by liquid-like adsorbed films, which protect the interior of the pores (where cavitation takes place) from surface pollutants and irregularities that serve as nucleation centers and thus as facilitators of spontaneous heterogeneous cavitation in bulk liquids. The creation of nuclei occurs without heterogeneous nucleation sites and is purely driven by thermal fluctuations at given external thermodynamic conditions, thus cavitation in desorbing fluids is homogeneous. A thorough understanding of cavitation phenomena involved in desorption of fluids from pores and pore networks is not only of interest for characterization of nanoporous materials, as mentioned above, but it can provide useful insight to the nature of cavitation events. Since cavitation is a widespread phenomenon, an improving knowledge of cavitation onset and energetics may have far-reaching implications.

In a continuation of our previous work [27, 35, 36, 52, 55, 142, 149] we study here the cavitation of nitrogen condensed in pores of well characterized mesoporous silicas of two types: ordered mesoporous crystals (SBA-16) and hierarchically structured materials (KLE, KLE/IL and SLN-326). Well-defined pore morphologies of these samples, uniformity of pores, and high precision of the measurements enable quantitative insights into cavitation from a comparison of the experimental isotherms on different samples with the results of molecular simulation of fluid desorption from individual spherical pores. The focus is made on establishing relations between the conditions of cavitation and the pore size and shape.

3.1.2 Methods and Materials

Materials

SBA-16 is a member of the SBA-n family [25]. It is a 3D silica mesoporous crystal that is comprised of a network of cage-like pores placed at the sites of a body-centered cubic lattice [150], which belongs to the $Im\overline{3}m$ space group. Its pore structure can be described, according to electron crystallographic studies, as being wrapped by a periodic minimal I-WP surface of Schoen [151], where regularly spaced cages are connected by significantly smaller windows (pore entrances) [150, 152]; each cage has eight entrances. SBA-16 materials are generally prepared via templated synthesis from segregated non-ionic tri-block copolymers and are characterized by uniform pore size distributions (PSD) with tunable pore dimensions (ranging from 6 to 12 nm), high surface area, and good thermal and mechanical stabilities [153, 154]. SBA-16 intrinsically possesses complementary micro- and narrow meso- pores in its wall structure, in addition to the network of mesopores ordered with $Im\overline{3}m$ symmetry [155]. For our study, we selected SBA-16 materials with pore diameters varying from 6.5 to 12 nm. Nitrogen adsorption-desorption isotherms for these materials were either obtained from literature (see Table 3.1), or measured on freshly synthesized SBA-16 samples [156]. Pore size information was obtained by first measuring high-resolution nitrogen sorption isotherms, and then by applying a hybrid NLDFT method [36] to the adsorption branch of the isotherm which takes into account the effect of delayed capillary condensation in the metastable pore fluid (see also section 3.1.4).

Hierarchically structured silica materials of KLE type were prepared according to the synthesis scheme described previously [157]. In aqueous solutions, KLE block copolymer forms an FCC lattice of isolated spherical micelles. In templated silica matrix, these micelles produce spherical mesopores that are connected through small micropores, originating from the hydrophilic poly(ethylene oxide) blocks penetrating the silica matrix. The pore structure of KLE silica, as confirmed by SAXS and TEM studies [36, 157], can be described as spherical mesopores with a pore diameter of ca. 14 nm, connected via narrow (~ 1 nm) micropores in the pore walls. This represents an ideal morphology to study the adsorption and desorption phenomena.

KLE/IL silica was synthesized using the same block copolymer as in the case of KLE-silica with an ionic liquid used as a second template [158]. This leads a trimodal pore size distribution. KLE/IL silica contains main spheroidal mesopores of 11-14 nm in diameter connected by micropores of ∼1 nm in diameter similar to those in KLE-silica's. In addition, there are narrow cylindrical mesopores of 2-3 nm in diameter (from the ionic liquid). The compositions of the parental solutions used for KLE and KLE/IL silica were optimized to ensure accessibility of mesopores [36, 158, 159]. KLE-C23 has a pore structure similar to that of KLE/IL silica, but with spheroidal pores in the range of 24 nm [160].

A novel SLN-326 silica material with a hierarchical pore structure was prepared by employing a copolymer mixture of SE1010 ("SE" = (poly(styrene)-(poly(ethylenoxide)) + STEOM (Poly (styrene)539-co-poly((3triethoxysilyl)propylmethacrylate)73) as a template (a ratio of STEOM /SE1010 = 0.75 was used). SLN-326 silica consists of spherical mesopores of diameter ∼35 nm, which are connected through ∼5 nm worm-like mesopores and additional micropores [159].

3.1.3 High-resolution adsorption measurement

Nitrogen (77.4 K) and argon (77.4 K, and 87.3 K) adsorption/desorption isotherm measurements were performed with an Autosorb-I-MP adsorption instrument (Quantachrome Instruments, Boynton Beach, FL) in the relative pressure range from 1×10^{-6} to 1. The analysis station of the volumetric adsorption apparatus was equipped with both the standard pressure transducers, in the dosing volume (manifold) of the apparatus, and with high precision pressure transducers (Baratron MKS), dedicated to read the pressure in the sample cell itself. Hence, the sample cell is isolated during equilibration, which ensures a very small effective dead volume and therefore a highly accurate determination of the adsorbed amount. The saturation pressure p_0 is measured throughout the entire analysis by means of a dedicated saturation pressure transducer, which allows the vapor pressure to be monitored for each data point. This improves the precision in determination of p/p_0 and the accuracy of the adsorption measurements. The samples were out-gassed overnight at 150 °C prior to the adsorption analysis.

Table 3.1: Summary of Experimental Data Used for Comparison with Simulated Isotherms[a]

sample	geometry	NLDFT internal pore diameter [nm]	p_{cav}/p_0	simulated pore, accessible/internal [nm]	nucleation barrier [$k_{\mathrm{B}}T$]	figure	source
SLN-326	sphere	35	0.487	-	75.2[b]	3.1,3.3,3.4	New
KLE-C23	sphere	24	0.484	-	73.7[b]	3.1,3.4	New
KLE/IL	sphere	13.9	0.480	-	72.5[b]	3.1,3.4	[36]
KLE	sphere	13.9	0.482	-	-	-	[36]
SBA16-130C	sphere	11.7	0.480	-	71.4[b]	3.1,3.5	[155]
SBA16-10.5	sphere	10.5	0.471	-	66.4[b]	3.3,3.5	New
SBA16-100C	sphere	9.5	0.460	-	-	3.5,3.9a	[155]
SE3030	cylinder	9.4	0.482	-	-	3.5	[155]
SBA16-9.4	sphere	9.4	0.463	9.56/9.46	60.0	3.1	New
SE10C16	sphere	9.2	0.456	-	-	3.5	[159]
SBA16-8.8	sphere	8.8	0.458	9.02/8.92	57.3	3.3	New
SBA16-80C	sphere	8.2	0.439	8.29/8.19	53.8	3.5	[155]
SBA16-7.9	sphere	7.9	0.442	8.02/7.92	46.9	3.3	New
SBA16-60C	sphere	7.1	0.414	7.22/7.12	41.6	3.5,3.9b	[155]
SBA16-6.9	sphere	6.9	0.425	-	-	3.3	New
HMM-3	sphere	5.44	0.352[c]	5.54/5.44	30.3[c]	3.11	[161]
SLN-326	cylinder	5	0.421	-	-	3.4,3.5	New

[a] Pore size determined by NLDFT. Cavitation pressures obtained by measurement of the midpoint of the desorption step.

[b] Noted nucleation barriers were extrapolated from the theoretical relationship for 9.56 nm pore; see Fig. 3.10.

[c] The HMM-3 sample is considered just for comparison. Due to the small size of pores, the HMM-3 isotherm is reversible and does not exhibit hysteresis, so the pressure listed is the equilibrium pressure, rather than the cavitation pressure.

3.1.4 NLDFT method for pore size characterization

The adsorption data were analyzed using a hybrid NLDFT approach that allows quantification of both micro- and mesopores [36]. The hybrid NLDFT method allows one to calculate PSDs in materials containing pores of different geometry. The method is applied to the complete range of micro- and mesopores. Calculations of PSDs are performed from the adsorption branch of the isotherm. For materials with spheroidal pores, the method uses the spherical pore model in the region of hysteresis, and the cylindrical pore model in the region of reversible filling of micropores and narrow mesopores. In the region of hysteresis, the method takes into account the effect of delayed condensation, and uses the NLDFT metastable adsorption isotherms, while in the region of reversible capillary condensation/desorption the method uses NLDFT equilibrium isotherms. We have demonstrated successful application of this method in previous works [27, 28, 162, 163].

3.1.5 Monte Carlo simulation studies

Monte Carlo (MC) simulations were carried out to model N_2 adsorption in spherical silica pores of 5.54 nm, 7.22 nm, 8.02 nm, 8.29 nm, 9.02 nm, and 9.56 nm in accessible diameter. These sizes correspond to internal pore diameters of selected SBA-16 samples studied experimentally (Table 3.1). Nitrogen was modeled as a spherical Lennard Jones (LJ) particle of effective diameter $\sigma_{\mathrm{FF}} = 0.36154$ nm and LJ energy parameter $\epsilon_{\mathrm{FF}}/k_{\mathrm{B}} = 101.5$ K [162]. The fluid-fluid potential was truncated at $r_c = 5\sigma_{\mathrm{FF}}$. The solid-fluid interaction was modeled as mean-field "smeared-out" LJ potential [27], with parameters [164] of $\sigma_{\mathrm{SF}} = 0.317$ nm and $\epsilon_{\mathrm{SF}}/k_{\mathrm{B}} = 147.3$ K, and the effective surface density of adsorption sites was $\rho_{\mathrm{S}} = 15.3\ \mathrm{nm}^{-2}$. The chemical potential μ was converted to the bulk vapor pressure p using the Johnson-Zollweg-Gubbins (JZG) equation of state for LJ fluid [165]. Each system was held at the constant temperature $T = 77.4$ K of experimental measurements.

The accessible pore diameter is defined as the diameter of the pore interior that is available for adsorbed molecules. This differs from the internal diameter, used in calculation of pore diameter via the NLDFT method, and the external, or center-to-center diameter. This definition of accessible diameter provides the correct asymptotic value of the fluid density in the pore to the equation of state. The accessible diameter omits the volume excluded due to LJ repulsion between the adsorbate and the pore wall. For the model utilized, the difference in external and accessible diameters is 0.18 nm, which is the difference between the center of a hypothetical solid particle in the pore wall with the edge of nitrogen (adsorbing) particle, found when $U_{\mathrm{SF}}(r) = 0$. Therefore, the external pore sizes calculated were 5.72 nm, 7.40 nm, etc. To avoid confusion, we will only refer to the accessible pore diameters when referring to simulation, and internal diameters when referring to experimentally determined pore diameters.

Simulations were performed using grand canonical ensemble (GCMC) and gauge cell MC simulation methods. Standard algorithms [10, 166] were employed with simulation runs of at least 300,000 MC steps per molecule. Each step included an attempt of a molecule displacement in each cell or attempts of molecule removal, insertion, or transfer between the cells (where applicable). The maximum number of molecules in the largest pore was about 7,600.

The Gauge Cell method [9, 10] models equilibrium between the fluid in a pore sample system and a finite size reservoir of a limited capacity; the so-called gauge cell. The two cells of constant volume are considered in thermal and chemical equilibrium, maintained using particle exchange between them, somewhat similar to GCMC. The total number of particles, $N_{\mathrm{total}} = N_{\mathrm{pore}} + N_{\mathrm{gauge}}$, is conserved. The gauge cell MC setup is referred to as the mesocanonical ensemble [9]; it approaches the canonical ensemble when the gauge cell is infinitely small ($V_{\mathrm{gauge}} = 0$) and the grand ensemble at the infinitely large gauge cell ($V_{\mathrm{gauge}} \to \infty$). The gauge cell fulfills two purposes: its limited size restricts density fluctuations in the system, and it serves as a reference for finding the chemical potential of the pore fluid (since the condition of equilibrium between the pore and the gauge is the equality of chemical potentials). The suppression of fluctuations allows for simulations of metastable and labile states, resulting in a continuous van der Waals's type isotherm that can be thermodynamically integrated. The backwards trajectory of unstable states corresponds to critically sized bubbles, which would spontaneously evaporate in an open system. The number of simulations was increased near spinodal conditions to improve statistics. For further details of the gauge cell method, see Ref. [9, 10, 49].

The simulated isotherms were compared to the experimental isotherms to ensure the model's validity. This was achieved through fitting simulated data to two characteristic experimental points at $p/p_0 = 0.95$ (complete filling of all pores) and $p/p_0 = 0.40$ (complete filling of connecting pores prior to capillary condensation in main mesopores). The simulated isotherms corresponded to spheroidal pores of diameter equal to the mean mesopore diameter determined from the experimental adsorption isotherms. The general fit used was $N_{\mathrm{exp}}/N_{\mathrm{exp}}(0.95) = (N_{\mathrm{sim}} + \Delta N)/(N_{\mathrm{sim}}(0.95) + \Delta N)$. This fit implies that both simulation and experimental adsorption isotherms are normalized at $p/p_0 = 0.95$ to ensure the same pore volume. The value of ΔN was determined to ensure the same adsorption at $p/p_0 = 0.40$. In so doing, the gauge cell canonical isotherm was then mapped onto the corresponding experimental isotherm.

3.1.6 Experimental Results

We have studied the pore size dependence of the cavitation pressure on a series of well-characterized samples with the mean diameter of spheriodal mesopores ranging from 6.9 nm to 35 nm. Selected adsorption/desorption isotherms are presented in Figures 3.1, 3.2, and 3.3. All isotherms display a characteristic sharp step on the desorption branch, which is associated with cavitation-induced spontaneous evaporation from mesopores. The mean pore diameters and the relative pressures of cavitation are given in Table 3.1. The mean pore diameter was determined with the hybrid NLDFT method applied to the adsorption branch (section 3.1.4) [28]. Since the experimental desorption steps are never vertical due to the distribution of pore sizes and other sample non-idealities, the reported pressures of cavitation were defined in the middle of the desorption steps. Thus for a quantitative analysis, we assumed that the middle of the desorption step corresponded to the onset of cavitation in pores of the mean diameter. This assumption seems reasonable, yet it implies a some uncertainty in the reported values.

As clearly seen from Figure 3.1, there is a little difference in the positions of cavitation for samples with pores larger than ∼11 nm disregarding the sample nature. The cavitation in KLE/IL, KLE-C23, SLN-326, and SBA-16 (11.7 nm) samples occurs essentially at the same relative pressure of $p/p_0 = 0.48 \pm 0.01$. At the same time, the cavitation step for

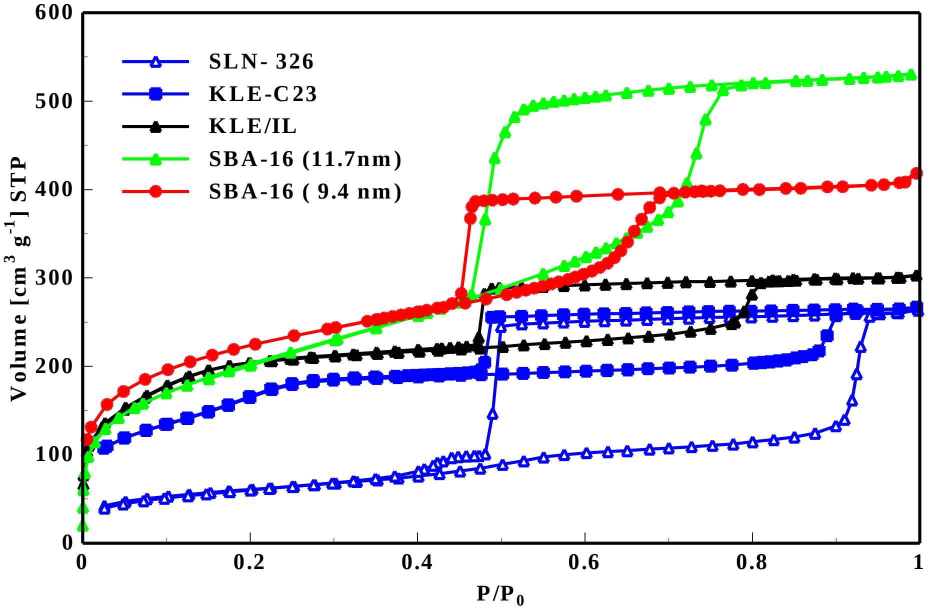


Figure 3.1: Nitrogen adsorption (at T = 77.4 K) in SBA-16 and hierarchically structured silica samples with pore main cavity diameters > 9.4 nm.

9.4 nm pore sample of SBA-16 is noticeably shifted to lower pressures. To show that this shift was not caused by the difference in the sizes of connecting pores, we plot PSDs for SBA-16 (9.4 nm), KLE-C23 and KLE/IL samples in Figure 3.2. These distributions show two groups of pores: spheroidal mesopores, in which cavitation occurs, and connecting channels, or necks. While the sizes of mesopores are significantly different, the sizes of connecting pores in all samples are essentially the same. Thus, the shift in the cavitation pressure cannot be explained by the classical pore blocking effect, which would imply the difference in the size of connecting pores.

The pore size dependence of the cavitation pressure is demonstrated in Figure 3.3 with a series of SBA-16 samples with the main mean pore diameter ranging from 6.5 nm to 10.5 nm. To make a visualization comparison easier and to an avoid overlap of desorption steps, the isotherms are presented in reduced adsorption units (reduced to the adsorption at $p/p_0 = 0.95$) and the isotherms for 7.8, 8.8 and 10.5 nm samples are shifted upwards by 0.25, 0.50, and 1.0, respectively. The unmodified isotherms are available in Supplementary Information. The isotherm on 9.4 and 11.7 nm SBA-16 samples given in Figure 3.1 fit nicely into this trend.

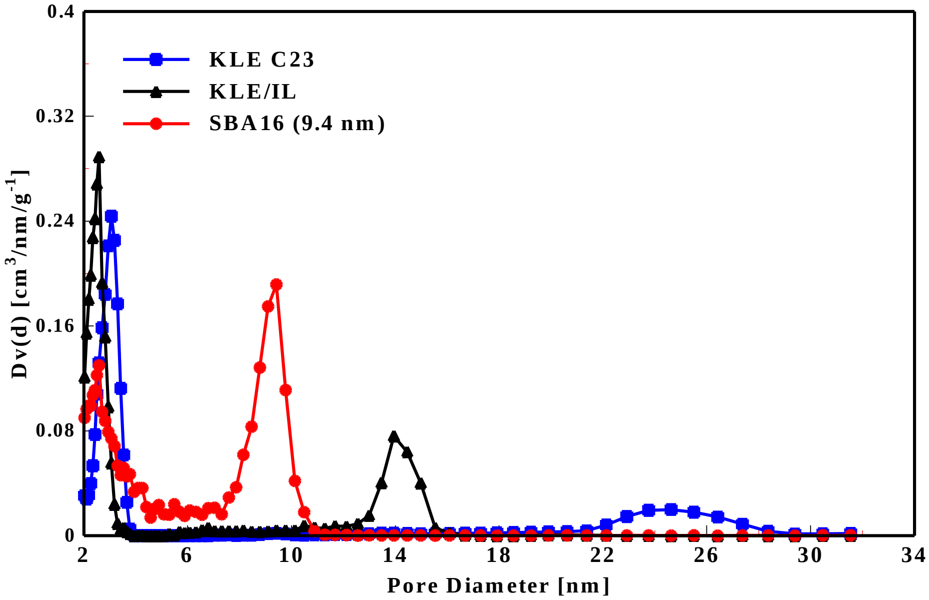


Figure 3.2: Selected NLDFT pore size distribution curves

The pore size dependence of the cavitation pressure is summarized in Figure 3.4. In addition to the data for SBA-16, KLE-type silica and SLN-326 silica, which are materials with cage-like (spheroidal) mesopores, we also added two data points reflecting cavitation in wormlike (cylindrical) pores. The data point for $\sim$5 nm pore corresponds to SLN 326 sample discussed below, and the data point for 9.4 nm pore refers to SE3030 silica that was addressed in our previous work [36]. The cavitation pressure in spheroidal pores increases almost linearly from $\sim$ 0.42 to $\sim$ 0.49 with the increase of pore diameters from $\sim$ 7 to $\sim$11 nm and then saturates at $\sim$0.49 in the range of pore diameters from $\sim$11 to 35 nm.

It is well understood that as the confinement size decreases, the liquid spinodal, which determines the limit of metastability and the conditions of cavitation, shifts to lower pressure [21, 50, 51, 167]. On the other hand, one can hypothesize that the observed saturation of the pore size dependency on the cavitation pressure observed in pores of 11 — 35 nm corresponds to the achievement of the limit of stability in metastable bulk liquid due to homogeneous nucleation. If this were true, one could offer an efficient method for studies of the limits of cavitation stability in metastable bulk liquids by measuring desorption isotherms on materials with large mesopores.

The pressure at which cavitation occurs is determined by the probability of nucleation in

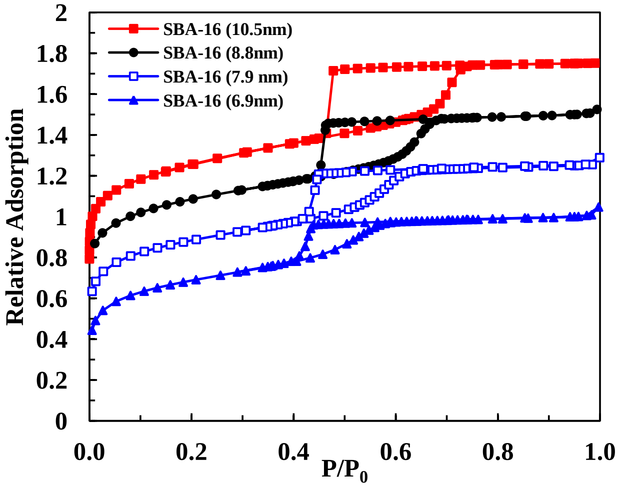


Figure 3.3: Nitrogen adsorption/desorption isotherms in SBA-16 silica materials with pore diameters < 11 nm. The shift of the cavitation step can be clearly seen. The isotherms are presented in reduced adsorption units (reduced to the adsorption at $p/p_0 = 0.95$); they are shifted upwards progressively by 0.25 to avoid overlap of desorption steps to illustrate the dependence of cavitation pressure on pore size. The original non-normalized isotherms are given in Supplementary Information.

a pore of a given size during the observation time. It is proportional to $\exp(\Delta\Omega_{\mathrm{cav}}(d)/k_{\mathrm{B}}T)$, where $\Delta\Omega_{\mathrm{cav}}(d)$ is the nucleation barrier associated with the formation of a critical cavity in a pore of diameter d [55]. The pore size dependence (Figure 3.4) suggests that the nucleation barrier should depend strongly on the pore size at $d < 11$ nm and show little or no dependence afterwards, while the fluid state in the central part of the pore gradually approaches that of the bulk liquid. We therefore may assume that in pores larger than 14 nm the cavitation barrier approximately equals that for the homogeneous nucleation in the bulk liquid nitrogen. This hypothesis is confirmed in MC simulation described below.

It is worth noting that metastable states and associated cavitation and hysteresis behaviors can be observed for nitrogen adsorption at 77.4 K in SBA-16 and other mesoporous silicas with spheroidal pores only when the pore diameter exceeds ca. 5 nm. Samples with smaller pores exhibit reversible adsorption-desorption isotherms [155, 161]. The isotherm reversibility in explained by a decrease of the nucleation barriers down to $\sim 30k_{\mathrm{B}}T$, which are easily crossed due to thermal and other fluctuations in adsorption experiments. Note that since the hysteresis behavior depends on the pore shape as well, the isotherm reversibility in cylindrical pores is limited by ~ 4 nm that corresponds to the lowest relative pressure

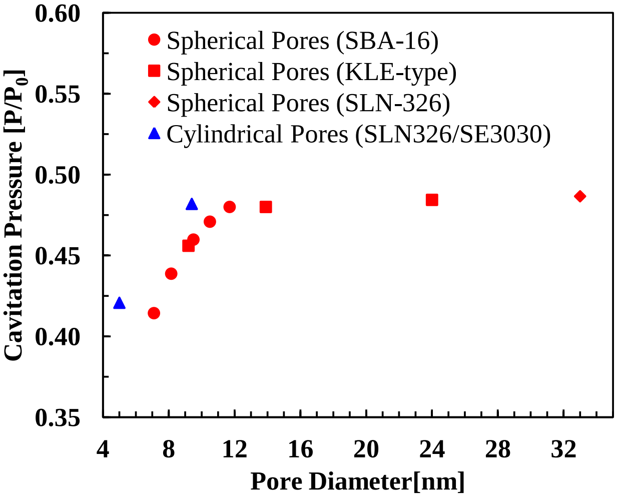


Figure 3.4: Effect of pore diameter on cavitation pressure. Experimental data for the samples with spheroidal pores are in red and with cylindrical pores in blue. Sample denotation is given in the insert. The cylindrical pore $d \approx 5$ nm refers to the wormlike channels in SLN-326, while the cylindrical pore at $d \approx 9$ nm is SE3030.

of hysteresis of $\sim$0.42 [21].

The difference in cavitation pressure in spherical and cylindrical pores can be shown with the isotherm on SLN-326 sample, which exhibits a two-step desorption branch and a pronounced inflection of the adsorption branch far below the main capillary condensation step. As seen in Figure 3.5, where we present a series of scanning desorption isotherms, the hysteresis loop can be decomposed into two non-overlapping loops, which correspond to two different groups of pores: large spheroidal pores with the mean diameter of $\sim$ 35 nm and cylindrical worm-like channels with the mean diameter of $\sim$5 nm. The existence of these two groups of pores is confirmed by the argon isotherms presented in Figure 3.6. The Ar isotherm at 87.3 K has similar features as the N_2 isotherm at 77 K. The Ar isotherm at 77.4 K has only one hysteresis loop that corresponds to the smaller pores, since at this low temperature (6.5 degrees below the argon triple point) the pores larger than $\sim$18 nm cannot be filled from the vapor phase anymore by capillary condensation [139, 168, 169].

While the cavitation mechanism of desorption from large pores is obvious, the mechanism of desorption from smaller pores requires an additional consideration. Indeed, it may be driven either by cavitation or by pore blocking. In order to detect which mechanism is dominant, we applied a test suggested in our previous work [36]. This test is based on

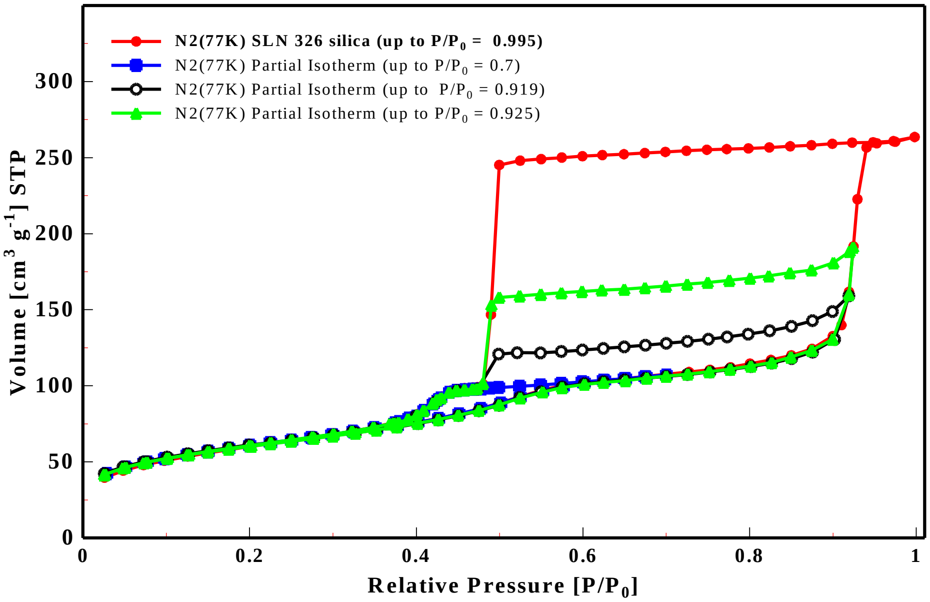


Figure 3.5: Nitrogen adsorption in SLN-326 silica at 77.4 K. Scanning isotherms demonstrate two groups of mesopores: cages of size 35 ± 3 nm and worm-like channels of 5 ± 1 nm.

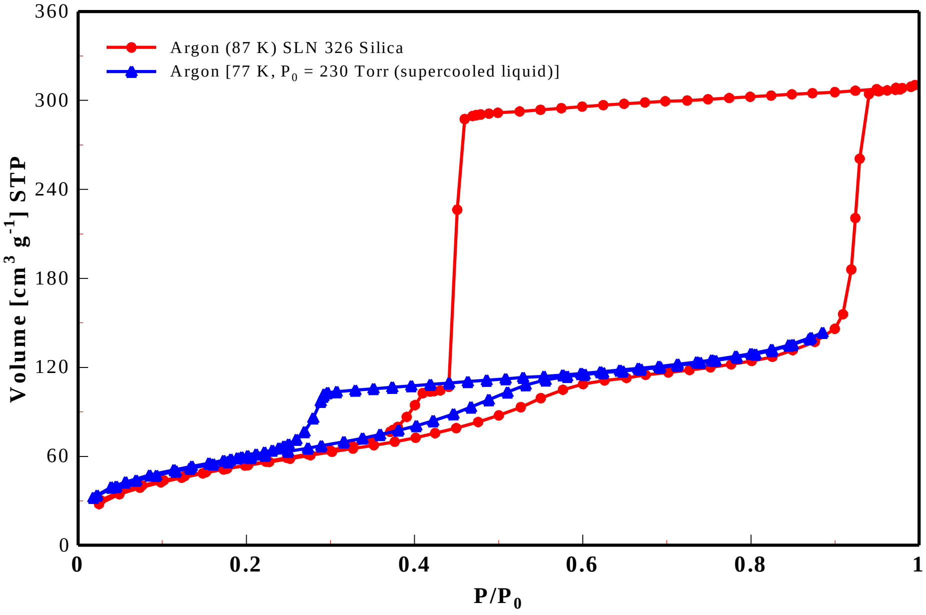


Figure 3.6: Argon adsorption at 87.3 and 77.4 K

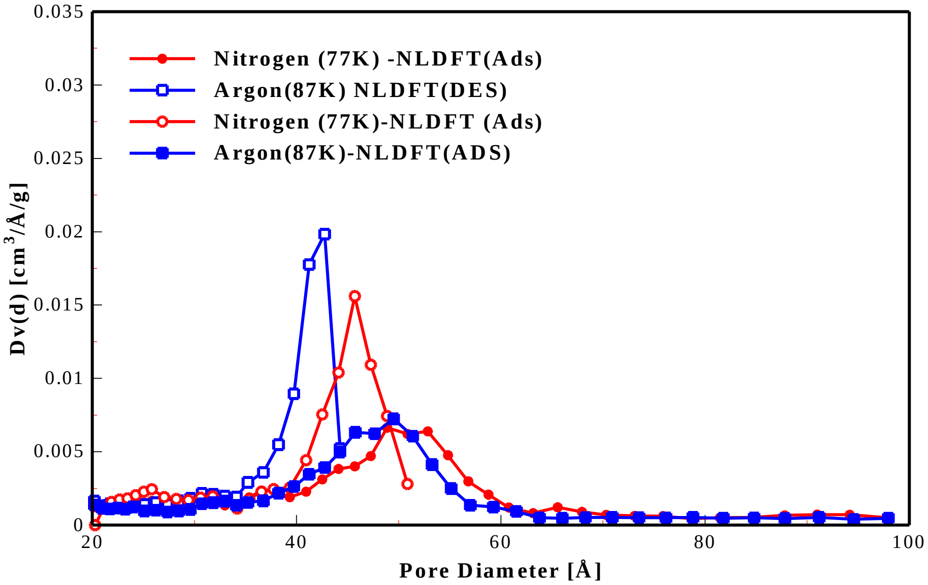


Figure 3.7: NLDFT PSDs of worm-like channels calculated from the scanning isotherms of N_2 and Ar

measuring adsorption isotherms with different adsorbates (such as nitrogen and argon here) and/or at different temperature and comparing PSDs calculated from the data obtained at these different conditions. In the case of pore blocking, the pressure of evaporation is controlled by the size of connecting pores. Therefore, PSDs calculated from the desorption branches should be independent of the choice of the adsorbate or temperature. In the case of cavitation, the pressure of desorption depends on the adsorbate and temperature and is not correlated with the size of connecting pores. Hence, PSDs calculated from the desorption branch of the hysteresis loop are artificial; they do not reflect the real pore sizes and they should depend on the choice of the adsorbate and/or temperature. This is exactly what is seen in Figure 3.7, where we present the PSDs calculated from the nitrogen at 77.4 K and argon at 87.3 K isotherms in the range from 2 to 10 nm. The PSDs calculated from the desorption branches are significantly different. At the same time, the PSDs calculated from the adsorption branches agree nicely and reflect the real sizes of connecting pores. This analysis confirms that (i) desorption from the cylindrical mesopores of SLN-326 is induced by cavitation and (ii) cavitation pressures in spherical and cylindrical pores are different.

3.1.7 Results of Monte Carlo Simulations and Comparison with Experiment

Monte Carlo simulation results for pores of different diameters (from 5.54 to 9.56 nm) are shown in Figure 3.8a. These isotherms combine the data obtained in the GCMC and gauge cell simulations as the particle density $N\sigma^3/V$ versus the bulk relative pressure, p/p_0. The isotherms determined in the gauge cell simulations are the canonical isotherms that would be obtained with the Widom particle insertion method [8] in the canonical ensemble [9]. The canonical isotherms are continuous and have a sigmoid shape resembling van der Waals's loops typical for mean field theories of first order phase transitions. It is important to recognize that the sigmoid shape comes here from the finite size of the system rather than from a mean field approximation [51]. The canonical isotherm is composed of three parts: (i) adsorption branch of low density vapor-like states that terminates at a vapor spinodal $\mathrm{S_V}$; (ii) desorption branch of high density liquid-like states that terminates at a liquid spinodal $\mathrm{S_L}$; and (iii) backward trajectory of labile states connecting the vapor and liquid spinodals; these states of negative compressibility would be totally unstable in experiment or the grand canonical ensemble. The metastable and labile states that are not achievable in the grand canonical ensemble were sampled as the mesoscopic canonical ensemble conditions of the gauge cell MC simulation [9].

The position of vapor-liquid equilibrium (vertical line BF) is determined from the condition of the condition of the nil work of the transition between equilibrium vapor-like and liquid-like states. This condition is equivalent to the Maxwell rule of equal areas (area $\mathrm{BS_VD}$ = area $\mathrm{FS_LD}$, D is the intersection of lines BF and $\mathrm{S_VS_L}$). The corresponding equilibrium chemical potential, μ_e, is found using thermodynamic integration along the canonical isotherm:

$$\oint_{\mu_\mathrm{e}} N \, \mathrm{d}\mu = 0 \tag{3.1}$$

The states on the adsorption branch at $\mu > \mu_\mathrm{e}$ and desorption branch at $\mu < \mu_\mathrm{e}$ of the canonical isotherm are metastable, and the spinodal points represent the thermodynamic limits of metastability.

GCMC adsorption-desorption isotherms form hysteresis loops like the one (ACGEA)

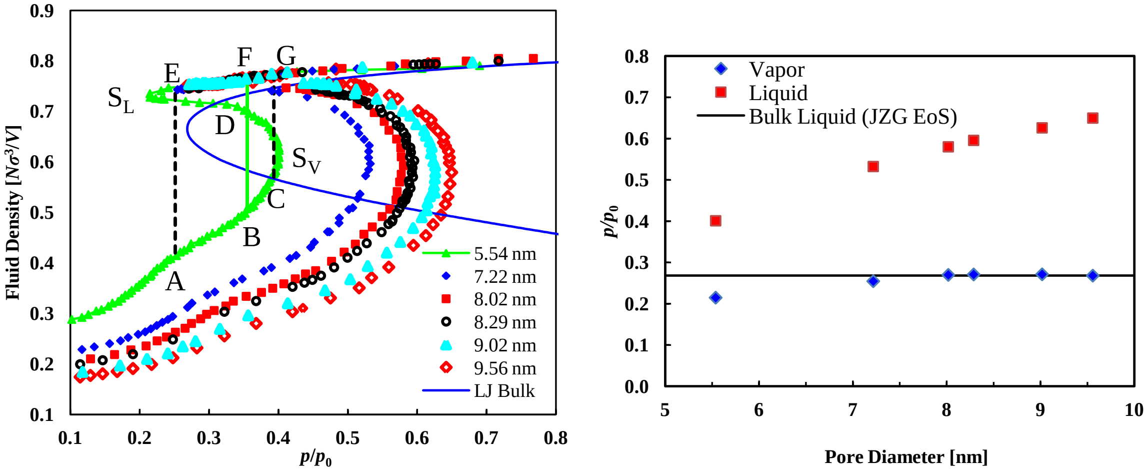


Figure 3.8: (a) Simulated nitrogen isotherms in pores of different sizes on SBA-16 silica at $T = 77.4$ K. Vertical line EA corresponds to cavitation in a metastable liquid state in GCMC simulation. Line FB represents vapor-liquid equilibrium. Line CG represents capillary condensation in GCMC. $\mathrm{S_L}$ and $\mathrm{S_V}$ are the limits of stability in the liquid and vapor states, as calculated by the gauge cell method. (b) Summary of calculated spinodal points, (■ red squares: vapor spinodals, ◆ blue diamonds: liquid spinodals). In systems $d > 8$ nm, nearly identical values of spinodals are calculated, corresponding closely to the JZG equation prediction of the bulk value of $p/p_0 = 0.268$. This value corresponds to the horizontal line.

shown on the 5.54 nm isotherm. The abrupt steps CG and EA correspond to the spontaneous capillary condensation and desorption/cavitation transitions observed in the GCMC simulation, below and above the respective spinodals. Virtual experiments of GCMC simulations mimic experimental conditions of real adsorption measurements: the sample is set in equilibrium with a macroscopically large (thermodynamically infinite) reservoir of vapor kept at given vapor pressure and temperature. However, the positions of spontaneous capillary condensation and desorption in simulations may differ significantly from those observed experimentally. The spontaneous transition in a metastable phase is a fluctuation driven process that requires formation of a critical nucleus. The probability of nucleation at given conditions depends on the level of natural thermal fluctuations and the observation time, which in real experiments is significantly larger than in simulations. Thus, spontaneous transitions in GCMC simulations occurs closer to the spinodals that in real experiments.

Naturally, the locations of VLE and both spinodals shifts to higher pressures as pore width increases; however, while the vapor spinodal continues to increase steadily with the pore diameter, the liquid spinodal reaches a plateau of $p/p_0 \approx 0.270$ (see Figure 3.8b). This supports our hypothesis that cavitation in large pores occurs at conditions near that of a bulk fluid. The JZG equation predicts a nearly identical bulk liquid spinodal of $p/p_0 = 0.268$. However, this comparison must be treated with caution as this equation was not explicitly designed to calculate spinodals [165]. Also noteworthy is that for the condensation transition the situation is qulitatively different. The vapor spinodal pressure monotonically increases with the pore size but the fluid always condenses at pressures lower than the bulk VLE pressure p_0. The condensation transition has no relevance to the bulk vapor spinodal, which occurs at oversaturated conditions (at $p/p_0 = 6.94$, according to the JZG equation).

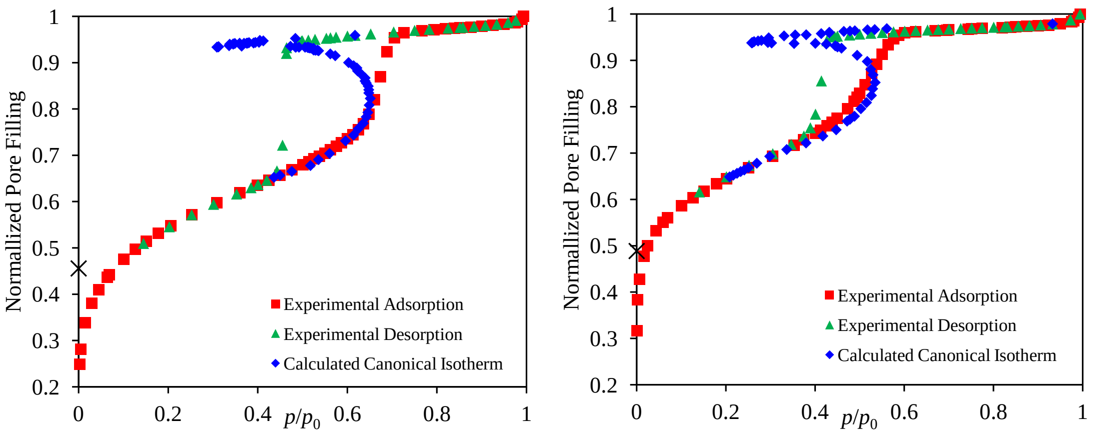


Figure 3.9: Comparison of MC simulations and experiments. (a) 9.4 nm SBA16-100C (exp) and 9.56 nm gauge cell (sim), and (b) 7.1 nm SBA16-60C (exp) and 7.22 nm gauge cell (sim). Isotherms are fitted to 2 points, one near saturation and another in the linear region of the vapor-like branch. The calculated isotherm is shifted up to account for adsorption in connecting and matrix pores. This shift is indicated by × on the y-axis. Note that the liquid spinodal is far removed from the point of experimental desorption (cavitation).

The MC simulated isotherms are generally in good agreement with the experimental isotherms given in Figures 3.1 and 3.3. In Figure 3.9, we present two typical examples for SBA-16 samples, where the experimental isotherm is compared with the simulated canonical isotherm in the spherical pore of the mean diameter for the given sample. To display simulated data as functions of the vapor pressure the JZG equation of state was employed to relate the vapor pressure and the respective chemical potential. One cannot expect to get the exact agreement since the samples contain mesopores of different sizes (see pore size distributions in Figure 3.2) that is not captured in single-pore simulations. Most importantly, the samples contain additional micropores that are filled at pressures below the condensation region for the main spherical mesopores. For the purpose of comparison, the experimental and simulated isotherms were normalized at $p/p_0 = 0.95$, and in order to account for real adsorption of the micropores, the simulated isotherm was shifted up to fit the experimental isotherm prior to the region of capillary condensation in mesopores. The quantitative value of this shift is displayed as a crosshair on the y-axis of Figure 3.9.

The simulated and experimental isotherms overlay nicely, indicating that the simulations correctly describe the physics of adsorption/desorption processes. The compressibility of the condensed fluid (the slope of the desorption isotherms prior to cavitation) aligns in experiments and simulations, as well as the slopes of the adsorption isotherms prior to the capillary condensation. As expected, capillary condensation in experiments occurred near the position of the vapor spinodal $\mathrm{S_V}$. At the same time, cavitation in experiments occurred prior to the achievement of the liquid spinodal $\mathrm{S_L}$, and the difference in the respective relative pressures is significant.

To get a better understanding on the conditions of the cavitation onset in experiments, we calculated the nucleation barriers in metastable condensed fluid from the canonical isotherms. The continuity of the canonical isotherm allows one to employ the thermodynamic integration to calculate the grand thermodynamic potential, $\Omega(\mu, T)$,

$$\Omega(\mu, T) - \Omega(\mu_r, T) = -\int_{\mu_r}^{\mu} N(\mu, T)\,\mathrm{d}\mu \tag{3.2}$$

where μ_r is an ideal reference state with sufficiently low pressure. The states on the backward branch between the vapor spinodal and the vapor-liquid equilibrium correspond to the critical nuclei (cavities, or bubbles) that should be formed to trigger cavitation in the metastable condensed liquid [9, 55]. The nucleation barrier at a given chemical potential $\mu < \mu_\mathrm{e}$ represents the work of formation of the critical nucleus. The latter equals to the difference of the grand thermodynamic potentials of respective liquid and "bubble" states, $\Delta\Omega_\mathrm{cav} = W_\mathrm{c} = \Omega_\mathrm{l}(\mu, V, T) - \Omega_\mathrm{b}(\mu, V, T)$. Here, Ω_l and Ω_b are the grand thermodynamic potentials along the liquid and backward branches of the canonical isotherm. This is graphically represented by the area from $\mathrm{S_L}$ to F (or any other desirable μ), between the unstable and liquid branches in Figure 3.8a.

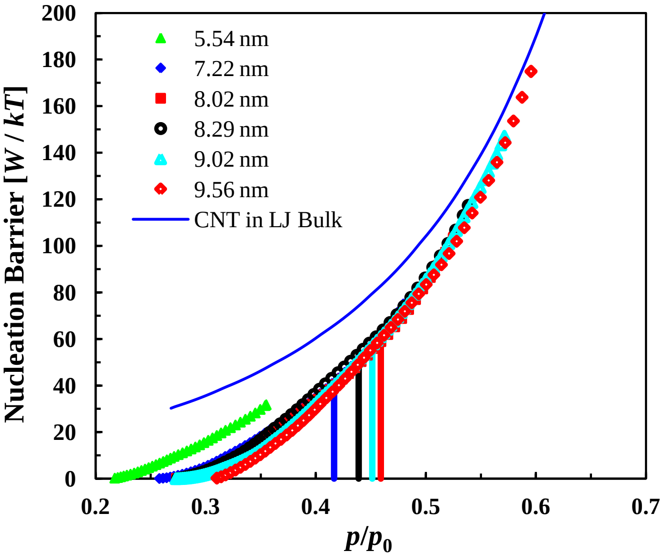


Figure 3.10: Calculated nucleation barriers of nitrogen at 77.4 K confined to spherical silica pores of different sizes. The solid blue curve corresponds to the classical nucleation theory estimate of the barrier. Solid vertical lines show the positions of cavitation observed experimentally (See Figure 3.4 and Table 3.1). The calculated curves are terminated at the respective pressures of equilibrium vapor-liquid transition.

The nucleation barrier of cavitation W_c as a function of the vapor pressure is plotted in Figure 3.10 for different pore sizes. The nucleation barrier monotonically increases from the liquid spinodal (where it is zero by definition) to the vapor-liquid equilibrium, where the nucleation barrier of evaporation reaches a maximum equaled to the nucleation barrier of condensation defined as $\Delta\Omega_\mathrm{cond} = \Omega_\mathrm{v}(\mu, V, T) - \Omega_\mathrm{b}(\mu, V, T)$ (due to the Maxwell rule). The nucleation barriers for condensation are shown by dotted lines. Most remarkably, the nucleation barriers in pores larger than 8 nm at relative pressures larger than $\sim$0.5 collapse on one universal dependence. This finding suggests that the conditions of bubble nucleation in sufficiently large mesopores does not depend on the size of confinement, and thus, are characteristic to homogeneous nucleation in the bulk liquid.

For comparison, we also calculated the nucleation barriers using the classical nucleation theory (CNT). According to CNT [170], the work of critical bubble formation equals

$$W_\mathrm{c} = \frac{4\pi}{3} r_\mathrm{s}^2 \gamma_\mathrm{s} \tag{3.3}$$

where γ_s is the surface tension of a bubble with radius r_s. The radius of a critical nucleus is then approximated using the Laplace equation,

$$r_\mathrm{s} = \frac{2\gamma_\mathrm{s}}{p_\mathrm{l} - p_\mathrm{v}} \tag{3.4}$$

where p_l and p_v are the pressures of bulk liquid and vapor corresponding to the same chemical potential μ via the bulk equation of state, and also assuming the ideal vapor phase and non-compressible liquid phase. The surface tension of the LJ model of nitrogen was estimated from simulation data reported in the literature [171]. A value of $\gamma\sigma^2/\epsilon = 0.922$ was obtained by interpolation using $k_\mathrm{B}T/\epsilon = 0.762$ and $r_c/\sigma = 5$. CNT has several severe drawbacks, namely the assumption that cavities are spherical and that the fluid outside the cavities represents a uniform liquid phase [172]. Also, CNT does not predict nil nucleation barriers at the spinodal. However, CNT is expected to provide correct asymptotic values as the bubble size increases. The CNT result for the LJ model of nitrogen is displayed as a solid line in Figure 3.10. As expected, CNT largely overestimates the nucleation barriers in nanometer-sized pores in the practical region of cavitation pressures ($p/p_0 < 0.5$). However, the MC calculated barrier of the largest pore of 9.56 nm asymptotically approaches the CNT value at high relative pressures. This observation suggests that the latter dependence can be employed for predicting nucleation barriers in larger pores and making extrapolations for the bulk liquid.

For the smallest of the simulated pores (5.54 nm), experimental desorption (HMM-3 sample) occurred at the vapor-liquid equilibrium without any hysteresis. The corresponding nucleation barrier is only 32 $k_\mathrm{B}T$, which is easily crossed by thermal fluctuations at the conditions of standard adsorption experiments. As seen in Figure 3.10, the nucleation barrier at the equilibrium sharply increases with the pore size from ~ 75 $k_\mathrm{B}T$ in 7.22 nm pore to ~ 175 $k_\mathrm{B}T$ in 9.56 nm pore. These barriers are insurmountable during the experiments, and thus the adsorption-desorption isotherms exhibit prominent hysteresis. The position of desorption in these samples is determined by the conditions of cavitation.

From the comparison of simulated and experimental data, we evaluated the nucleation barriers that corresponded to the experimentally observed cavitation in pores of difference sizes. The nucleation barriers were determined at the experimental cavitation pressures

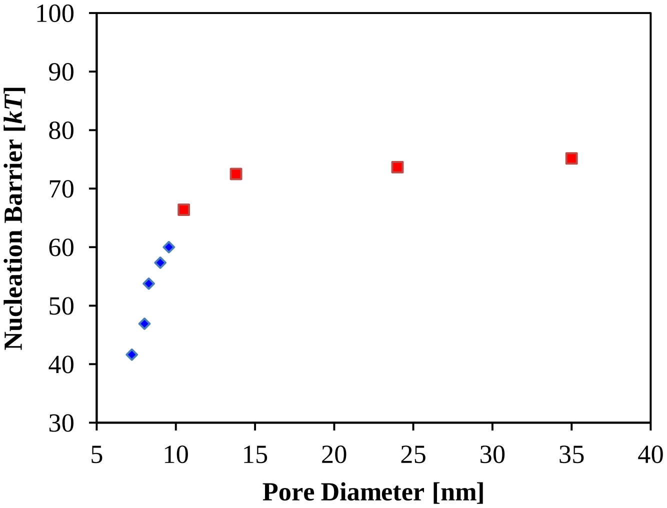


Figure 3.11: The pore size dependance of the nucleation barrier of cavitation. Nucleation barriers in large pores that cannot be adequately sampled with Monte Carlo simulations were extrapolated from the nucleation barrier relationship for a 9.56 nm pore (see Figure 3.10). They are shown by red squares.

reported in Table 3.1 and shown by solid vertical lines in Figure 3.10. Cavitation pressures were adjusted slightly by interpolation to correct for the minor differences between experimentally measured NLDFT diameters and the internal diameters used in simulation. The pore size dependence of the nucleation barriers is given in Figure 3.11. The nucleation barrier is plotted against the mean diameter of main mesopores of the samples explored experimentally. For pores $d > 10$ nm, the nucleation barrier was obtained from the results of simulation in 9.56 nm pore.

As the pore size increases, so does the nucleation barrier of cavitation from $\sim 40\ k_{\mathrm{B}}T$ at 7 nm to 70 $k_{\mathrm{B}}T$ at 11 nm. In larger pores up to 35 nm, the nucleation barrier varies insignificantly in the diapason of $70 - 75\ k_{\mathrm{B}}T$ that corresponds to minor variations of the cavitation pressure in large pore samples (Figure 3.5). It is worth noting that nucleation barriers of $\sim 70\ k_{\mathrm{B}}T$ correspond to the largest degrees of metastability achievable in macroscopic experiments with overheated cryogenic liquids like nitrogen and argon prior to their spontaneous boiling [173]. This estimate suggests that (i) the results of simulations of nucleation barriers for confined fluids can be extrapolated to the bulk systems, and (ii) the conditions of cavitation in large mesopores are similar to the conditions of homogeneous cavitation in bulk metastable liquids.

3.1.8 Discussion and Conclusions

By means of high-resolution adsorption-desorption experiments with specially prepared mesoporous silica materials, we studied cavitation in metastable liquid nitrogen at its boiling temperature of 77.4 K. Complementary experiments were performed with argon at 77.4 K and 87.3 K (argon boiling temperature). All samples possessed 3D pore networks with uniform cage-like spheroidal mesopores connected by more narrow channels. Two classes of samples were explored: ordered mesoporous crystals SBA-16 with the $Im\overline{3}m$ cubic pore network and hierarchically structured silicas of KLE, KLE/IL, and SLN-326 types. The samples were characterized by the mean internal diameter of spheroidal mesopores, which varied from sample to sample in the range $6 - 35$ nm. The pore size distributions were determined by the NLDFT hybrid method [36].

The cavitation mechanism of evaporation of condensed nitrogen confined to cage-like pores is displayed by a sharp step on the desorption isotherm. This mechanism was confirmed by the independence of the desorption step position of the size of connecting pores, as well as by a comparative analysis of adsorption-desorption isotherms of nitrogen and argon at the normal boiling temperatures of nitrogen and argon. It is well-understood from earlier studies that the onset of cavitation is related to the achievement of the limit of stress stability in the metastable condensed liquid. The so-called limiting stress hypothesis was suggested in the seminal works of Everett [174] and Dubinin [175] to explain the position of the lower closure point of the hysteresis loop formed by the adsorption and desorption isotherms. However, it is generally assumed that the cavitation pressure depends on the adsorbate and is independent of structural and chemical properties of the porous solid [20]. In contrast to this classical viewpoint, our data shows that the vapor pressure at the onset of cavitation does depend on the pore size for the samples with pores smaller than $\sim$11 nm and remains practically unchanged for the samples with larger pores.

We have found the difference in the cavitation pressure in spherical and cylindrical pores of the same diameter. This conclusion was confirmed by a detailed analysis of the characteristic hysteretic behavior observed in SE3030 silica, which consists of wormlike pores of diameter 9.4 nm, and SLN-326 silica, which contains extra-large cage-like pores with the mean diameter of 35 nm and worm-like cylindrical channels of ~ 5 nm. By measuring scanning desorption isotherms and, in addition, Ar adsorption at 87.3 K and 77.4 K, we showed that two characteristic steps on the desorption branch corresponded to two independent cavitation processes in spheroidal and cylindrical pores. Within this context, it is worth restating that cavitation occurs in the materials containing mesopores accessible only through narrow channels (necks/windows smaller than ~ 4 nm) that are remained filled at the pressure characteristic to the onset of cavitation, which depends on the mesopore shape and size.

The onset of cavitation is preceded by the formation of a critical nucleus of the vapor phase in the metastable liquid. Thus, the position of cavitation is determined by the nucleation barrier, defined as the critical nucleus free energy, which depends on the degree of metastability. We suggest that the observed independence of the cavitation pressure of the size of confinement indicates that the conditions of bubble nucleation in pores larger than $\sim$11 nm approach the nucleation conditions in the bulk liquid. In smaller pores, the nucleation barrier depends on the pore size and shape: the smaller the pore, the smaller the barrier, and thus the vapor pressure of cavitation increases with the pore size, from $\sim$0.42 to $\sim$0.49 (for spheroidal mesopores). According to this consideration, the absence of experimentally observed hysteresis in cage-like pores smaller than $\sim$6 nm is explained by low nucleation barriers that can be overcome at the conditions of vapor-liquid equilibrium.

To test this hypothesis and to evaluate the nucleation barriers, we performed grand canonical Monte Carlo and gauge cell Monte Carlo simulations of nitrogen adsorption and desorption in spherical silica pores ranging from 5.54 nm to 9.56 nm in diameter. Simulated and experimental adsorption isotherms were in good agreement. Exploiting the correlation between the experimental cavitation pressure and the simulated nucleation barrier, we found that the nucleation barrier increased almost linearly from ~ 40 to ~ 70 $k_{\mathrm{B}}T$ in the range of pores from ~ 7 to ~ 11 nm, and varied in narrow diapason of $70 - 75$ $k_{\mathrm{B}}T$ in larger pores (up to 35 nm). The nucleation barriers of this magnitude correspond to the limit of sustainable metastability and the onset of spontaneous boiling in macroscopic metastable nitrogen droplets [173]. This coincidence confirms that the conditions of cavitation in pores larger than 11 nm are similar to the conditions of homogeneous nucleation in the bulk.

The conclusion that cavitation in sufficiently large pores does not depend on the size of confinement is supported by calculated density profiles of condensed nitrogen (see Supplementary Information). As the pore size increases, the density in the pore center approaches the density of metastable liquid nitrogen, and the size of critical nucleus becomes progressively smaller than the pore diameter; respectively, the stabilizing effect of fluid-solid attractive interactions diminishes. An additional confirmation comes from the fact that the constructed dependence of the nucleation barrier on the vapor pressure asymptotically approaches the predictions of the classical nucleation theory in the bulk at the relative pressures larger ~ 0.6.

The established correlation between the nucleation barrier at the onset of cavitation and the pore size gives a rough quantitative estimate for the criterion of adsorption hysteresis. The nucleation barrier of ~ 40 $k_{\mathrm{B}}T$, which corresponds to the smallest pore in which we observed adsorption hysteresis (~ 7 nm), can be regarded as a boundary of sustainable metastability in condensed nitrogen at the standard conditions of adsorption measurements. The absence of hysteresis in smaller pores suggests that nucleation barriers smaller that ~ 40 $k_{\mathrm{B}}T$ are overcome due to natural thermal fluctuations and temperature and pressure variations during the equilibration time set by the experimental protocol. It is worth underscoring that the conditions of sustainable metastability in experiments and simulations are significantly distinct. Nucleation barriers insurmountable in simulations are significantly smaller due to the smaller observation times and the lack of natural variations of external thermodynamic parameters inherent to real experiments [50].

In conclusion, our study provides new insights into the origin of adsorption-desorption hysteresis in nanoporous materials, which is crucial for a correct and comprehensive structural characterization of advanced nanomaterials with hierarchical pore structure. The established relationship between the cavitation pressure and the pore size may be instrumental in discriminating and validating pore network morphologies and pore shapes. Our findings imply that there is a limit to the influence of the confinement on the onset of cavitation, and thus, cavitation of nanoconfined fluids may be employed to explore cavitation in macroscopic systems. This is an important outcome, which may give rise to novel experimental techniques for measuring limiting tensile stresses, nucleation barriers, and conditions of cavitation in metastable liquids.

Supplementary Information

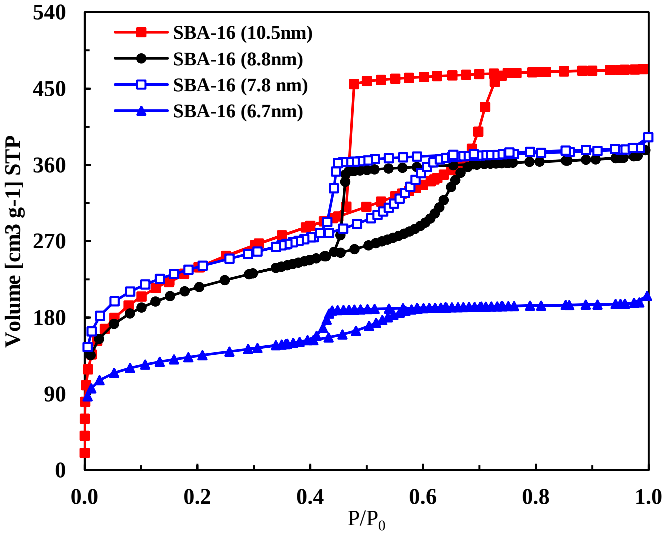
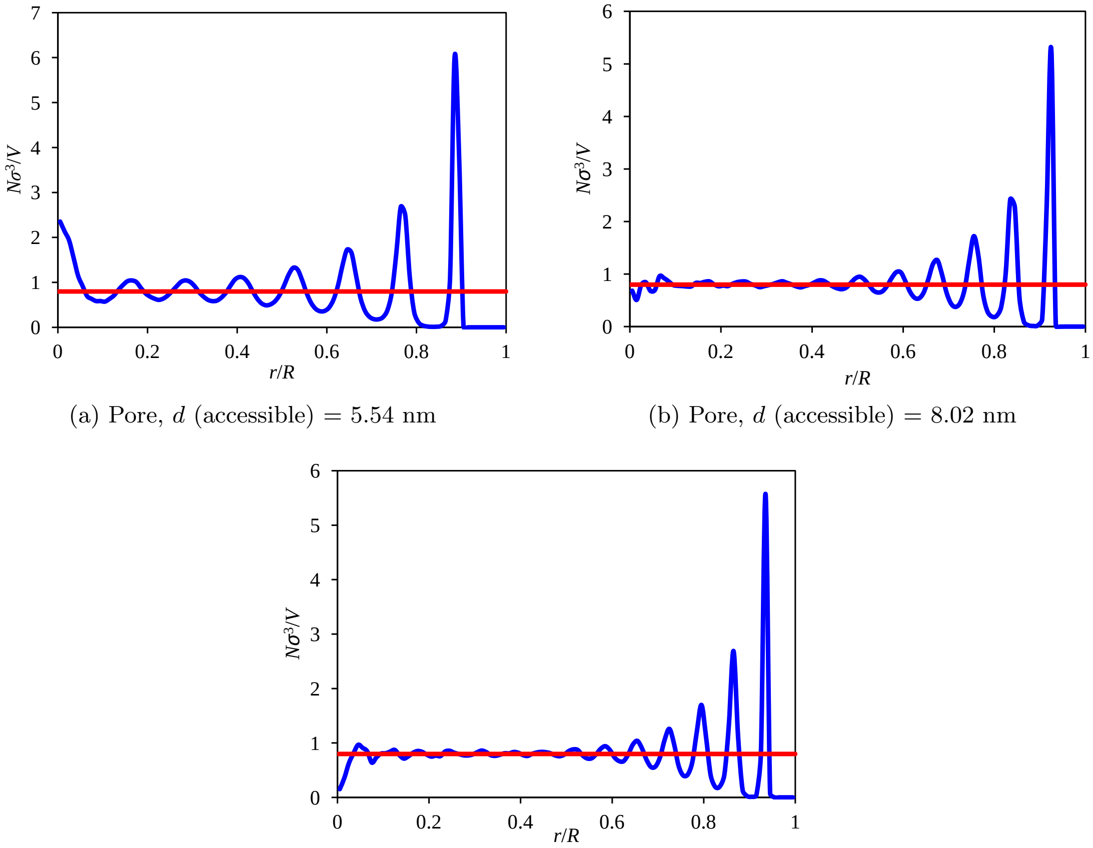


Figure 3.12: Nitrogen adsorption/desorption isotherms in SBA-16 silica materials with pore diameters < 11 nm. These data were presented in Figure 3.3 with scaled and shifted adsorption coordinates. Notice the cavitation pressure (taken at the mid-point of the desorption transition) decreases with mean pore size.

3.2 Monte Carlo Simulation of Cavitation in Pores with Nonwetting Defects

3.2.1 Introduction

The role of cavitation during the evaporation of a fluid from porous materials has been a much discussed topic [1, 35, 36, 176–181]. When the formation of an equilibrium meniscus at the vapor-liquid interface is somehow hindered, desorption occurs at a pressure less than the vapor-liquid equilibrium (VLE) pressure for the pore of a given size. In particular, there is much interest in so-called “ink-bottle” mesopores, where a pore body is connected to other pores by smaller channels or necks. As known from the experiments and respective theories, the neck diameter has a profound effect on the position of the evaporation pressure [35]. The smaller is the neck diameter, the smaller is the relative pressure of desorption



(a) Pore, d (accessible) = 5.54 nm

(b) Pore, d (accessible) = 8.02 nm

(c) Pore, d (accessible) = 9.56 nm

Figure 3.13: Density profiles from selected simulations. Densities measured at $\mu/\epsilon_{\text{FF}} = -9.5$ ($p/p_0 = 0.821$) and correspond to a liquid-filled pore. The red horizontal line is the value of the bulk density at the same chemical potential (pressure). Values were sampled equiradially, so small radii have larger statistical error. Note that $R = d$ (external)/2, where d (external) = d (accessible) + 0.18 nm. Sharp layering appears close to the adsorbing wall. In small pores, the oscillating layers are still present at considerable amplitude in the pore center. Larger pores exhibit a fluid of near bulk density at their centers.

p_{des}/p_0, (p_0 is the vapor pressure of the bulk). This is referred to as the pore blocking regime of desorption. However, below a certain critical neck diameter, which is about 4–5 nm in the case of nitrogen adsorption at 77.4 K, the experimental relative desorption pressure no longer depends on the neck size and varies in a narrow interval ($p_{\mathrm{des}}/p_0 = 0.47 \pm 0.03$) [1]. This regime of desorption is related to the cavitation of metastable fluid in the pore body. As the external vapor pressure is lowered, the fluid in the pore body becomes over-stretched (metastable), and eventually, unstable. As the limit of fluid stability is approached (near the spinodal point), a critically sized bubble nucleus forms, which grows rapidly causing spontaneous desorption of the fluid from the pore body, despite the pore neck remaining filled.

Although the importance of the cavitation phenomenon in the process of capillary evaporation was well understood [182–185], only during the last decade has it been studied extensively, both experimentally [1, 35, 36, 152, 155, 176, 177, 181, 186–188] and with computer simulation [1, 52, 55, 178, 180, 189–193]. Sarkisov and Monson first implemented grand canonical molecular dynamics in a slit-shaped pore geometry and observed emptying of the main pore body while the connecting necks remained filled [192, 193]. Quantitative experimental studies of cavitation [35] became possible with the advent of highly-ordered, templated mesoporous silicas [25]. These are designer materials with a narrow, uniform distribution of pore sizes that can be tailored to desired sizes [155]. These materials provide an opportunity for direct comparison of the results of theoretical and simulation approaches to experimental data. A comprehensive study of cavitation in ordered 3D cage-like structures of SBA-16 and FDU-1 materials was performed by Ravikovich and Neimark [35], who combined the adsorption measurements with non-local density functional theory (DFT) calculations to determine the pore dimensions. They found that the transition from the cavitation regime of desorption to the pore blocking regime occurs not only with the increase of the pore neck size, but also with a temperature decrease for a given pore geometry. With similar methods, Thommes et al. [36] explored cavitation on mesoporous silicas of hierarchical structure. Morishige et al. [181] also observed desorption pressures near $p/p_0 = 0.47$ for nitrogen in silica ink-bottle pores, and found that increasing temperatures allows for cavitation in pores with larger connecting necks. Vishnyakov and Neimark [52] studied the transition from pore blocking to cavitation with a decrease of the neck size using Monte Carlo (MC) simulations. Libby and Monson [191] modeled desorption from ink-bottle pores with lattice DFT and MC and confirmed that the transition from the pore blocking to cavitation regimes depends on the pore geometry and temperature. Rasmussen et al. [1] studied the effect of the pore body size on the onset of cavitation. Based on MC simulations performed for a variety of experimentally studied systems, they found that the cavitation pressure increases with the increase of the pore body size from 7 to 10 nm, but in a very narrow range of relative pressures from ∼0.45 to ∼0.5. In pores larger than ∼10 nm the cavitation pressure did not depend on the pore size and the cavitation occurs similarly to that in the bulk fluid. As such, it was concluded that the cavitation of nitrogen at its boiling temperature cannot occur at $p/p_0 > 0.50$–0.52. More recent molecular simulation studies of cavitation [178, 180, 191, 194], as well as multiple experimental studies [36, 152, 176, 187, 188], confirm the main features of this phenomenon described above.

The current work has been triggered by a discussion in the literature about the mechanism of desorption of fluids confined to porous silicon (PSi), a material with non-intersecting hexagonally ordered unidirectional pore channels. In PSi samples prepared and studied by different groups, a sharp desorption step is observed at relative pressures significantly larger that the range of cavitation pressures in other mesoporous materials, e.g. at $p_{\mathrm{des}}/p_0 > 0.6$–$0.7$, for the case of nitrogen [177, 195–197]. At the same time, this pressure is smaller than the equilibrium vapor pressure characteristic for the pore channels of given size that would be expected of the desorption pressure in open-ended cylindrical channels [28]. Moreover, with experiments performed with specially designed, deposited and layered PSi samples, the authors of refs [177, 196, 197] did not find sizable variations of the desorption pressure in the open-ended channels, channels closed at the bottom, and ink-bottled channels thus posing a question about the validity of the basic concepts of the mechanisms of capillary condensation hysteresis accepted in the literature [18, 19]. One explanation for these phenomena is a corrugation of pore channels leading to pore blocking effects [196]. This was confirmed by computational studies by modeling linear pores with mesoscopic roughness [195, 198]. In addition, chemical smoothing of PSi channels appears to shift the desorption mechanism towards near-equilibrium desorption, suggesting roughness is responsible for characteristic desorption pressure [199]. However, Grosman and Ortega [177], as well as Naumov et al. [195, 198], suggested that desorption in PSi channels may exhibit cavitation. The main obstacle for accepting cavitation as the primary mechanism of desorption in PSi is the conclusion derived in our previous paper [1] that the cavitation cannot occur at such large relative pressures. However, this conclusion entails that the pore walls are wetting, and as such, the cavitation occurs in a homogeneous fashion, with the critical bubble more likely formed at the pore center rather than at the solid surface. To support the cavitation hypothesis, Grosman and Ortega [177] suggested that the cavitation of metastable nitrogen in PSi channels occurs at some heterogeneities on the pore walls, and it is a heterogeneous, rather than homogeneous, process, which due to smaller nucleation barriers may take place at respectively larger vapor pressures.

In this work, as a continuation of our previous studies of homogeneous cavitation [1], we evaluated the influence of a heterogeneity of the pore wall on the vapor pressure at which cavitation occurs. We considered the heterogeneous defects on the pore wall as non-wetting spots of molecular size. We applied two approaches for quantitative estimates, the macroscopic classical nucleation theory (CNT) and detailed MC simulation in grand canonical (GCMC) and mesocanonical (MCMC) ensembles. As an instructive example, we model the behavior of a Lennard-Jones (LJ) fluid in a spherical pore with wetting LJ walls, which contained a single circular non-wetting defect. To provide quantitative relevance to experimental studies and earlier simulations, the potential parameters for the LJ model of nitrogen adsorption on silica at nitrogen's normal boiling temperature of 77.36 K employed in previous work were used here [163, 164]. We found that as the size of the non-wetting defect increases, the mechanism of cavitation changes from homogeneous to heterogeneous. Since it is doubtful that defects, which would be non-wetting to condensed nitrogen, exist in PSi, the results of this work cannot be immediately applied to the PSi desorption pressure enigma discussed above. However, the phenomenon of heterogeneous cavitation cannot be ruled out for other systems with polar fluids, such as water.

3.2.2 Heterogeneous vs Homogeneous – Classical Nucleation Theory Approach

Cavitation is necessarily preceded by the formation of a critically sized bubble, or nucleus. Bubbles smaller than the critical size collapse back into the fluid state. Once a critically sized bubble nucleates, thermal fluctuations can increase the bubble's size to a point where it grows irreversibly and displaces the adsorbed fluid; this is desorption by cavitation. CNT [200, 201] describes the thermodynamics of formation of a nucleus in the bulk system. Since CNT operates with macroscopic values, it cannot give reliable quantitative predictions when applied in the nanoscale [202]. However, CNT allows one to obtain a transparent qualitative description of the nucleation phenomenon. We use CNT to estimate the work of bubble formation for two different nucleation events: homogeneous nucleation—when the bubble forms in presumably uniform fluid inside the pore and thus does not depend on the pore size and surface chemistry, and heterogeneous nucleation—when the bubble nucleates on the pore wall and does depend on the surface defects.

According to CNT, the work of homogeneous nucleation of a critical vapor bubble of spherical shape is determined by the Gibbs equation, as the difference between the work of surface formation and the work of compression,

$$W_{\mathrm{homo}}(r_{\mathrm{b}}) = 4\pi r_{\mathrm{b}}^2\gamma - \frac{4}{3}\pi r_{\mathrm{b}}^3\Delta p, \tag{3.5}$$

where γ is the liquid-vapor surface tension, and r_{b} is the radius of critical bubble, and Δp is the capillary pressure, or the pressure difference between the vapor inside the bubble and the liquid outside the bubble,

$$\Delta p = p_{\mathrm{l}} - p_{\mathrm{v}} = \frac{2\gamma}{r_{\mathrm{b}}}, \tag{3.6}$$

where p_{v} and p_{l} are the pressures of equilibrium liquid and vapor phases determined for a given chemical potential through a bulk equation of state. A standard assumption of ideal vapor and incompressible liquid is usually used to relate the capillary pressure and, respectively, the radius of the critical bubble, to the relative vapor pressure, p/p_0, through the Kelvin-Laplace equation,

$$\frac{2\gamma}{r_\mathrm{b}} = -\frac{R_\mathrm{G}T}{V_\mathrm{m}} \ln \frac{p}{p_0}, \tag{3.7}$$

where R_G is the gas constant and V_m is the molar volume of the liquid adsorbate. In our previous work, we found that eq (3.5) correctly predicts the asymptotic value of the nucleation barrier calculated by MC simulation for a LJ fluid confined to spherical nanopores as the pore size increases, but it becomes progressively inaccurate for pores smaller that 10 nm [1]. For the convenience of dimensionless analysis, all scales can be reduced to the fluid molecular diameter σ, and eq (3.7) can be re-written as

$$\frac{r_\mathrm{b}}{\sigma} = \frac{\chi_\mathrm{c}}{\ln{(p/p_0)}}, \tag{3.8}$$

where $\chi_\mathrm{c} = 2\gamma V_\mathrm{m}/R_\mathrm{G}T$ is the dimensionless parameter that relates the theoretical nucleus size, $a_\mathrm{c} = 2\gamma\sigma V_\mathrm{m}/R_\mathrm{G}T$, at a characteristic pressure $p/p_0 = 1/e = 0.37$, to the molecular diameter σ. For nitrogen at its normal boiling temperature $T = 77.4$ K (the experimental standard in adsorption measurements), $\sigma = 0.36$ nm, $a_\mathrm{c} = 0.96$ nm, and $\chi_\mathrm{c} = 2.7$. These numbers give the reader the characteristic scale of the phenomenon we are concern with. It is worth noting that we use the bulk values of surface tension γ without Tolman-type corrections [203] for the bubble's curvature; this simple model is sufficient for demonstration of qualitative mechanisms of cavitation.

To investigate heterogeneous bubble nucleation, we consider a spherical pore of radius R with a non-wetting defect of radius r_def. The bubble must be pinned to the defect perimeter, which represents the three phase contact line, see schematics in Figure 3.14, top. Note that because in this case the contact line cannot move, the concept of the contact angle in not applicable. The bubble pinned to the defect boundary exists even at the saturation conditions, $p/p_0 = 1$, when the bubble interface is flat (e.g. $r_\mathrm{b} \to \infty$) and the bubble forms a spherical cap. Upon desorption, as the relative pressure decreases, the equilibrium bubble grows maintaining a spherical vapor-liquid interface of radius r_b, which decreases with pressure according to eq (3.7). The pinned bubble is bound by two coinciding spherical caps of radii R and r_b; its volume through simple geometrical relationships, see footnote [1]. It is important to note that at a given relative pressure there are two equilibrium configurations, small and large, for the bubble of given size r_{b}, provided the bubble radius r_{b} is larger than the defect radius r_{def} and smaller than the pore radius R, $r_{\mathrm{def}} < r_{\mathrm{b}} < R$. It is easily seen that the small bubble is stable against variations of its volume at given vapor pressure, and large bubble is unstable. In other words, the unstable large bubble represents the critical nucleus that should be formed due to thermal fluctuations.

In the bottom of Figure 3.14, we present the desorption isotherm that corresponds to the growing bubble pinned at the defect, calculated according to the Kelvin-Laplace equation (eq (3.7)). It represents the normalized fraction of the pore volume filled by condensed fluid (i.e. the relative fluid volume outside the bubble, $[(4/3)\pi R^3 - V(r_{\mathrm{b}})]/[(4/3)\pi R^3]$). This isotherm has two branches: the upper branch is that of the stable bubbles with the radius varying from R at $p/p_0 = \exp(-a_{\mathrm{c}}/R)$ to r_{def} at $p/p_0 = \exp(-a_{\mathrm{c}}/r_{\mathrm{def}})$. Note that for clarity the isotherms were continued to $p/p_0 = 1$, corresponding to the flat liquid-vapor interface. In this case, the radius of the stable bubble exceeds the radius of the pore, $r_{\mathrm{b}} > R$. The lower backward branch is that of the unstable bubble, or critical nuclei, with the radius varying from r_{def} to R. The leftmost turnover point corresponds to the spinodal, at which the stable and unstable bubble configurations merge; the radius of such spinodal bubble equals to the radius of the defect, $r_{\mathrm{b}} = r_{\mathrm{def}}$. Calculations in Figure 3.14 were performed for the pore of $R = 7.36\sigma$, to correspond with MC calculations presented below, and with defects of several sizes. As the defect size increases, the spinodal point moves towards larger vapor pressures.

Nucleation of the critical bubble is a fluctuation-driven event. The probability to form a critical bubble depends on the work of its formation from the stable bubble at given vapor pressure, or the nucleation barrier that is determined by the work of surface formation and the work of compression. Thus, the nucleation barrier of heterogeneous cavitation is determined by the differences of surface areas and volumes of unstable and stable bubbles

[1] The volume of each bubble corresponds to the difference in volumes of intersecting spherical caps, one cap being the bubble, a sphere of radius r_{b} and second the pore, a sphere of radius R. The height of the cap is governed by the whether the bubble is the small or large configuration, and the size of the defect, r_{def}. The volume of the stable (smaller) bubble is $V_s = \frac{\pi}{3}\left[R^3(\cos^3\alpha - 3\cos\alpha + 2) + r_{\mathrm{b}}^3(\cos^3\beta - 3\cos\beta + 2)\right]$ and the larger, unstable bubble is $V_u = \frac{\pi}{3}\left[R^3(\cos^3\alpha - 3\cos\alpha + 2) - r_{\mathrm{b}}^3(\cos^3\beta - 3\cos\beta - 2)\right]$, where α and β are $\alpha = \arcsin\left(r_{\mathrm{def}}/R\right)$ and $\beta = \arcsin\left(r_{\mathrm{def}}/r_{\mathrm{b}}\right)$. Similarly, the surface areas of the stable and unstable bubble are $S_s = 2\pi r_{\mathrm{b}}^2(1 - \cos\beta)$ and $S_u = 2\pi r_{\mathrm{b}}^2(1 + \cos\beta)$, respectively.

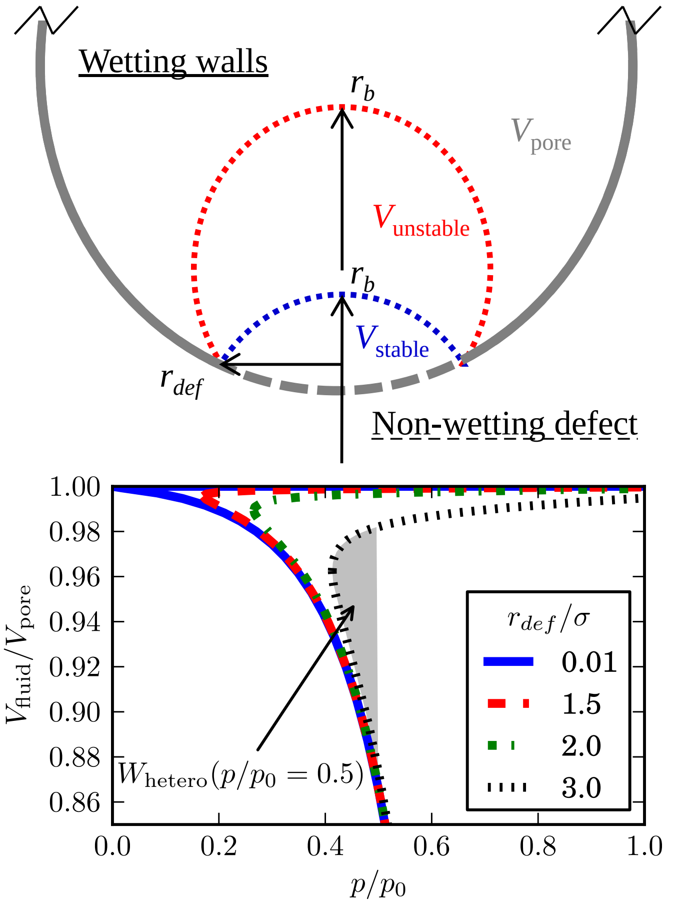


Figure 3.14: Top: Schematics of heterogeneous bubble growth at the circular non-wetting defect in the spherical pore. The bubble is pinned to the defect; equilibrium conditions permit two solutions, the smaller stable bubble, and the larger unstable bubble. Note that the radius r_{b} is identical for these two configurations and it is determined by eq (3.6). Bottom: Isotherms calculated using eq (3.8) and the appropriate bubble volume (see footnote [1]), displayed as the fraction of pore volume filled by liquid condensate versus the relative pressure. The ascending, upper branch corresponds to the stable bubble, and the descending, lower branch to the unstable bubble. The branches meet at a spinodal-type point where their respective bubbles' volumes are equal and the work of formation vanishes; at this point $r_{\mathrm{b}} = r_{\mathrm{def}}$ and the bubble is semi-spherical. The work of formation of an unstable bubble is related to the area between the two branches at a given pressure.

by the analog to eq (3.5), as

$$W_{\text{hetero}}(r_{\text{b}}, r_{\text{def}}) = \gamma(S_u - S_s) - \frac{2\gamma}{r_{\text{b}}}(V_u - V_s) \tag{3.9}$$

where the subscript s, u refers to the stable or unstable bubble of given radius r_{b}. At the spinodal ($r_{\text{b}} = r_{\text{def}}$), the nucleation barrier vanishes. At $r_{\text{b}} > r_{\text{def}}$, the nucleation barrier can be calculated by integrating the adsorption isotherm between stable and unstable states along the chemical potential, as indicated by the shaded area in Figure 3.14, bottom.

Either heterogeneous or homogeneous nucleation in condensed fluid is possible at given external conditions. It is obvious that the energy barrier of heterogeneous nucleation is always smaller than that of homogeneous nucleation, $W_{hetero}(p/p_0) < W_{\text{homo}}(p/p_0)$. However, to compare the relative probabilities of homogeneous and heterogeneous nucleation, one has to take into account not only the difference in nucleation barriers but also the volume factor, which increases the probability of homogeneous nucleation. To trigger heterogeneous cavitation, the critical bubble may nucleate only at the site of the defect. To trigger homogeneous cavitation, the critical bubble has the whole volume of liquid as potential nucleation ‘sites’. Assuming that the homogeneous critical nucleus of radius r_{b} can be formed with its center located within the spherical core of the pore of radius $R - r_{\text{b}}$, the volume factor can be estimated as the ideal term in the nucleus free energy,

$$F_{\text{id}} = -k_{\text{B}}T \ln \left[\frac{\frac{4}{3}\pi(R - r_{\text{b}})^3}{\Lambda^3} \right], \tag{3.10}$$

where k_{B} is the Boltzmann constant and Λ is the de Broglie length of a fluid molecule. As such, the ratio of probabilities of a homogeneous nucleation event as opposed to a heterogeneous one at given external conditions can be evaluated as

$$\frac{P_{\text{homo}}(r_{\text{b}})}{P_{\text{hetero}}(r_{\text{b}}, r_{\text{def}})} = \frac{\frac{4}{3}\pi(R - r_{\text{b}})^3}{\Lambda^3} \exp\left[-\frac{W_{\text{homo}}(r_{\text{b}}) - W_{\text{hetero}}(r_{\text{b}}, r_{\text{def}})}{k_{\text{B}}T} \right], \tag{3.11}$$

where W_{homo} and W_{hetero} are determined by eq (3.5) and (3.9), respectively, for a given r_{b}. The prefactor reflects the entropic contribution given by eq (3.9). This term can be thought of as the number of potential microscopic nucleation sites in the volume of condensed

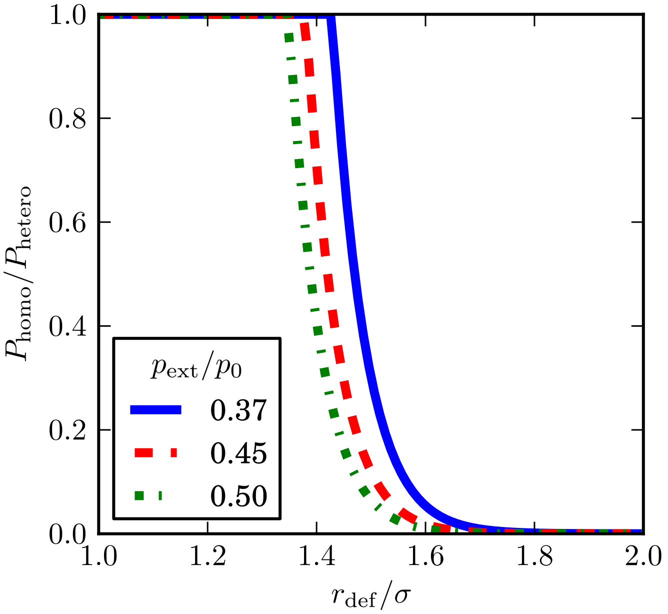


Figure 3.15: The CNT ratio of probabilities of homogeneous and heterogeneous nucleation as a function of the defect size for a spherical pore $R/\sigma = 7.36$, at several characteristic pressures. Small defects do not influence nucleation. Defects larger than $r_{\text{def}}/\sigma \sim 1.5$ present a large enough free energy benefit for heterogeneous nucleation to overcome the entropic advantage of homogeneous nucleation.

fluid. Equation (3.11) predicts a transition from homogeneous cavitation to heterogeneous cavitation as the diameter of the defect grows, and is illustrated in Figure 3.15. The ratio of nucleation probabilities was calculated at various defect sizes for the pore $R = 7.36\sigma$, at several typical external pressures. The transition from homogeneous nucleation to heterogeneous nucleation is observed when the diameter of the defect is about 3σ. It is important to remember that although the predictions of CNT are qualitatively correct, they cannot be taken as quantitative estimates due to the macroscopic nature of CNT. A more precise description can be achieved by using MC simulation of the adsorption-desorption process that is described below.

3.2.3 MC Model and Simulation Details

Monte Carlo simulations were used to model cavitation of Lennard-Jones (LJ) fluid in a spherical pore. The pore wall interacts with the adsorbate particles via a continuous, site-averaged LJ potential. We tested a pore with external diameter (distance between the centers of two opposing LJ solid particles) $d_{\text{ext}} = 2R = 5.72\ nm = 15.82\sigma$ and accessible diameter (diameter of the volume in which $U_{\text{adsorption}} > 0$) [204] $d_{acc} = 5.54\ nm = 14.72\sigma$. The size of the defect, in diameter, was varied from 0 to 2.21 nm (6.12σ). The calculations were done for the LJ model of nitrogen with fluid-fluid interaction parameters of $\epsilon_{\text{FF}}/k_{\text{B}} = 101.5$ K and $\sigma_{\text{FF}} = \sigma = 0.36154$ nm [162]. The potential energy was truncated when inter-particle distance was greater than 10σ. The solid-fluid LJ potential was calculated by integral average over the pore wall surface [27, 205]. The solid-fluid interaction parameters were selected to emulate adsorption of nitrogen on silica glass (energy parameter $\epsilon_{\text{SF}}/k_{\text{B}} = 147.3$ K, size parameter $\sigma_{\text{SF}} = 0.317$ nm and surface density of adsorption sites $\rho_{\text{S}} = 15.3\ \text{nm}^{-2}$) [164]. For all simulations, temperature was set to the normal boiling point of liquid nitrogen, $T = 77.36$ K. The Johnson-Zollweg-Gubbins (JZG) equation of state for LJ fluids was used to relate the chemical potential to the relative vapor pressure [165].

A non-wetting defect of variable size was introduced as a round spot on the pore wall which does not exert the attractive potential. Utilizing the cylindrical symmetry of the system, the adsorption potential inside the pore at a point characterized by the distance from the pore center r and the polar angle θ can be expressed as the integral over the sphere surface excluding the surface of spherical cap occupied by the defect,

$$U_{\text{SF}}(r, \theta, \delta, R) = 4\epsilon_{\text{SF}}\rho_{\text{S}}R^2 \int_0^{2\pi}\int_\delta^\pi \left(\frac{\sigma_{\text{SF}}^{12}}{\xi^{12}} - \frac{\sigma_{\text{SF}}^{6}}{\xi^{6}}\right) \sin\theta_0 \, \mathrm{d}\theta_0 \, \mathrm{d}\phi_0 \tag{3.12}$$

where r and θ are radius and inclination of a point within the sphere, 2δ is the apex angle of the defect, and subscript “0” indicates integration variables over the spherical surface. Figure 3.16 illustrates the geometry of the pore/defect system. The distance from a point within the sphere (e.g. an adsorbate molecule) and on the sphere (e.g. an adsorption site) is given by the following:

$$\xi^2 = R^2 + r^2 - 2Rr\left[\sin\theta\sin\theta_0\cos(\phi_0 - \phi) + \cos\theta\cos\theta_0\right] \tag{3.13}$$

By performing the integration in θ_0 from δ to π, rather than 0 to π, a conical volume is introduced with no contribution to the potential. A potential energy map of the xz-plane of a system with a relatively large defect is presented in Figure 3.16. A simple trigonometric equation relates δ to the radius of the defect,

$$r_{\text{def}} = R \sin \delta \tag{3.14}$$

The value r_{def} in eq (3.14) specifies the external, center to center radius of a spherical defect on the pore wall. The accessible diameter of the defect can be estimated by subtracting $\sigma_{\text{SF}}/\sigma_{\text{FF}}$ from $d_{\text{def}} = 2r_{\text{def}}$. For convenience, all diameters herein will refer to the accessible volume. Increasing the defect diameter reduces the total adsorption potential, which affects the condensation and desorption pressures [1]. However, because the defect is not-wetting, its surface will always be a more favorable nucleation site; thus its area rather than the relative difference in adsorption potential is the primary variable we are concerned with.

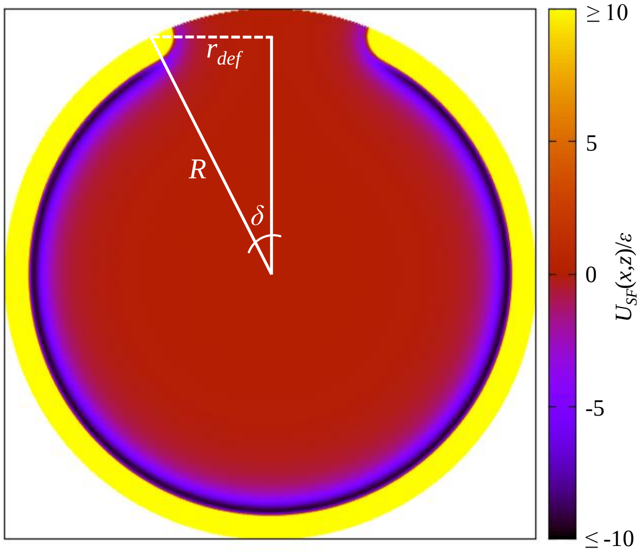


Figure 3.16: The solid-fluid potential energy map on the xz-plane of a 5.54 nm pore with a 2.21 nm non-wetting defect introduced. Overlay: geometry of eq (3.12) and (3.14).

The adsorption potential (eq (3.12)) is a continuous function of two position variables, r and θ. It was integrated numerically and tabulated on a grid of 2000 × 500, respectively. The spherical-linear interpolation method [206] was used to calculate the potential energy from the tabulated values. Boundary conditions at the non-wetting defect were set as hard-wall repulsion. In general, it is possible to extend this model to a partial wetting defect by adding the integrated LJ potential of the conical defect, but with a value of solid-fluid interaction potential smaller than ϵ_{SF}.

Monte Carlo simulations were performed using the grand canonical ensemble (GCMC) [166] and the mesocanonical ensemble (MCMC) [9, 10]. The MCMC method, also known as the gauge cell method, introduces a fixed reservoir of particles (called the gauge cell) that are permitted to be exchanged with the system cell (the pore). The MCMC method is instrumental in calculating chemical potentials in small and inhomogeneous system [10]. It has been extended since its original formulation [9] to multi-component fluids [49] and polymer chains [207, 208]. The MCMC method allows one to efficiently simulate nucleation phenomena and generate unstable, intermediate states by suppressing fluctuations that would, if unconstrained, facilitate a spontaneous phase change in an open system. As such, MCMC makes possible a continuous, backwards trajectory of the adsorption isotherm that corresponds to the unstable bubbles, or nuclei, qualitatively similar to the theoretical isotherm shown in Figure 3.14. Once such a continuous canonical isotherm is generated, the nucleation barrier at given vapor pressure can be calculated as the work of formation of the unstable bubble from the stable one by thermodynamic integration [1, 54, 55, 209].

GCMC was used to emulate experimental adsorption conditions of constant chemical potential, volume and temperature. Because nucleation of the critical bubble is a fluctuation-driven process, the cavitation events are distributed over a certain range of pressures. To obtain this distribution, a number of sequential GCMC desorption trajectories were performed in the following manner:

1. A true random seed [210] is used to build a random, independent configuration of particles at a given stable liquid density.

2. The configuration is equilibrated using a canonical ensemble (NVT) MC simulation.

3. The equilibrated configuration is used as the initial configuration for GCMC simulation at $\mu \gg \mu_{\text{cav}}$.

4. If the simulation undergoes phase change to the vapor-like state, the simulation is stopped and bulk chemical potential is recorded as 1 sample in the distribution of cavitation chemical potentials. If the simulation finishes in the liquid state, μ is decreased by a small amount and a new GCMC simulation is started.

5. Step 4 is repeated until the simulation reaches a low-density, vapor-like state, the last value of μ is taken as the chemical potential of cavitation μ_{cav}.

Multiple desorption trajectories are calculated independently for a given system, and the distribution of cavitation chemical potential is obtained and analyzed using traditional statistical methods. All simulations used the same number of MC steps so they can be compared to each other; 300,000 MC steps (each step a single attempt to displace or exchange a particle, with equal probability) of 500 equilibrating sets that are discarded, and 500 production sets that are averaged and used as the result. It has been shown that the width of GCMC hysteresis (and thus, the cavitation pressure) is affected by the length of the underlying Markov chain [211]. Therefore, comparisons are only made between average cavitation pressures calculated using equal length GCMC runs.

3.2.4 Results

We calculated grand canonical and gauge cell canonical isotherms for LJ nitrogen at 77.36 K adsorbing in a silica pore of diameter 5.54 nm with a single non-wetting heterogeneous defect with diameters ranging from 0 to 2.12 nm. The MC isotherms are displayed in Figure 3.17. On the vapor branch of the canonical isotherms that corresponds to the formation of the adsorbed film, the adsorption depends on the size of the defect. As the defect size increases, the averaged adsorption potential, as well as the surface available for the film formation, decreases, resulting in a lower adsorption relative to the defect-free pore at the same vapor pressure. This trend continues with the position of the vapor spinodal, the maximum chemical potential of the vapor branch that reflects the limit of stability of the adsorbed film. However, the position of the liquid spinodal (the minimum chemical potential of the liquid branch that reflects the limit of stability of the liquid phase) shows two distinct behaviors, depending on the size of the defect. When the defect is less than 1.41 nm, the position of the liquid spinodal increases only slightly with the size of the defect. At larger defects, the spinodal shifts to higher pressures, progressively with the size of the defect. A similar dependence on the defect size is observed on the GCMC isotherms. The condensation pressure increases with defect size, but the cavitation pressure remains mostly unchanged, for defects smaller that 1.41 nm. With larger defects, the cavitation pressure increases progressively with the defect size.

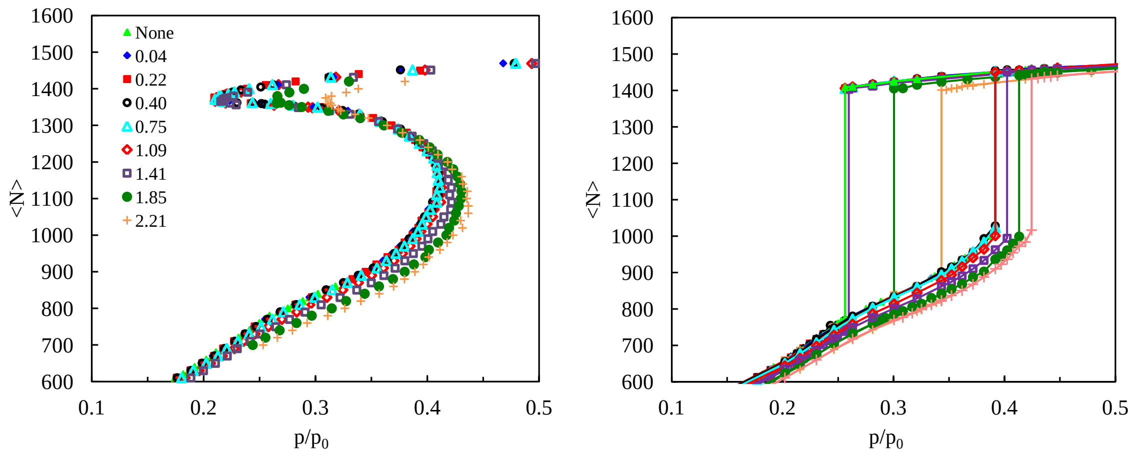


Figure 3.17: MCMC isotherms (left) and GCMC isotherms (right) for LJ nitrogen adsorbing in a silica pore of 5.54 nm at 77.36 K, with a heterogeneous defect of varying diameter in nm.

To estimate the distribution of cavitation pressure in GCMC simulations, we performed a series of parallel GCMC desorption runs starting from randomly chosen initial configurations equilibrated at a fluid density known *a pirori* to be a stable liquid. Standard statistical methods where used to obtain the mean cavitation pressure and its dispersion. The cavitation pressures were calculated for the defects of 0, 0.40, 1.41, 1.47, 1.61, 1.85 and 2.21 nm in diameter. Selected distributions from this study are presented in Figure 3.18. The distributions are bell-shaped. In the pores with larger defects that exhibit predominantly heterogeneous mechanism of cavitation, the distribution of cavitation pressures appears to be narrower than that in the pores with small defects and the homogeneous mechanism of cavitation. The averaged results of these desorption trajectories are displayed in Figure 3.19. The data can be divided into two regimes of weak and strong dependence on the defect size. From 0 to 1.41 nm diameter of the defect, the average cavitation pressure marginally increases. The increase is small but statistically significant (see Figure 3.19, inset). For defects larger than 1.41 nm, a strong dependence is observed. It is important to remember that the reported cavitation pressures should not correspond to experimental values. The conditions of a MC simulation represent a "perfect" situation of fixed temperature and vapor pressure and with significantly smaller fluctuations than in real experiments. However, relative comparisons between calculated GCMC pressures are permissible.

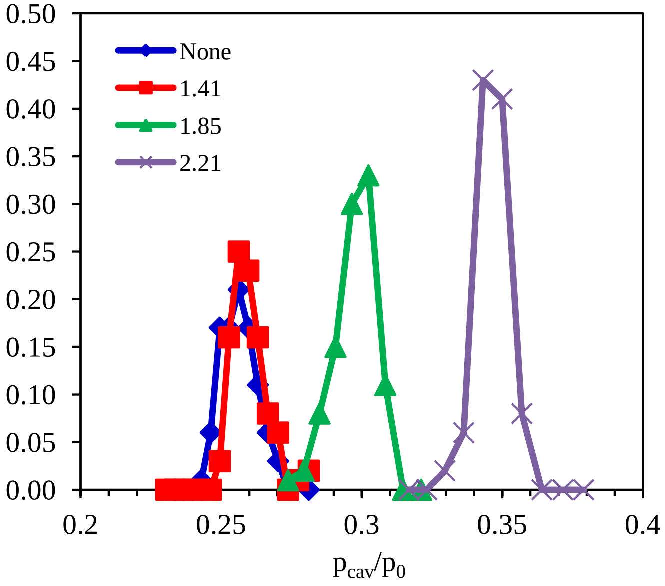


Figure 3.18: Selected distributions for the pressure of cavitation calculated from a series of independent GCMC desorption paths: pores with no defect (diamond), $d_{\text{def}} = 1.41$ nm (square), 1.85 nm (triangle), and 2.21 nm (cross).

Determination of the type of nucleation was done by tracing the progress of each MC simulation for states near the point of desorption. During the MC simulation, the coordinates of each successful MC move were saved, up to 825,000 moves total. At a specified fluid density (small enough to ensure a phase change occurred; $N\sigma^3/V = 0.4$ for this study), the tracking was stopped and the current configuration written to disk. With this information, the original system (up to 825,000 successful moves ago) can be fully recreated, and examined step-by-step. Figure 3.20 shows several intermediate states along a desorption path for the pore with none, small (0.40 nm) and large (1.85 nm) defect. The snapshots were created by dividing the available trajectory into 20 frames, and calculating the average particle positions of each frame using 40 equal-spaced samples from the configuration trajectory. Thus, the color intensity indicates the probability of a particle intersecting the xz-plane for a given span of simulation time. With configuration snapshots available, formation of the critical nuclei can be observed as fluid particles 'desorb' from the pore body. We found that if the defect is larger than 1.41 nm, all nucleation appears to be heterogeneous, that is, the critical

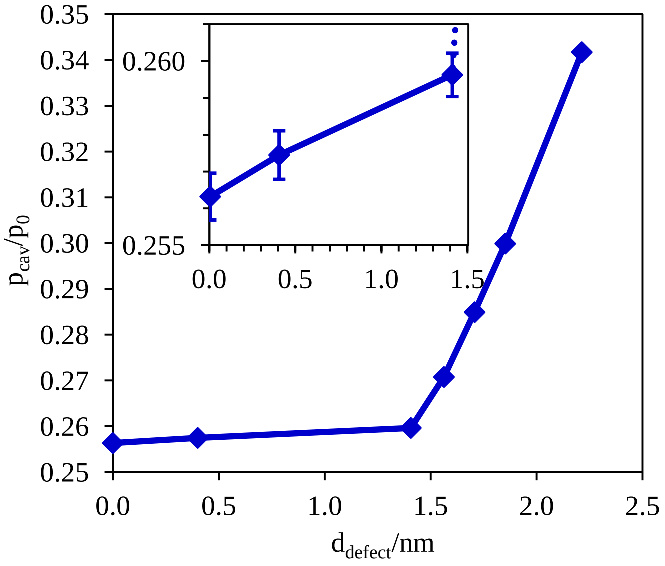


Figure 3.19: The average pressure of cavitation calculated in a series of independent GCMC simulations. The sharp change of slope indicates the transition from homogeneous to heterogeneous nucleation, determined by examination of particle configurations. Inset zooms on the small defect regime of homogeneous cavitation; error bars represent the standard error of the distribution.

nuclei forms at the site of defect and expands from that point, as illustrated in Figure 3.20. Below this defect size, nucleation is almost always homogeneous; forming at or near the center of the pore. Interestingly, the location of the critical bubble varies on the xz-plane more than expected. In pores with a defect, the location of the critical cavity varied from the center of the pore to near the defect (but still entirely within the liquid) to the site of the defect. Even in the no-defect system, the cavity forms near the center, but not always at the center. The relative frequency of each type of cavitation is presented in Figure 3.21. To compile the statistics of homogeneous and heterogeneous cavitation events from MC runs, trajectories from 30 desorption runs were stored and analyzed. The CNT prediction is also displayed in Figure 3.21 for comparison. The CNT curve in Figure 3.21 was obtained with eq (3.11) by interpolating the cavitation pressure for a given d_{def}, and then calculating the critical bubble radius (eq (3.6)) from that pressure. Instead of using the Kelvin-Laplace relation as above, the equilibrium liquid and vapor pressures were obtained through the JZG equation of state for LJ fluid [165]. The liquid-vapor surface tension ($\gamma\sigma^2/\epsilon = 1.022$ for infinite cutoff LJ fluid) was calculated by linear interpolation at $T^* = 0.762$ from three

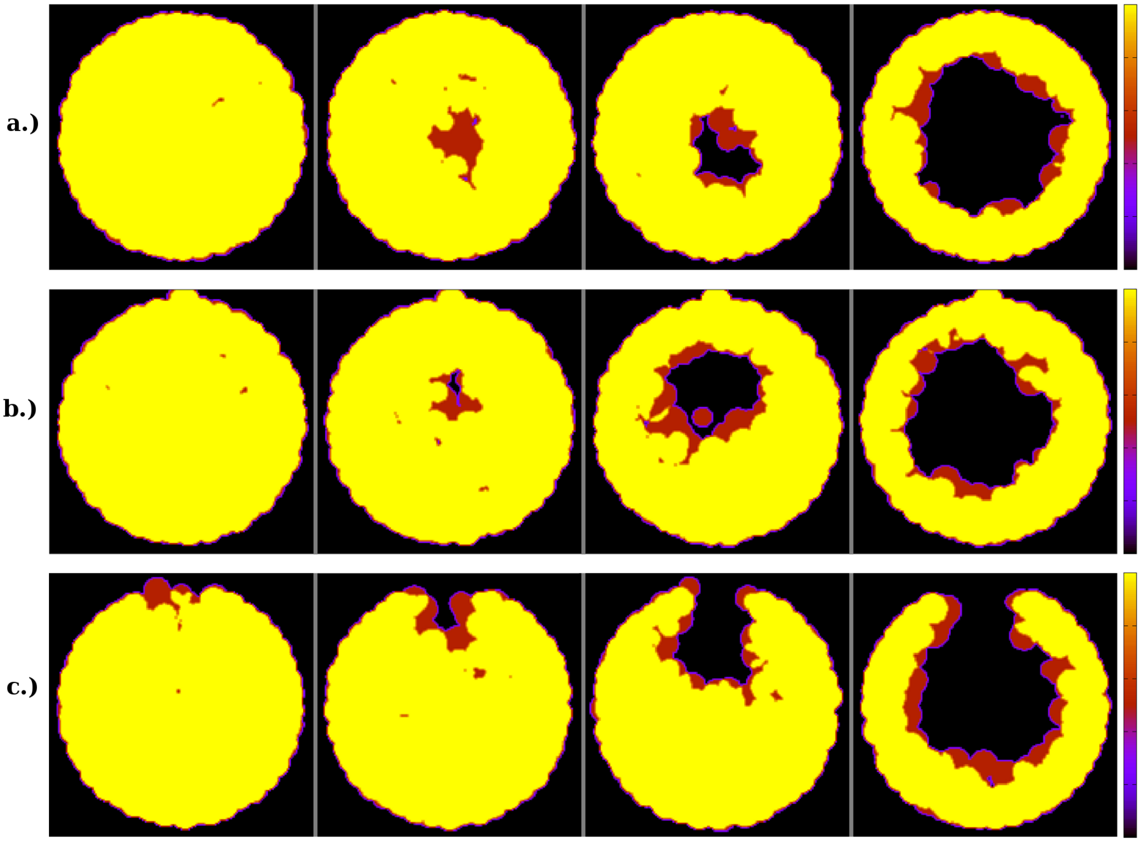


Figure 3.20: Snapshots of growing bubbles quantified by the particle probabilities on the xz-plane during desorption; yellow indicates the probability of unity and black the probability of zero. a.) Pore without defect shows the nucleation of the critical cavity at the center of the pore. b.) Pore with a small defect (0.40 nm) still exhibits homogeneous nucleation. c.) Pore with 1.85 nm defect exhibits a flat interface at complete pore filling, followed by the growth the bubble pinned to the defect; this bubble grows until the vapor-like state is reached.

studies of LJ surface tension [171, 212, 213], and then averaged. The same qualitative shape of the probability curve is obtained using both CNT and MC. CNT predicts the critical defect size to be ∼1.0 nm (∼2.8σ, very close the prediction in Fig. 3.15 of ∼3σ), while the MC transition to heterogeneous cavitation takes place at $d_{\mathrm{def}} \sim 1.5$ nm, or about 4σ. This quantitative discrepancy between MC and CNT predictions is expected.

Nucleation barriers were calculated via thermodynamic integration of the canonical isotherm from gauge cell calculations to yield the difference in grand thermodynamic potential,

$$\Omega(\mu) - \Omega(\mu_{\mathrm{ref}}) = -\int_{\mu_{\mathrm{ref}}}^{\mu} N(\mu)\,\mathrm{d}\mu \tag{3.15}$$

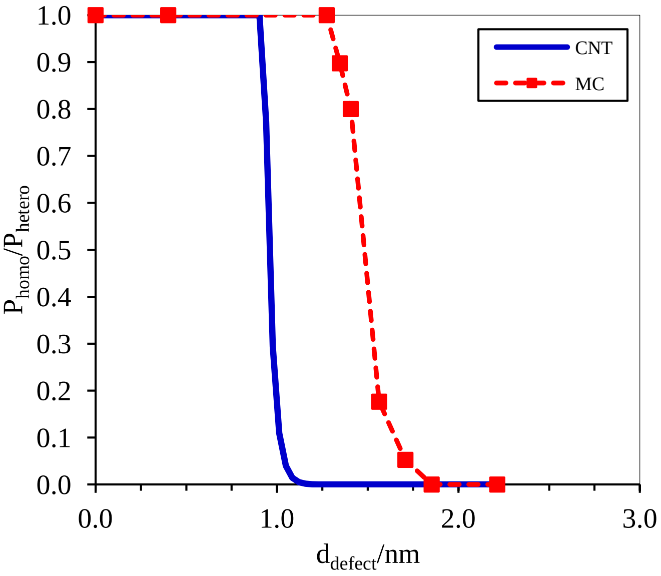


Figure 3.21: Ratio of the probability of homogeneous nucleation opposed to heterogeneous nucleation. The solid line indicates the prediction from CNT for a system similar to that studied by MC (LJ nitrogen using JZG equation [165]). The squares are the results from MC simulation trajectories, discussed below.

Since cavitation is a nucleation based phenomenon, an energy barrier must be crossed for it to occur. The calculated nucleation barriers are presented in Figure 3.22. Systems exhibiting homogeneous cavitation (i.e., $d_{\text{defect}}/\text{nm} = 0$, 0.40, and 1.41) are found to have nearly identical nucleation barriers. This is expected, as systems with cavitation at similar pressures would be crossing the same energy barrier. When heterogeneous cavitation is found to be the dominant mechanism, the nucleation barrier for a given pressure shifts significantly to lower values. In other words, for a given nucleation barrier, the pressure is found to be constant for homogeneous cavitation, and increasing with defect size for heterogeneous cavitation. Observed cavitations pressures (Fig. 3.19) indicate that the thermal barrier to overcome for GCMC nucleation, regardless of its nature, is 8–10 $k_{\text{B}}T$, which is in an acceptable range comparing to literature results [172]. For comparison, results from CNT are included as well on the right hand side of Fig. 3.22. The CNT curves were calculated using a similar method as described for Fig. 3.21[2]. We see that a defect of $d_{\text{def}}/\text{nm} = 0.90$ is

[2] It is worth noting that the numerical results of CNT and GCMC calculations for no defect pores differ somewhat from the data in our previous paper because of the different value of the LJ potential cut-off of 10σ instead 5σ. This also affected the CNT calculations by altering the equation of state and the surface tension value (1.022 vs. 0.922).

the critical size where the predominant nucleation mechanism changes from homogeneous to heterogeneous. The smallest defect (0.72 nm) has a larger barrier than the homogeneous fluid, while the largest defect (1.08 nm) has a smaller barrier, thus dictating the nature of nucleation. The intermediate curve (0.90 nm) is similar in value to the homogeneous curve; at low pressures (small critical bubble), homogeneous nucleation would be more likely, and at higher pressures (larger critical bubble), heterogeneous nucleation would be more likely. As above, the overall transition occurs at a smaller ddef than observed in MC simulations, as expected.

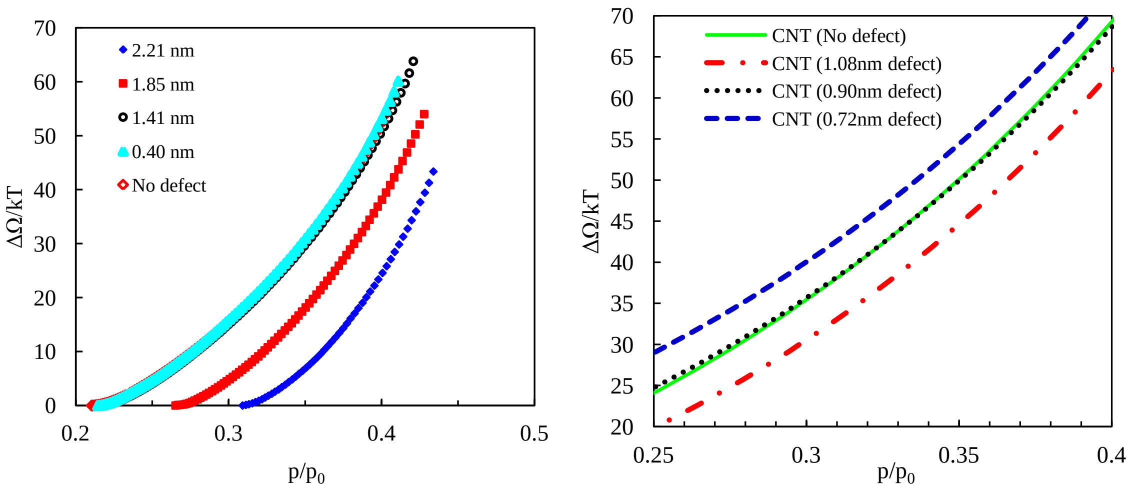


Figure 3.22: The work of formation of the critical nuclei calculated by thermodynamic integration of the canonical MC isotherms (left) and using the CNT equations for homogeneous and heterogeneous cavitation (right). For comparison of the rates of homogeneous and heterogeneous cavitation, the work of homogeneous cavitation (eq (3.5)) was corrected by the volume factor (eq (3.10)). CNT predicts that the transition to heterogeneous cavitation occurs at a defect ddef = 0.90 nm, while MC simulations indicate $d_{\text{def}} \sim 1.40$ nm. Note the large difference in the nucleation barriers calculated with CNT and MC simulations; the reason for this discrepancy is that the size of nuclei in this example does not exceed $d \sim 0.90$ nm, which is too small for the CNT theory to be quantitatively accurate.

3.2.5 Conclusions

We investigated a possible scenario of heterogeneous cavitation of metastable fluid in the process of desorption from mesoporous materials. As a model system, we considered an LJ fluid confined to spherical pores with non-wetting surface defects. We showed that the non-wetting defect facilitates the formation of heterogeneous nuclei, and the probability of heterogeneous cavitation increases with the defect size. As such, the transition from the homogeneous to heterogeneous cavitation occurs in pores with sufficiently large defects. We employed the classical nucleation theory (CNT) and Monte Carlo (MC) simulations to quantify the impact of surface defects on the mechanism of cavitation.

The CNT model was modified to account for heterogeneous nucleation of bubbles at the non-wetting defect. We found that the CNT model describes the transition from the homogeneous to heterogeneous cavitation at sufficiently large surface defects. In such pores, the vapor pressure of cavitation exceeds the vapor pressure characteristic of homogeneous cavitation. The results of the CNT model were examined with the MC simulation study using the mesocanonical ensemble for generating the critical nuclei configurations and determining the nucleation barriers, and the grand canonical ensemble for calculating the distribution of pressures at which cavitation occurs in simulation. The MC results are qualitatively similar to that predicted by CNT and provide a more reliable quantitative estimate of the pressures of cavitation presented in Figure 3.19. We found that defects smaller than 1.41 nm (or about 5 molecular diameters) in diameter do not change the nature of bubble nucleation from the homogeneous case with no defect present. Conversely, surface defects larger than 1.41 nm facilitate competitive heterogeneous nucleation. This leads to a sharp increase in the cavitation pressure with the size of the defect, as the nucleation barrier for forming a heterogeneous bubble pinned to the defect becomes progressively smaller than the nucleation barrier of homogeneous cavitation.

It is worth to note that although the experimental motivation of this work was a series of papers on the mechanism of nitrogen and argon adsorption on porous silicon samples [177, 195, 196], our conclusions cannot be applied to this particular system, since it is unlikely that non-wetting (to nitrogen) defects may be present on a silicon surface. At the same time, the scenario of heterogeneous cavitation may be an important factor in adsorption of water and other polar vapors [214]. To consider these systems, it will be necessary to perform a similar simulation study with forcefields adequate for water and water-substrate interaction. Among the other physical processes for which heterogeneous nucleation of bubbles in pores may be an important factor are: drying of porous and fibrous materials [183], sap breakage in plants [215], fluid flow in soil and rock [216], as well as physiological phenomena such as decompression sickness.

3.3 Capillary Condensation Hysteresis in Overlapping Spherical Pores

3.3.1 Introduction

Gas adsorption is a standard technique for characterization of porous materials. Often experimental adsorption isotherms are irreversible, showing adsorption-desorption hysteresis. Proper interpretation of the physical mechanisms of hysteresis is crucial for assessment of the pore size distributions from adsorption isotherms [29, 217–219]. For a number of materials, predicting hysteresis when modeling adsorption requires taking into account not just pore geometry, but pore connectivity [180, 189, 220] and/or morphological defects [194, 221].

Recent developments in material science [25] enabled fabrication of mesoporous silicas and carbons with three-dimensional regular and hierarchical pore networks and tunable pore size distributions [155]. Usually, the cage-like wider mesopores are connected by narrower channels; such pores are called "ink-bottle" pores [137], referring to larger pores as *cavities* and to smaller connecting channels as *necks*, as illustrated in Figure 3.23a. Ink-bottle pores are of interest due to the unique confinement effects observed upon adsorption depending on the ranges of cavity and neck sizes [1, 27, 36].

While the capillary condensation process upon adsorption in ink-bottle pores is determined by the size of the cavity and takes place near the vapor spinodal of the confined fluid [18], the desorption (evaporation) process is more complicated. Experimental observations with specially designed ordered materials and respective theoretical analysis [27] revealed three different mechanisms of evaporation from cage-like mesopores: (i) pore blocking controlled desorption, (ii) spontaneous evaporation due to cavitation, and (iii) near-equilibrium desorption. The prevalence of a given mechanism, and thus the pressure p_{d} at which desorption occurs, depends mainly on the relation between the size of the cavity and the size of the necks [1, 27, 36]. Near-equilibrium desorption is possible from the cavities that have immediate access to the vapor phase through relatively wide openings, and thus are effectively unblocked. Once the vapor pressure reaches the vapor-liquid equilibrium (VLE) pressure $p_{\mathrm{e}}^{\mathrm{cavity}}$, desorption from the main cavity proceeds via a receding meniscus; therefore $p_{\mathrm{d}} \approx p_{\mathrm{e}}$. However, if wide cavities are connected with narrower necks, so that the fluid in the neck has a lower VLE pressure $p_{\mathrm{e}}^{\mathrm{neck}}$ than that of the fluid in the cavity, the neck effectively "blocks" desorption from the cavity. Emptying of the pores occurs at the pressure of equilibrium desorption of the neck, and as such $p_{\mathrm{d}} = p_{\mathrm{e}}^{\mathrm{neck}}$ and becomes a function of the neck size. Finally, if the connecting necks are even narrower, the fluid in the cavity becomes substantially metastable, and the vapor pressure may reach the spinodal point of the confined liquid before the equilibrium meniscus can form in the neck. Thus, the fluid desorbs from the pore by the cavitation mechanism, which involves fluctuation-driven formation and growth of a bubble [193]. In this scenario, $p_{\mathrm{d}} \ll p_{\mathrm{e}}$, and p_{d} has little or no dependence on the size of the necks [1]. Available models of pore blocking [52] and cavitation [1] (which represent the pores as spheres connected with cylinders) describe adsorption in materials like FDU-1 [222] or SBA-16 [150] silicas reasonably well.

However, there exist mesoporous materials with cage-like pores, which likely overlap rather than connect by narrow channels. One such class of materials are the novel three-dimensional ordered mesoporous (3DOm) carbons [223], synthesized by templating densely packed spherical nanoparticles. Nitrogen adsorption experiments in 3DOm carbons have revealed the IUPAC H1 type of hysteresis [18]. While experimental values of capillary condensation pressures are in good agreement with predictions from density functional theory (DFT) [224], the values of capillary evaporation pressures ($p_{\mathrm{d}}/p_0 \sim 0.6$–$0.7$, p_0 is the bulk VLE pressure) [223, 225] cannot be explained either as homogeneous cavitation or as equilibrium desorption. Other novel materials possessing similar structure and revealing similar capillary condensation hysteresis are ultraporous DVB resins [226].

In order to reveal the possible mechanisms of evaporation in such a "neckless" ink-bottle pore, we modeled a system of two overlapping spheres (Figure 3.23b). We should emphasize that we do not consider classical ink-bottle pores with connecting channels (necks), depicted in Figure 3.23a. Adsorption and confinement effects were modeled with Lennard-Jones (LJ) particles using grand canonical Monte Carlo (GCMC) [166] and mesocanonical Monte Carlo (MCMC) simulations techniques [9, 10]. The fluid-solid interaction was treated as

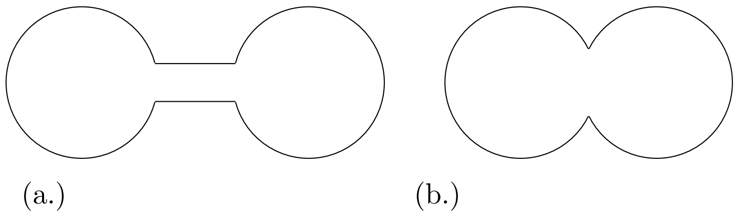


Figure 3.23: Models of interconnected spherical pores: (a.) Classical ink-bottle pore (b.) Overlapping pores, considered in the current work.

an integration of the LJ potential over two spherical layers, where the degree of overlap is set before integration. The resulting potential is continuous and depends on the window size. The goal of our simulations is to estimate the influence of the window size on the capillary condensation and evaporation pressures. In the current work we do not consider explicitly the connection of the pore with external reservoir, but a closed system of two pores, representing a characteristic element of the pore network. However, such model is sufficient to reveal the role of the window size.

The structure of the rest of the section is the following. The pore model and MC methods used are described in Section 3.3.2. Section 3.3.3 presents results of MC simulations, its comparison to previous models, and discussion. Conclusions are given in Section 3.3.4.

3.3.2 Model and Simulation Details

Monte Carlo simulations were used to model adsorption/desorption in the two-pore model, Figure 3.23b, with external pore diameter (the diameter of the sphere as the distance of a line drawn through the centers of solid atoms at opposite pore wall surfaces) $d_{\text{ext}} = 5.72$ nm $= 15.82\sigma_{\text{FF}}$ and internal diameter, defined as the distance encompassing two opposite fluid particles whose adsorption potential is zero. For the solid-fluid potential used, $d_{\text{int}} = d_{\text{ext}} - 1.7168\sigma_{\text{SF}} + \sigma_{\text{FF}}$, so $d_{\text{int}} = 5.54$ nm $= 15.32\sigma_{\text{FF}}$. We considered two characteristic values of the window size, $5\sigma_{\text{FF}}$ and $10\sigma_{\text{FF}}$. The adsorbing fluid was LJ nitrogen, with fluid-fluid interaction parameters of $\epsilon_{\text{FF}}/k_{\text{B}} = 101.5$ K and $\sigma_{\text{FF}} = 0.36154$ nm [162]. The potential energy was truncated when inter-particle distance was greater than $5\sigma_{\text{FF}}$. A spherical-integrated, site-averaged LJ potential was used to model the attractive adsorption potential [27, 205]. The solid-fluid interaction parameters were selected to emulate adsorption of nitrogen on silica (energy parameter $\epsilon_{\mathrm{SF}}/k_{\mathrm{B}} = 147.3$ K, size parameter $\sigma_{\mathrm{SF}} = 0.317$ nm and surface density of adsorption sites $\rho_{\mathrm{S}} = 15.3$ nm^{-2}) [164]. For all simulations, temperature was set to the normal boiling point of liquid nitrogen, $T = 77.36$ K. The dimensions and chemistry of the model system were selected for quantitative comparison to the single-sphere pore model, for which there are published results [1, 55].

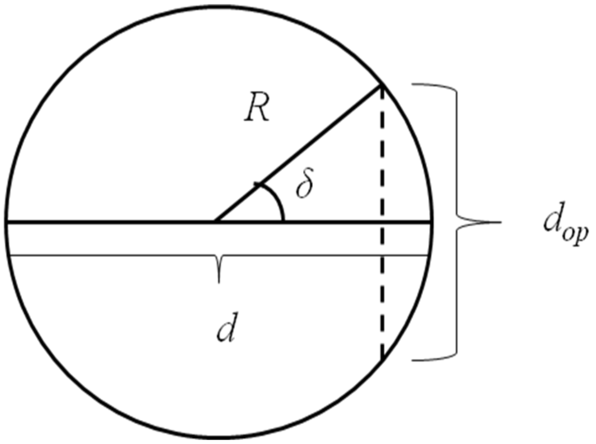


Figure 3.24: Model of the spherical pore with a window

The interaction potential between a fluid particle and the substrate was obtained by the superposition of potentials from a sphere with a window. The latter can be obtained by partial integration of LJ potential over the spherical domain depicted in Figure 3.24. The interaction potential between a fluid particle and the pore wall at a given point with spherical coordinates (r, θ, ϕ) is given by equation

$$U_{\mathrm{SF}}(r, \theta, \delta, R) = 4\epsilon_{\mathrm{SF}}\rho_{\mathrm{S}}R^2 \int_0^{2\pi}\int_\delta^\pi \left(\frac{\sigma_{\mathrm{SF}}^{12}}{\xi^{12}} - \frac{\sigma_{\mathrm{SF}}^6}{\xi^6}\right) \sin\theta_0 \,\mathrm{d}\theta_0 \,\mathrm{d}\phi_0 \tag{3.16}$$

where δ is size of the polar angle that contributes to the window, R is the external radius of the pore body, and subscript 0 indicates integration variables over the spherical surface. The distance ξ from a given point within the sphere (e.g. an adsorbate molecule) to a point on the surface of the sphere (e.g. an adsorption site) is

$$\xi^2 = R^2 + r^2 - 2Rr\left[\sin\theta\sin\theta_0\cos(\phi_0 - \phi) + \cos\theta\cos\theta_0\right] \tag{3.17}$$

Equation (3.16) does not depend on the azimuth coordinate ϕ due to the axial symmetry of the sphere with a round window. A simple trigonometric equation relates δ to the size (diameter) of the window, as shown in Figure 3.24. The value d_{op} specifies the external (center to center) diameter of the circular window between the two spheres. An interaction potential map of the considered systems is displayed in Figure 3.25. Equation (3.16) was integrated numerically over 200 bins in the (θ_0, ϕ_0) coordinates. We found that additional bins did not reduce integration error. The resulting function was stored in a 2001 × 2001 lookup table, and bilinear interpolation was used to compute the solid-fluid interaction energy for a given particle in cylindrical coordinates (ρ, z).

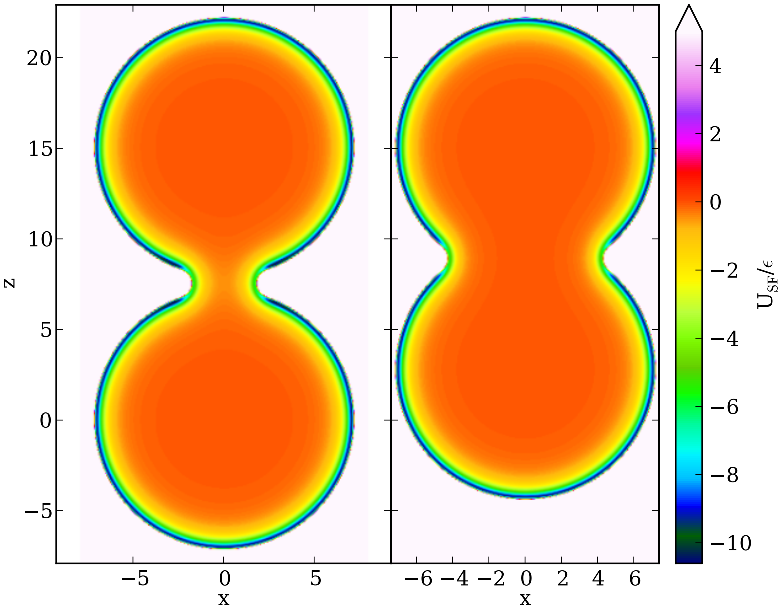


Figure 3.25: The solid-fluid potential energy map of a system of two spheres of $15.82\sigma_{\mathrm{FF}}$ each with (a) $5\sigma_{\mathrm{FF}}$ window (b) $10\sigma_{\mathrm{FF}}$ window. Both distance and potential are given in LJ units.

Monte Carlo simulations were performed in the grand canonical ensemble (GCMC) [166] and the mesocanonical ensemble (MCMC), also known as the gauge cell method [9, 10]. The MCMC method is instrumental in calculating chemical potentials in small and inhomogeneous system [10], and analysis of metastable and labile states inside the hysteresis loop [167]. It has been extended since its original formulation [9] to multi-component fluids [49] and polymer chains [207, 208]. MCMC introduces a fixed reservoir of particles (gauge cell) that is permitted to exchange fluid with the system cell (pore). Use of the gauge cell allows for efficient calculation of the chemical potential μ at given adsorption N (up to an order of magnitude faster than traditional Widom insertions) [208], as well as suppressing fluctuations that would, if unconstrained, facilitate a phase change in an open system. Therefore, the gauge cell allows for stabilization in the system cell of metastable and labile states, like critical nuclei, since the combined system (system cell + gauge cell) is thermodynamically stable. As such, the MCMC simulation enables generation of a continuous canonical adsorption isotherm $N(\mu)$ of van der Waals type with a backwards trajectory that connects the vapor and liquid spinodals. In doing so, the Helmholtz free energy, and respectively, the work of formation of the critical nuclei that precedes cavitation desorption, can be calculated using thermodynamic integration.

GCMC was used to emulate experimental adsorption conditions; that is, constant chemical potential, volume and temperature in an open system. GCMC in this sense was used to test the stability of a given configuration of particles. The final configuration from a MCMC simulation, which can exist as a stable, metastable or unstable system, was used as the initial configuration for a GCMC simulation. The chemical potential was set from previous MCMC simulations as well. The stability of a state generated in MCMC can be verified in GCMC simulation started that particular configuration. Labile configurations cannot be stabilized in the open system implied by GCMC.

3.3.3 Results and Discussion

Using GCMC and MCMC simulations we have studied adsorption of LJ nitrogen in the two-pore system constructed of overlapping spheres of $15.82\sigma_{\mathrm{FF}}$ diameter with two window sizes: $5\sigma_{\mathrm{FF}}$ and $10\sigma_{\mathrm{FF}}$. Adsorption isotherms for both systems are presented in Figure 3.26, along with the canonical isotherm obtained from MCMC and the GCMC isotherm for a single spherical pore of the same size from Ref. [1]. Adsorption amounts were normalized as dimensionless fluid density $\rho = N\sigma_{\mathrm{FF}}^3/V$, where V is the total volume of the system calculated with the internal diameter.

Figure 3.26a demonstrates a qualitative difference between the canonical isotherms for

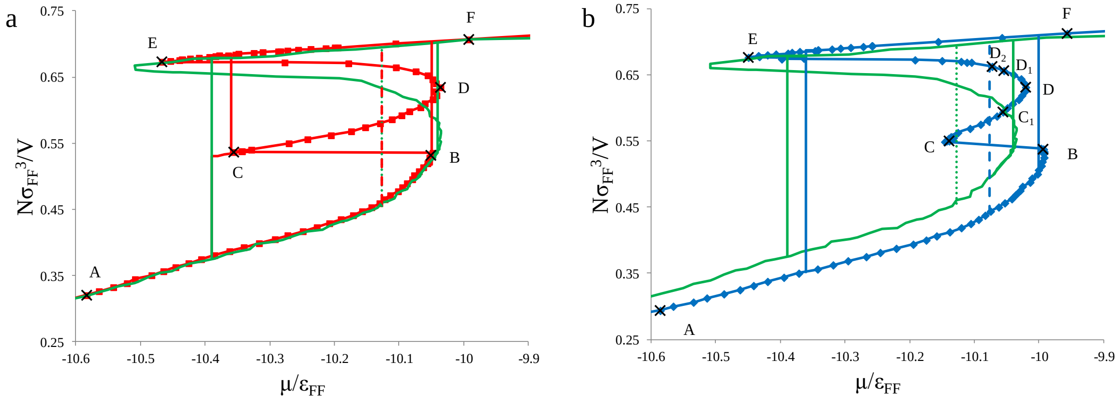


Figure 3.26: Isotherms calculated by MCMC (lines with points) and GCMC (solid lines without points) from simulations in the system with (a) $5\sigma_{\mathrm{FF}}$ window and (b) $10\sigma_{\mathrm{FF}}$ window. Green solid lines on both figures represents the adsorption isotherm for the single spherical pore of the same size from Ref. [1]. One of the desorption steps for the two-pore system in Fig. 3.26a coincides with the desorption transition for a single pore. Vertical dash lines indicate the positions of vapor-liquid equilibrium chemical potentials calculated for each system by applying the Maxwell rule to continuous canonical isotherms.

a single spherical pore and a two-pore system. The two-pore system isotherm has an intermediate loop, which is indicative of dissimilar phase behavior in the connected pores. The physical states of the two-pores system corresponding to the characteristic points of the isotherm are revealed from the snapshots in Figure 3.27. Line AB corresponds to growth of the film on the pore walls (vapor-like state), and does not differ from the analogous branch of the isotherm for a single pore. At the point B the film reaches the limit of its stability and capillary condensation takes place in one of the pores (point C). Thus, the fluid in one of the pores is in liquid-like state, while the other remains in the vapor-like state; this is clearly seen on the snapshot for the point C in Figure 3.27. Line CD presents a number of quasi-equilibrium states, similar to the one presented by point C. These states do not represent truly equilibrium configurations, but exist because of the stabilization by the gauge cell. At the point D, the second pore fills, except for a bubble in it; such states present till the point E, where the bubble disappears. Line EF, where both pores are in liquid-like state, does not differ from the corresponding part of the isotherm for a single pore.

The MCMC method allows one to construct a continuous function of $N(\mu)$, which can be thermodynamically integrated to determine the grand potential. The chemical potential at which VLE occurs can then be found where the grand potentials of the filled pore and pore with the adsorbed film are equal. We calculated the values of such equilibrium chemical potentials μ_e for each system by application of Maxwell's rule of equal areas, applied to the grand potential [51]

$$\oint_{\mu_\mathrm{e}} N \,\mathrm{d}\mu = 0 \tag{3.18}$$

Corresponding hypothetical VLE transitions are presented as vertical dash lines in Figure 3.26. These calculations reveal that for the small window size ($5\sigma_\mathrm{FF}$) the position of equilibrium transition does not deviate substantially from the one for the single sphere.

The adsorption branch of GCMC isotherm for the two-pore model with $5\sigma_\mathrm{FF}$ window, presented in Figure 3.26a, almost coincide with the isotherm for a single pore; the capillary condensation takes place at approximately the same chemical potential. However, the desorption branch of the GCMC isotherm has qualitatively different behavior than the one for a single pore. The two-steps desorption shows that emptying of the neighboring pores in the two-pore model does not take place simultaneously, but the cavitation events are subsequent. One of the desorption steps for the two-pore system in Figure 3.26a coincides with the desorption transition for a single pore. Intermediate states between two cavitation events correspond to the quasi-equilibrium states on the CD branch of MCMC isotherm. Overall, since for a two-pore system with a small window, we observe small deviations in equilibrium, condensation, and cavitation pressures, one should expect that the experimental behavior of the pore network with small windows should be similar to that in the system of independent spherical pores.

The MCMC isotherm for the two-pore system with the large window ($10\sigma_\mathrm{FF}$) is presented in Figure 3.26b, the snapshots for selected points on this isotherm are given in Figure 3.28. The MCMC isotherm has an intermediate loop, however this loop is much less pronounced than the one for the $5\sigma_\mathrm{FF}$ window. Therefore, the range of chemical potential corresponding to quasi-equilibrium states, when the fluid in one of the pores is in liquid-like state and the other remains in the vapor-like state (line CD), is narrower than for the system with small window.

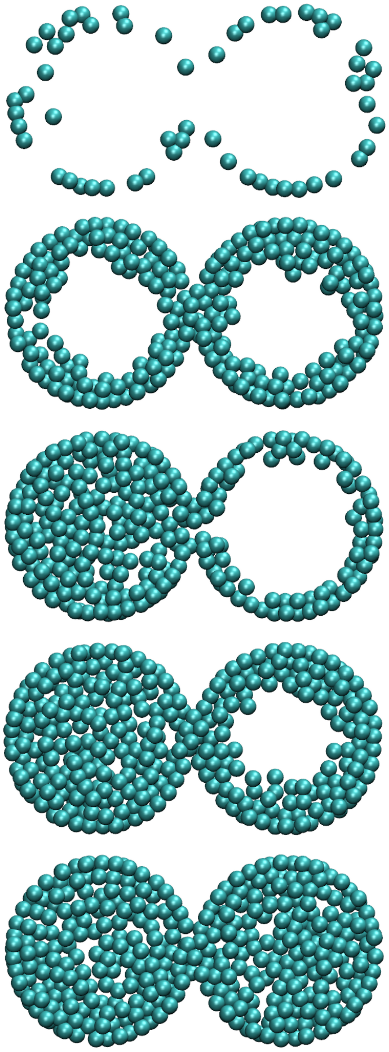


Figure 3.27: Snapshots of the molecules adsorbing in a system of two spheres with $5\sigma_{\rm FF}$ window. The number of molecules in the system and corresponding point on the isotherm in Figure 3.26a are (top to bottom): 1325 – A, 2200 – B, 2225 – C, 2625 – D and 2785 – E. Only molecules with centers located within 1σ slice parallel to the plane of the picture are shown.

For the system with the large window, a meniscus on the window between the pores in the region CD is revealed in the snapshots (see snapshots for points C and C_1 in Figure 3.28). The proximity of the limit of stability of these states (point C) to the condition of equilibrium in the one-pore system (dotted line in Figure 3.26b), suggests that desorption from the partially saturated state may occur near the equilibrium. This scenario was first reported in Ref. [35], as an explanation for the sub-steps on experimental scanning isotherms.

Snapshots for points C and C_1, D_1 and D_2 are presented in Figure 3.28 to reveal an interesting "flip-flop" behavior, when the pore in the vapor state changes with the pore in the liquid state (C and C_1), or the bubble "moves" from one pore to another (D_1 and D_2). Apparently, since the pores are equivalent, such flip-flop is not reflected in the thermodynamic state of the system.

Both the capillary condensation and cavitation transitions predicted by GCMC isotherm for the two-pore model with $10\sigma_{\mathrm{FF}}$ window (Figure 3.26b) are shifted to higher chemical potentials, relative to the isotherm for a single pore. The hypothetical equilibrium VLE transition, calculated from MCMC isotherm, using the Maxwell rule (3.18) is represented by the dash line, which is also shifted to higher chemical potentials. These observations suggest that experimental behavior of the pore network with large windows might deviate from that in the system of independent pores.

Another set of GCMC simulations were performed to test the stability of the quasi-equilibrium states branch (CDE in Figure 3.26) on the canonical ensemble MCMC isotherm [227]. In a single pore model, any states between the vapor and liquid spinodal points (B and E, respectively) are known to be labile states, and exist in the canonical ensemble on a single, backwards trajectory between the two points (i.e. the chemical potential on this trajectory decreases with the increase of number of molecules). However, the canonical isotherm of the two-sphere model presents three branches between the spinodals, two backwards (BC and DE) and one forward (CE). To test the stability of these intermediate states, we take canonical ensemble configurations, set them in an open system as starting configurations for GCMC simulations by removing the gauge cell and fixing the chemical potential to the value of the previous simulation. The results of this study are presented in Figure 3.29. We found a significant range of stable one-sphere filled states (e.g. points C and C_1), for

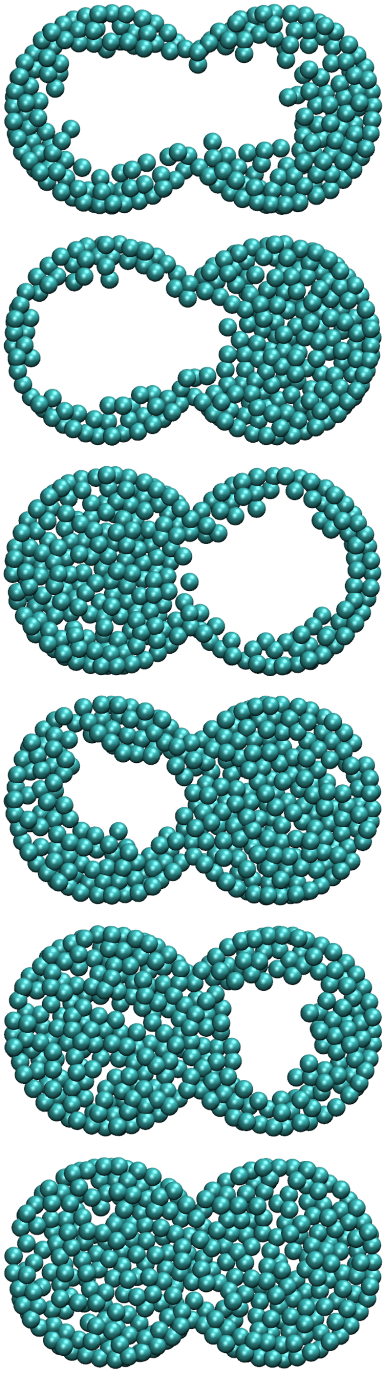


Figure 3.28: Snapshots of the molecules adsorbing in a system of two spheres with $10\sigma_{\mathrm{FF}}$ window. The number of molecules in the system and corresponding point in Figure 3.26b are: 2150 – B, 2200 – C, 2275 – C_1, 2625 – D_1, 2650 – D_2, and 2705 – E. Only molecules with centers located within 1σ slice parallel to the plane of the picture are shown.

both $10\sigma_{\mathrm{FF}}$ and $5\sigma_{\mathrm{FF}}$ systems on the forward-trajectory branch CD. Any system starting on the backward-trajectory branch DE were found to be unstable, as density fluctuations pushed the system to either a liquid-like or vapor-like state. There were no states observed on the MCMC isotherms between points B and C. It confirms that the states when the fluid in one of the pores is in liquid-like state, and in the other it is in the vapor-like state (i.e. a filled sphere in contact with an empty sphere via a meniscus) can be a stable, and further suggests the possibility of step-wise adsorption and desorption in such systems. It is important to note that although these intermediate states may be stable, that does not necessarily mean they will be observed in experiment. The incremental increase of external pressure during adsorption measurements (and corresponding decrease during desorption) would not permit access to the mid-density region we observe in simulation. However, careful scanning isotherms could hypothetically probe this region.

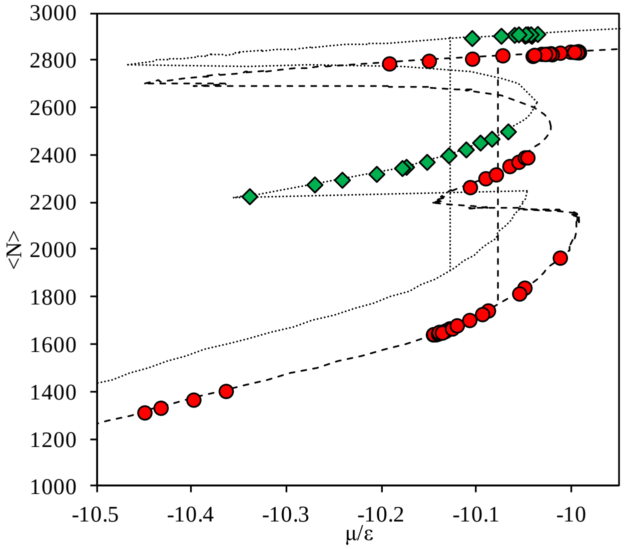


Figure 3.29: States stable in GCMC simulation. Heavy dash line is the MCMC canonical isotherm for the $10\sigma_{\mathrm{FF}}$ window, fine dash line for the $5\sigma_{\mathrm{FF}}$ window. Circles and diamonds ($10\sigma_{\mathrm{FF}}$ and $5\sigma_{\mathrm{FF}}$, respectively) represent the $\langle N \rangle$ from a GCMC run using configurations from MCMC simulations as the initial state.

Finally, to demonstrate the difference between desorption in the system with $5\sigma_{\mathrm{FF}}$ window and the system with $10\sigma_{\mathrm{FF}}$ window, we ran another series of GCMC simulations. We calculated the scanning desorption isotherms starting from the points obtained by MCMC within the CD region (see Figure 3.26). The stability of these points was revealed above, now we test the limits of stability. Starting from a configuration when the fluid in one pore is in liquid state and in the other is in vapor state, and sequentially decreasing the chemical potential, we monitor when the evaporation takes place in the filled pore. Results are presented in Figure 3.30, scanning desorption isotherms are displayed with solid black lines connecting branches of corresponding MCMC isotherms. Thus, for the system with $5\sigma_{\mathrm{FF}}$ window, emptying of the filled pore takes place close to the liquid-vapor spinodal, while for the system with $10\sigma_{\mathrm{FF}}$ window we should expect evaporation at substantially higher chemical potential, close to the equilibrium value for the single spherical pore. It confirms that the pores with large windows should evaporate in experiment at higher pressures than the pores with small windows.

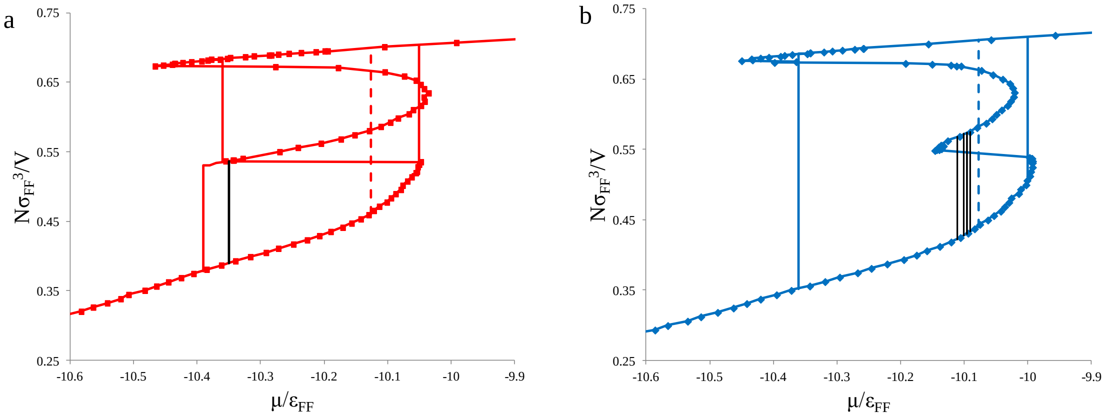


Figure 3.30: Scanning desorption isotherms for (a) $5\sigma_{\mathrm{FF}}$ window and (b) $10\sigma_{\mathrm{FF}}$ window. Scanning desorption isotherms are displayed with solid black lines connecting branches of corresponding gauge cell isotherms. Starting from a configuration when the fluid in one pore is in liquid state and in the other is in vapor state, and sequentially decreasing the chemical potential, we monitor the chemical potential at which evaporation takes place in the filled pore. Results for several runs are presented. Emptying of the filled pore in the system with $10\sigma_{\mathrm{FF}}$ window takes place at substantially higher chemical potential then in the system with $5\sigma_{\mathrm{FF}}$ window.

3.3.4 Conclusions

The adsorption and desorption of Lennard-Jones fluid in the system of two overlapping spherical pores was modeled using mesocanonical and grand canonical Monte Carlo simulations. The system under consideration differed from the classical ink-bottle pore model in that it did not have connecting channels, which provide an additional adsorption capacity. To reveal the role of the window size between the pores, we ran simulations for pores of the same diameter, but with two different sizes of the window. This was compared with published results of simulations for the single spherical pore of the same diameter and adsorption potential. For the system with the small window, the values of chemical potential at capillary condensation and evaporation predicted by GCMC simulations almost coincide with that of the single sphere of the same size. For the system with the large window, the positions of both capillary condensation and evaporation are noticeably shifted to higher values, reflecting the reduction of confinement effects.

Mesocanonical simulations allowed for trajectories of metastable and unstable states in the systems under consideration. In particular, we revealed points that correspond to the limit of stability of metastable configurations, when the fluid in one pore is in a liquid state and in the other is in a vapor state with the meniscus at the window. For the system with the small window, the chemical potential of such a configuration is close to that of the liquid-like spinodal obtained in GCMC calculations. This implies that the vapor pressure at which the capillary evaporation takes place in a system of overlapping spherical pores is close to the characteristic pressure of a single spherical pore, given a small degree of overlap.

For the system with the large window, the limit of stability of one-pore-filled configurations corresponds to a chemical potential substantially higher than that of the liquid-vapor spinodal, and closer to the true vapor-liquid equilibrium determined by the Maxwell rule. This observation suggests that in a partially filled pore network with relatively large overlap of pores, the evaporation along a scanning desorption isotherm may proceed at least partially in a near-equilibrium fashion. This conclusion complies with the earlier experimental studies of scanning desorption isotherms in FDU-1 materials with cage-like pore networks [194].

The substantial difference between the mechanism of emptying the system with the small window and the large one was also revealed by simulating scanning desorption isotherms. Simulations of desorption starting from a stable mid-density states revealed that pores with small window empty close to the vapor-liquid spinodal, while the pores with large window empty at higher chemical-potential, close to the equilibrium value of the single spherical pore.

Overall, we can make an analogy between the considered model and the classical ink-bottle pore: for both models the capillary condensation pressure is determined by the diameter of the main cavity, but the mechanism of desorption is controlled by the diameter of the necks/windows. The results can be used for understanding desorption mechanism in ordered materials with cage-like pores [223, 226] or in some disordered materials [189, 228, 229]. The presented model of interconnected pores can be implemented for DFT calculations of adsorption/desorption isotherms, to make a kernel for materials characterization.

Chapter 4

Modeling Confined Polymers: New Method and its Applications

4.1 Calculation of Chemical Potentials of Chain Molecules by the Incremental Gauge Cell Method

4.1.1 Introduction

The determination of the chemical potential of chain molecules in Monte Carlo (MC) simulations is a long studied and difficult problem. Chemical potential in homogeneous simple fluids is readily calculated using Widom's particle insertion technique [8]. However, for moderately complex molecules (such as chains of 3 or more monomers), the probabilities of non-overlapping insertions are very low, resulting in long simulations with large statistical errors. Application of the Rosenbluth and Rosenbluth configurational bias [92] to test-chain insertions was used by Harris and Rice [230], and then by Siepmann [93] to calculate the chemical potential of chains on a lattice. Biased insertions were later generalized to continuously deforming molecules by Frenkel et al. [72] and later by de Pablo et al. [90]. Insertion of configurationally biased molecules overcomes the sampling issues mentioned above, however the likelihood of an acceptable biased insertion in a dense fluid decreases greatly when chain length n exceeds 20. Various computational strategies were designed to improve this limit [98, 123, 231], most notably the pruned-enriched Rosenbluth method (PERM) of Grassberger [85]. This method involves generating populations of configurations, then periodically 'pruning' them by discarding the most improbable, and 'enriching' by duplicating the most favorable. Using this method one can obtain an accurate measure of the chemical potential for very long chains; however, this approach is only valid for so called 'static' Monte Carlo [85] (i.e., only one molecule's conformation is sampled, and the simulation does not involve a Markov chain). An attempt to extend PERM to conventional 'dynamic' Markov chain MC did not show an improvement over the simple Rosenbluth insertions [119]. Another method of overcoming this finite chain-length drawback was proposed by Kumar et al. [11] Instead of inserting the entire chain in one step, this method, called the modified Widom (MW) method, is based on "incrementing" the chain (testing insertions of monomers added to the chain end) to calculate the incremental chemical potential of the monomer μ_{incr}. The chemical potential of the chain is then the sum of the incremental chemical potentials of the monomers that the chain is comprised of. The obvious drawback of the MW method is that rigorously, one requires n MC simulations for a single chain of n monomers. Other methods for the calculation of chemical potential include thermodynamic integration over the chain with variable interaction potentials [77], reverse Widom method via staged particle deletions [134], extension of the expanded ensemble to chains of variable length [123], grand canonical simulation of configurationally biased chains [99], and a hybrid method of biased Widom insertions with the scanning method [131]. The technique proposed in this work is based on the "incremental" strategy [11], as we found this approach the most suitable for modeling nanoconfined polymeric systems that we are interested in. To this end, the MW method served as the reference technique for validation and justification of the proposed method.

The dependence of the incremental chemical potential μ_{incr} on the chain length n is a much-debated topic [80–83, 232]. It was originally asserted [11] that μ_{incr} depends on n only for short chains, $n < 5$, and is essentially independent of the chain length for long chains, but this conclusion was made within computational constraints which allowed for calculations of chains shorter than 30 monomers. This conclusion of the independence of the incremental chemical potential on the chain length became known as the "chain increment ansatz" [233]. Kumar later refined [80] the initial assumption of μ_{incr} independence on n to include only "coil" polymers that are well above their θ temperature, where the chain configuration may be reasonably approximated by a self-avoiding random walk. The original assumption [11] was concluded to be invalid for *globular* polymers, i.e. chains below their θ temperature. At the θ temperature, the attractive and repulsive parts of monomer-monomer interactions effectively cancel out, and the real chain exhibits some characteristics similar to a free Gaussian (ideal) chain. Above the θ temperature, the chain increment ansatz says that the chemical potential of a coil polymer can essentially be calculated in a single simulation, by trial monomer insertions at the end of one chain in the system. The resulting incremental chemical potential can then be multiplied by the number of monomers in the chain and corrected for short-chain behavior, to obtain the total chemical potential of the chain in very few simulations. The ansatz was tested for free chains up to 30 monomers using the modified Widom method [11], for long chains (~100 monomers) by using PERM [115], and thoroughly with fluids of short n-alkanes ($n = 6$ to 16) at high densities by comparing the modified Widom method with Rosenbluth insertions [81].

In this work, we extend the gauge cell MC method [9, 10] introduced for calculating chemical potentials in dense and confined simple fluids to polymeric systems via the incrementing strategy. In the gauge cell method, the simulation system is constructed as follows: a system of interest, or sample cell, is placed in chemical equilibrium with a reference control volume, or gauge cell. The gauge cell brings two advantages: it serves as a meter of the chemical potential, and its limited volume restricts density fluctuations in the sample cell. This schematic corresponds to a mesocanonical ensemble [49]: it is equivalent to the canonical ensemble in the target cell if the gauge cell volume is infinitely small, and is equivalent to the grand canonical ensemble if the gauge cell volume is infinitely large. The gauge cell method finds numerous applications in simulation studies of phase transitions in strongly heterogeneous systems. It has been employed in studies of capillary condensation in various geometries [50–52], liquid bridge [53] and droplet nucleation [54], bubble cavitation in metastable liquid [1, 55], adsorption deformation [56] and the segregation of surfactants [57]. The gauge cell method has been extended from its original formulation [9] to extremely small confinements with the ideal gas gauge cell (IGGC) method [10], mixtures with multi-component gauge cell method [49], coupling with Widom insertions [234], configurational bias [59, 60], and simplified insertions using explicit chemical potential [61].

To extend the gauge cell method to linear homopolymers, we begin by considering a system cell that contains a polymeric fluid of one or more chains of identical monomers, and a gauge cell that contains a fluid of ideal non-bonded monomers. Standard MC moves are used to sample state space in the system cell; this work includes monomer displacement and reptation of the chain (but can also include configurationally-biased regrowth or any other canonical ensemble move to equilibrate the chain in the system cell), particle insertion from the gauge into the sample cell, and particle removal from the sample cell to the gauge cell. On an attempted insertion, the particle removed from the gauge cell is attached to an end of the polymer chain. Correspondingly, on an attempt of removal, a terminal monomer is detached from the polymer chain and placed into the gauge cell. Thus, in contrast with the Widom method, the gauge cell method deals with real insertion and removal of particles. Particle insertions and removals allow the chain to grow or shrink according to given equilibrium conditions and allowed thermal fluctuations. Since the equation of state of the reference ideal gas of monomers in the gauge cell is known, we can also calculate the chemical potential of the chain in the sample cell. Preliminary results and a short derivation of the incremental gauge cell method for a fluid of chains with identical lengths was presented in Ref. [207], and will be expanded and generalized in this work.

The rest of the section is structured as follows: in Section 4.1.2, we formulate the gauge cell method for a mixture of chains of various lengths, and give its statistical mechanical derivation. In Section 4.1.3, we discuss its practical implementation. Simulation details are provided in Section 4.1.4. In Section 4.1.5, we apply the suggested method to a single isolated polymeric chain of LJ particles connected by harmonic spring bonds and compare our results to the literature data. Results obtained for single chains in a nano-confinement are presented in Sec 4.1.6 and 4.1.7. A comparison of the computational efficiency of the incremental gauge cell method and the MW method is presented in Section 4.1.8. Finally, conclusions and a critical discussion of the method are presented in Section 4.1.9.

4.1.2 The Incremental Gauge Cell Method

Definition of the incremental chemical potential

To derive the statistical mechanical basis for the incremental gauge cell method, we start from the definitions of the chemical potential and the incremental chemical potential in a general case of a mixture of homopolymer chains of various lengths. Consider a three-dimensional canonical system of constant volume V at temperature T. Let N_i be the number of chains of length i (that is, N_1 refers to the number of monomers, N_2 to the number of dimers, and so on). The mixture is then described by the set of all chains, $\{N_\alpha\} = \{N_1, N_2, \cdots, N_i, N_{i+1}, \cdots\}$. The total number of chains is

$$M = \sum_{i=1}^{\infty} N_i \tag{4.1}$$

Similarly, the total number of monomers is

$$m = \sum_{i=1}^{\infty} iN_i \tag{4.2}$$

A general description of the canonical partition function for such a system, with no restrictions for intra-molecular connectivity, is

$$Q(\{N_\alpha\}, V, T) = \frac{1}{\Lambda^{3m} \prod_{i=1}^{\infty}(N_i!)} \int\limits_V d\mathbf{r}^m \exp\left[-\frac{\Phi(\mathbf{r}^m)}{k_\mathrm{B}T}\right] \tag{4.3}$$

where Λ is the thermal de Broglie wavelength of the monomers, $\mathbf{r}^m$ is the set of m position vectors describing the location of each monomer, and the total potential energy ϕ is the sum of monomer-monomer interactions, intramolecular interactions (such as bond-stretching, angle-bending, or torsion potentials), and external potentials such as an adsorption potential. The Helmholtz free energy of such a system can be described in terms of the canonical partition function,

$$F(\{N_\alpha\}, V, T) = -k_\mathrm{B}T \ln Q(\{N_\alpha\}, V, T) \tag{4.4}$$

If the system is large, and its Helmholtz free energy F is continuous and differentiated function of N, the chemical potential of chains of length ν at constant $\{N_{\alpha\neq\nu}\}$, volume and temperature is

$$\mu_\nu(\{N_\alpha\}) = \left.\frac{\partial F}{\partial N_\nu}\right|_{\{N_{\alpha\neq\nu}\},V,T} \tag{4.5}$$

Following the approach of Widom [8], this expression can be approximated as a finite difference, and the chemical potential can be defined as the difference of the free energy of a system with $\{N_\nu + 1\}$ and N_ν molecules,

$$\mu_\nu(\{N_\alpha\}) = F(\{N_1, \cdots, N_\nu + 1, \cdots\}, V, T) - F(\{N_1, \cdots, N_\nu, \cdots\}, V, T \tag{4.6a}$$

$$\mu_\nu(\{N_\alpha\}) = -k_\mathrm{B} T \ln \left[\frac{Q(\{N_1, \cdots, N_\nu + 1, \cdots\}, V, T)}{Q(\{N_1, \cdots, N_\nu, \cdots\}, V, T)} \right] \tag{4.6b}$$

$$= -k_\mathrm{B} T \ln \left[\frac{V^\nu}{\Lambda^{3\nu}\,(N_\nu + 1)} \right] - k_\mathrm{B} T \ln \left[\frac{1}{V^\nu} \int\limits_{(V^\nu)} d\mathbf{r}^\nu_\mathrm{ins} \langle \exp\left[-\phi(\mathbf{r}^\nu_\mathrm{ins}, \mathbf{r}^m)/k_\mathrm{B}T\right] \rangle_{\{N_1, \cdots, N_\nu, \cdots\}} \right] \tag{4.6c}$$

Correspondingly, the chemical potential of $(\nu + 1)$-mers is

$$\mu_{\nu+1}(\{N_\alpha\}) = F(\{N_1, \cdots, N_{\nu+1} + 1, \cdots\}, V, T) - F(\{N_1, \cdots, N_{\nu+1}, \cdots\}, V, T) \tag{4.7a}$$

$$\mu_{\nu+1}(\{N_\alpha\}) = -k_\mathrm{B} T \ln \left[\frac{Q(\{N_1, \cdots, N_{\nu+1} + 1, \cdots\}, V, T)}{Q(\{N_1, \cdots, N_\nu, \cdots\}, V, T)} \right] \tag{4.7b}$$

$$\begin{aligned} =& - k_\mathrm{B} T \ln \left[\frac{V^{\nu+1}}{\Lambda^{3\nu+3}\,(N_{\nu+1} + 1)} \right] \\ &- k_\mathrm{B} T \ln \left[\frac{1}{V^{\nu+1}} \int\limits_{(V^{\nu+1})} d\mathbf{r}^{\nu+1}_\mathrm{ins} \langle \exp\left[-\phi(\mathbf{r}^{\nu+1}_\mathrm{ins}, \mathbf{r}^m)/k_\mathrm{B}T\right] \rangle_{\{N_1, \cdots, N_\nu, \cdots\}} \right] \end{aligned} \tag{4.7c}$$

In Equations (4.6c) and (4.7c), the RHS is divided into the ideal and excess terms. As such, the excess chemical potential of a ν-mer can be determined by random insertion of ν monomers in volume V, and averaging the Boltzmann probability that these ν monomers may be bound in one chain with given bonded and non-bonded interaction potentials. If this approach were practical, it would represent a direct extension of the Widom insertion method to chain molecules.

We can now define the incremental chemical potential $\mu_\mathrm{inc}(\nu, \{N_\alpha\})$ as the difference of chemical potentials for the chains of $\nu + 1$ monomers and ν monomers, with all else held equal, by subtracting Equation (4.6a) from (4.7a):

$$\begin{aligned} &\mu_\mathrm{inc}(\nu, \{N_\alpha\}) \equiv \mu_{\nu+1}(\{N_\alpha\}) - \mu_\nu(\{N_\alpha\}) = \\ &\qquad F(\{N_1, \cdots, N_{\nu+1} + 1, \cdots\}, V, T) - F(\{N_1, \cdots, N_\nu + 1, \cdots\}, V, T) \end{aligned} \tag{4.8}$$

given $\nu > 0$. Equation (4.8) is the foundation of the MW method [11] and represents the incremental chemical potential of a ν-mer in a mixture of $\{N_\alpha\}$ chains. Determination of the chain chemical potential $\mu_\nu(\{N_\alpha\})$ is simply a matter of summing all incremental values. As described in Ref. [232], μ_{inc} is a well-defined thermodynamic quantity. Analogously to the Widom insertion method, Equation (4.4) is substituted into (4.8) to obtain a ratio of partition functions, as the difference between Equations (4.6c) and (4.7c),

$$\begin{aligned}\mu_{\text{inc}}(\nu, \{N_\alpha\}) &= -k_{\text{B}}T \ln \left[\frac{Q(\{N_1, \cdots, N_{\nu+1}+1, \cdots\}, V, T)}{Q(\{N_1, \cdots, N_\nu + 1, \cdots\}, V, T)} \right] \\ = &- k_{\text{B}}T \ln \left[\frac{V\,(N_\nu + 1)}{\Lambda^3\,(N_{\nu+1}+1)} \right] \\ &- k_{\text{B}}T \ln \left[\frac{1}{V} \int\limits_{(V)} d\mathbf{r}_{\text{ins}} \langle \exp\left[-\phi(\mathbf{r}_{\text{ins}}(\nu+1), \mathbf{r}^{m+\nu})/k_{\text{B}}T \right] \rangle_{\{N_1, \cdots, N_\nu+1, \cdots\}} \right]\end{aligned} \tag{4.9}$$

given $\nu > 0$, and where $\phi(\mathbf{r}_{\text{ins}}(\nu+1), \mathbf{r}^{m+\nu})$ is the total interaction potential energy of the $\nu+1$ trial monomer, located at $\mathbf{r}_{\text{ins}}$, with the rest of the system. This trial monomer is positioned by incrementing a ν-mer chain by one monomer. The angle brackets denote the canonical average over all particle positions $\mathbf{r}^{m+\nu}$. To make the calculations more efficient, we assume that monomer $\nu+1$ can only be inserted in a limited volume V_{ins} around monomer ν. As such, we assume infinite bond energy between the monomers if the bond length is too large or too small. This assumption is reasonable, since if the bond energy is large, this configuration will not contribute to the Boltzmann factor that is being sampled. By averaging only over the volume V_{ins}, rather than over the entire volume V, Equation (4.9) then becomes

$$\begin{aligned}\mu_{\text{inc}}(\nu, \{N_\alpha\}) &= -k_{\text{B}}T \ln \left[\frac{V\,(N_\nu + 1)}{\Lambda^3\,(N_{\nu+1}+1)} \right] \\ &- k_{\text{B}}T \ln \left[\frac{V_{\text{ins}}}{V} \langle \exp\left[-\phi(\mathbf{r}_{\text{ins}}(\nu+1), \mathbf{r}^{m+\nu})/k_{\text{B}}T \right] \rangle_{\{N_1, \cdots, N_\nu+1, \cdots\}, V_{\text{ins}}} \right] \\ &= \mu_{\text{inc}}^{\text{id}}(\nu, \{N_\alpha\}) + \mu_{\text{inc}}^{\text{ex}}(\nu, \{N_\alpha\})\end{aligned} \tag{4.10}$$

Here, we defined the ideal $\mu_{\text{inc}}^{\text{id}}(\nu, \{N_\alpha\})$ and excess $\mu_{\text{inc}}^{\text{ex}}(\nu, \{N_\alpha\})$ contributions into the incremental chemical potential. The excess incremental chemical potential $\mu_{\text{inc}}^{\text{ex}}(\nu, \{N_\alpha\})$

defined in Equation (4.10) can be calculated by inserting a trial monomer in the insertion volume V_{ins} at the chain end and averaging the respective Boltzmann probability. This constitutes the rigorous basis for the MW method [11] that is used for comparison with the incremental gauge cell method described below.

The case of $\nu = 0$ is determined in a similar fashion, with the ratio in Equation (4.9) equal to $Q(\{N_1 + 1, \cdots\}, V, T)/Q(\{N_\alpha\}, V, T)$. It represents the chemical potential of a monomer inserted into the mixture of $\{N_\alpha\}$ chains, $\mu_0(\{N_\alpha\}) = \mu_{\text{inc}}(0, \{N_\alpha\})$, and is expressed as

$$\begin{aligned}\mu_0(\{N_\alpha\}) =& - k_{\text{B}}T \ln \left[\frac{V}{\Lambda^3(N_1+1)}\right] - k_{\text{B}}T \ln \left[\langle \exp\left[-\phi(\mathbf{r}_0, \mathbf{r}^m)/k_{\text{B}}T\right]\rangle_{\{N_1,\cdots,N_\nu,\cdots\},V}\right] \\ =&\mu_0^{\text{id}}(\{N_\alpha\}) + \mu_0^{\text{ex}}(\{N_\alpha\})\end{aligned} \tag{4.11}$$

where $\mathbf{r}_0$ is the position vector of a trial monomer in the system. Because the first monomer of a chain has no bond potential, the average of Equation (4.10) is calculated over the entire system volume V, and expression similar to the standard Widom insertion is found. By summation of the incremental chemical potentials, and grouping the terms of Equations (4.10) and (4.11), we obtain an expression for the chemical potential of a ν-mer:

$$\begin{aligned}\mu_\nu(\{N_\alpha\}) = -k_{\text{B}}T \ln \left[\frac{V^\nu}{\Lambda^{3\nu}(N_\nu+1)}\right] - k_{\text{B}}T \ln \left[\langle \exp\left[-\phi(\mathbf{r}_0, \mathbf{r}^m)/k_{\text{B}}T\right]\rangle_{\{N_\alpha\},V}\right] - \\ k_{\text{B}}T \sum_{i=1}^{\nu-1} \ln \left[\frac{V_{\text{ins}}}{V} \langle \exp\left[-\phi(\mathbf{r}_{\text{ins}}(i+1), \mathbf{r}^{m+i})/k_{\text{B}}T\right]\rangle_{\{N_1,\cdots,N_i+1,\cdots\},V_{\text{ins}}}\right]\end{aligned} \tag{4.12}$$

where $\mathbf{r}_{\text{ins}}(i+1)$ is position of the $i+1$ trial particle. The first term is the reference state for the ν-mer component of the mixture, an ideal gas of non-interacting ν-mer chains with no non-bonded intramolecular potential. The second term is the excess chemical potential of inserting a single monomer into the system (exactly the equation for the Widom method); the last term is the sum of the excess incremental chemical potentials starting from the previously inserted monomer. Note that the volume ratio represents a correction for the insertion volume being different from the system volume. Upon summation over i, the ideal term in (4.10) gives the ideal part of the chain chemical potential. The incremental ratios of $\mu_{\text{inc}}^{\text{id}}$ in Equation (4.10) cancel in the summation, with only $(N_\nu + 1)$ remaining in the denominator. Also note that although V_{ins} appears explicitly in Equation (4.12), there is no dependence on it, as long as it is chosen large enough to sample all non-negligible conformations and small enough to allow adequate sampling. Equation (4.12) can be written in terms of ideal and excess chemical potentials,

$$\mu_\nu(\{N_\alpha\}) = \mu_\nu^{\text{id}} + \mu_\nu^{\text{ex}} = \mu_\nu^{\text{id}} + \mu_0^{\text{ex}}(\{N_\alpha\}) + \sum_{i=1}^{\nu-1} \mu_{\text{inc}}^{\text{ex}}(i, \{N_\alpha\}) \tag{4.13}$$

It is worth noticing that the excess incremental chemical potential $\mu_{\text{inc}}^{\text{ex}}(i, \{N_\alpha\})$ depends on the composition $\{N_\alpha\}$, but it does not depend on the length ν of the inserted chain. Thus, the incremental insertion of the longest chain present in the mixture (that is, calculating the canonical average in Equation (4.10) for each monomer of the longest chain) yields all incremental chemical potential values needed for calculating the chemical potentials of remaining chains of any length. The chain increment ansatz involves the excess incremental chemical potential defined by Equation (4.10) and yields that $\mu_{\text{inc}}^{\text{ex}}(\nu, \{N_\alpha\}) \approxeq \mu_{\text{inc}}^{\text{ex}}(\nu + 1, \{N_\alpha\})$ for $\nu \gg 1$.

For single chain systems considered below, Equation (4.10) reduces to

$$\mu_{\text{inc}}(\nu) = -k_{\text{B}}T \ln \left[\frac{V_{\text{ins}}}{\Lambda^3} \langle \exp\left[-\phi\left(\mathbf{r}_{\text{ins}}(\nu+1)/k_{\text{B}}T\right)\right] \rangle_{\{N_1,\cdots,N_\nu+1,\cdots\},V_{\text{ins}}} \right] \tag{4.10$'$}$$

In examples shown below, we report the values of μ_{inc} given by Equation (4.10$'$) with Λ set to unity for the sake of comparison to literature results.

Alternate and extended derivations for the incremental chemical potential can be found in Appendix A.

Foundations of the gauge cell method

The gauge cell method was originally introduced as a computational method to measure chemical potential in dense fluids confined by external potentials. Use of the gauge cell brings several advantages to the simulation. First, it serves as a meter of chemical potential of the fluid in the main cell. Second, its finite volume suppresses fluctuations and allows the stabilization of metastable and unstable states in the sample cell. And third, it's computationally more efficient than Widom trial insertions, as the exchange of particles from gauge cell to sample cell and back facilitates mixing of the particles in the sample cell. The simulation scheme is constructed as follows: a sample cell of volume V containing the mixture of $\{N_\alpha\}$ chains is placed in contact with a gauge cell of volume V_g, and both are immersed in a heat bath of temperature T. Exchange of particles between the main and gauge cells is permitted by allowing the ends of a designated "test" chain to accept or send monomers to gauge cell. Note the test chain is in addition to the existing mixture $\{N_\alpha\}$. The total number of particles between the test chain and the gauge cell $n_\Sigma = n_\mathrm{t} + n_\mathrm{g}$ is constant. Thermodynamically, equilibration of this construction corresponds to the minimization of the total free energy of the test chain and the gauge cell,

$$F_\Sigma(n_\Sigma, \{N_\alpha\}, V, V_\mathrm{g}, T) = F_t(n_\mathrm{t}, \{N_\alpha\}, V, T) + F_\mathrm{g}(n_\mathrm{g}, V_\mathrm{g}, T) \Rightarrow \min \tag{4.14}$$

where F_t is the free energy of the test chain, F_g is the free energy of the fluid in the gauge cell, and F_Σ is their sum. The minimization implies equality of the partial derivatives,

$$\left.\frac{\partial F_\mathrm{t}}{\partial n_\mathrm{t}}\right|_{\{N_\alpha\},V,T} = \left.\frac{\partial F_\mathrm{g}}{\partial n_\mathrm{g}}\right|_{V_\mathrm{g},T} \tag{4.15}$$

The finite difference approximation of the left hand side of Equation (4.15) is in fact the incremental chemical potential of the test chain, as defined in Equation (4.8), and is equal to the chemical potential of gauge,

$$F_t(n_\mathrm{t}+1, \{N_\alpha\}, V, T) - F_t(n_\mathrm{t}, \{N_\alpha\}, V, T) = \left.\frac{\partial F_\mathrm{g}}{\partial n_\mathrm{g}}\right|_{V_\mathrm{g},T} = \mu_\mathrm{inc}(n_\mathrm{t}, \{N_\alpha\}) = \mu_\mathrm{g}(n_\mathrm{g}) \tag{4.16}$$

This is the basis of the so called mean density gauge cell (MDGC) method [9]. After an equilibrium distribution is reached, the average number of monomers $\overline{n_\mathrm{g}}$ in gauge cell can be used to compute the incremental chemical potential of a chain of length $n_\Sigma - \overline{n_\mathrm{g}}$, assuming the equation of state of the reference fluid in the gauge cell is known. It is important to note that Equations (4.15), (4.16) are an approximation, which is only valid if the numbers of molecules of both cell are large enough [10]. Strictly, the chemical potentials can be calculated from the probabilities of observing a specific n_{g} on any given step. This ideology was implemented in the ideal gas gauge cell (IGGC) version of the gauge method [10], where the reference fluid was chosen to be an ideal gas.

To extend the IGGC formalism to chain molecules, we begin with the system defined above, a system $\{N_\alpha\}$ chains in volume V, with an additional single test chain of n_{t} monomers present. A gauge cell containing a fluid of n_{g} monomers with volume of V_{g} is allowed to exchange particles with the test chain only. Similar restrictions as the original gauge cell are applied: $n_\Sigma = n_{\mathrm{t}} + n_{\mathrm{g}}$, V, V_{g} and T are all held constant. Given some number of monomers in the gauge, the probability to observe a test chain of n_{t} and gauge cell containing $n_{\mathrm{g}} = n_\Sigma - n_{\mathrm{t}}$ is proportional to,

$$P_{n_{\mathrm{t}}} = P_{n_{\mathrm{g}}} \propto \exp\left\{-\frac{1}{k_{\mathrm{B}}T}\left[F_{\mathrm{t}}(n_{\mathrm{t}}, \{N_\alpha\}, V, T) + F_{\mathrm{g}}(n_{\mathrm{g}}, V_{\mathrm{g}}, T)\right]\right\} \tag{4.17}$$

Similarly, if the test chain is incremented by one monomer, the probability to observe such a system is proportional to,

$$P_{n_{\mathrm{t}}+1} = P_{n_{\mathrm{g}}-1} \propto \exp\left\{-\frac{1}{k_{\mathrm{B}}T}\left[F_{\mathrm{t}}(n_{\mathrm{t}}+1, \{N_\alpha\}, V, T) + F_{\mathrm{g}}(n_{\mathrm{g}}-1, V_{\mathrm{g}}, T)\right]\right\} \tag{4.18}$$

By taking the ratio of Equations (4.17) and (4.18), the incremental chemical potential (as defined in Equations (4.8) and (4.16)) emerges,

$$\frac{P_{n_{\mathrm{t}}+1}}{P_{n_{\mathrm{t}}}} = \frac{P_{n_{\mathrm{g}}-1}}{P_{n_{\mathrm{g}}}} \propto \exp\left\{-\frac{1}{k_{\mathrm{B}}T}\left[\mu_{\mathrm{inc}}(n_{\mathrm{t}}) + \mu_{\mathrm{g}}(n_{\mathrm{g}}-1)\right]\right\} \tag{4.19}$$

where μ_{g} is the chemical potential of the gauge cell, defined traditionally as $F_{\mathrm{g}}(n_{\mathrm{g}}) - F_{\mathrm{g}}(n_{\mathrm{g}} - 1)$. Rearranging in terms of incremental chemical potential, we obtain:

$$\mu_{\mathrm{inc}}(n_{\mathrm{t}}) = \mu_{\mathrm{g}}(n_{\mathrm{g}}-1) + k_{\mathrm{B}}T \ln\left(\frac{P_{n_{\mathrm{g}}}}{P_{n_{\mathrm{g}}-1}}\right) \tag{4.20}$$

Equation (4.20) is the basis for calculation of the incremental chemical potential via the gauge cell method. It relates the incremental chemical potential of the test chain to the chemical potential of gauge cell. Because $n_{\mathrm{t}}+n_{\mathrm{g}}$ is finite, the probabilities $P_{n_{\mathrm{g}}}$ and $P_{n_{\mathrm{g}}-1}$ can be recorded during simulation. So far, no assumption has been made regarding the nature of the gauge cell fluid. The most useful choice is an ideal gas. The chemical potential of an ideal gas (as defined with a finite difference) in the gauge cell is

$$\mu_{\mathrm{ig}}(i) = -k_{\mathrm{B}}T \ln\left(\frac{V}{\Lambda^3(i+1)}\right) \tag{4.21}$$

Substituting Equation (4.21) into (4.20) and simplifying, we obtain

$$\mu_{\mathrm{inc}}(n_{\mathrm{t}}, \{N_\alpha\}) = -k_{\mathrm{B}}T \ln\left(\frac{V}{\Lambda^3 n_{\mathrm{g}}}\right) + k_{\mathrm{B}}T \ln\left(\frac{P_{n_{\mathrm{g}}}}{P_{n_{\mathrm{g}}-1}}\right) \tag{4.22}$$

This is the main equation used for calculating μ_{inc} via the IGGC method. Correspondingly, if n_{t} is large enough, Equation (4.22) simplifies to the equation for the MDGC method,

$$\mu_{\mathrm{inc}}(n_{\mathrm{t}}, \{N_\alpha\}) = k_{\mathrm{B}}T \ln\left(\frac{\rho_{\mathrm{g}}}{\Lambda^3}\right) \tag{4.23}$$

where ρ_{g} is the monomer density in the gauge cell. In practice, they are calculated simultaneously; the IGGC yielding several points of decreasing statistical accuracy of $\mu_{\mathrm{inc}}(n_{\mathrm{t}})$, and MDGC yielding one point per simulation. In both cases, to quantitatively compare with Equation (4.10′), the term $1/\Lambda^3$ is moved to the reference chemical potential.

Similarly to the grand canonical MC simulation, the probability of attempting an insertion is equal to that of a removal, in order to preserve the symmetry of the underlying Markov chain. Acceptance probabilities for insertions and removals are derived from the detailed balance condition. The probability of accepting an insertion to a randomly selected end of the test chain is

$$\mathrm{acc}(n_{\mathrm{t}} \to n_{\mathrm{t}}+1) = \min\left\{1, \frac{V_{\mathrm{ins}} n_{\mathrm{g}}}{V_{\mathrm{g}}} \exp\left[-\frac{\phi_{\mathrm{ins}}(n_{\mathrm{t}}+1)}{k_{\mathrm{B}}T}\right]\right\} \tag{4.24}$$

where $\phi_{\mathrm{ins}}(n_{\mathrm{t}}+1)$ is the potential energy of the trial inserted monomer. There are two notable differences between Equation 4.24 and its corresponding simple-fluid equivalent [10]. First, the volume term in the numerator is the only volume in which the $n_{\mathrm{t}}+1$ monomer is allowed to be placed (V_{ins}), rather than the entire system volume (V). Second, there is no

N term in the numerator, as the number of molecules in the system does not change due to a monomer insertion (unless it is the first monomer of the chain). The corresponding equation for removal is

$$\mathrm{acc}(n_\mathrm{t} \to n_\mathrm{t} - 1) = \min\left\{1, \frac{V_\mathrm{g}}{V_\mathrm{ins}(n_\mathrm{g} + 1)} \exp\left[-\frac{\phi_\mathrm{rem}(n_\mathrm{t} - 1)}{k_\mathrm{B}T}\right]\right\} \tag{4.25}$$

where $\phi_\mathrm{rem}(n_\mathrm{t} - 1)$ is the potential energy of the terminal monomer selected for trial removal. Similarly to what is described in Equation (4.12), a special case exists for $n_\mathrm{t} = 0$. The insertion for $(0 \to 1)$ and removal of $(1 \to 0)$ monomers of the test chain are governed by the original gauge cell acceptance equations [10]. Together, these equations sample the mesocanonical ensemble distribution of particles for the test chain in contact with a gauge cell as described above.

4.1.3 Implementation

Practical implementation of the incremental gauge cell method is straightforward. Starting from a validated canonical ensemble MC algorithm, only one extra subroutine (particle exchange with the gauge cell) and several variables (n_g and V_g) need to be added. In the exchange routine, removal or insertion is selected with equal probability, as is which end of the chain to exchange with. For removal, the potential energy of the terminal monomer on the chain is calculated, and the probability of its removal is calculated using Equation (4.25) and accepted if greater than a random number on $[0, 1)$. Likewise, if insertion is selected, a new particle is generated at random in the insertion volume at the end of the chain, and its potential energy calculated. The probability of accepting it into the configuration of the main cell is calculated using Equation (4.24) and accepted similarly to particle removal. If the exchange move is accepted, the global potential energy is updated and the particle added (or removed) to the array of current coordinates. Care must be taken for the limiting cases of $n_\mathrm{t} = 0$ or 1; in these situations, exchange probabilities are calculated using the original gauge cell equations [10].

It is important to note that the particles contained in the gauge cell do not need to have coordinates or energy recorded because we assumed the gauge fluid is ideal gas. Therefore, we only have to additionally track the current number of particles in the gauge and the volume of the gauge cell. The selection of the gauge cell volume is crucial to obtain results for a desired value of $\mu_{\mathrm{inc}}(n_{\mathrm{t}})$. A gauge cell that is too small will not be able to obtain sufficient statistics as particles will tend to stay on the test chain; and if the volume is too large, the particles will tend to stay in the gauge cell and the test chain will not exist in the system cell. The size of the gauge cell can be calculated from an *a priori* estimate of chemical potential using Equation (4.23). In practice, it is useful to calculate the most suitable gauge cell volume during the simulation. Of course, for the mesocanonical ensemble to be valid, the gauge cell volume must be constant. But during the equilibration phase of the simulation (i.e. the discarded steps), no such restriction exists. The gauge volume can thus be adjusted using the simple relation, $V_{\mathrm{g}} = n_{\mathrm{g}}^{\mathrm{target}}/\rho_{\mathrm{g}}$, where $n_{\mathrm{g}}^{\mathrm{target}}$ is the desired average number of particles in the gauge cell and $\rho_{\mathrm{g}} = n_{\mathrm{g}}/V_{\mathrm{g}}$. Before the averaging (i.e. production) segment of the simulation commences, a final gauge volume is selected by either averaging previous volumes or using the last computed one. For this work, $n_{\mathrm{g}}^{\mathrm{target}}$ was selected as 10, a comprise of reasonable statistics and short simulations.

4.1.4 Simulation Details

A single freely-jointed linear Lennard-Jones (LJ) chain of length $n = 2$ to 500 was considered in two situations: in free spaces (approximating the zero-density limit) and confined in a spherical pore. Monomers were modeled as LJ beads that interact with all other nonbonded monomers. LJ parameters were selected to roughly mimic methylene monomers ($\epsilon/k_{\mathrm{B}} = 49.3$ K, $\sigma = 0.394$ nm) [235]. The LJ potential was truncated at 10σ. Covalent bonds were modeled using a bounded harmonic spring potential:

$$U_{\mathrm{bond}}(r) = \begin{cases} \frac{1}{2}\kappa_{\mathrm{b}}(r - r_0)^2 & \text{for } 0.5 \leq r/\sigma \leq 1.5 \\ \infty & \text{otherwise} \end{cases} \tag{4.26}$$

where r is the distance between bonded beads, r_0 is the equilibrium bond length, here equal to 1σ, and κ_{b} is the spring constant, taken to be $400\epsilon/\sigma^2$ [236]. The insertion volume (V_{ins}) is the volume where this potential is bounded (here between $r_{\mathrm{min}} = 0.5\sigma$ and $r_{\mathrm{max}} =$

1.5σ). Confinement effects were tested by utilizing a spherical enclosure of $d_{\text{acc}} = 10.05\sigma$, with either an attractive adsorption potential of LJ type, or a hard wall repulsion. Here, d_{acc} refers to the diameter of the pore that is accessible to the monomers, specifically the volume where the solid-fluid potential is less than zero [204]. Adsorption was modeled as an interaction between each monomer with a solid wall that was formed by a uniform layer of "smeared-out" LJ atoms. These solid-fluid interactions were integrated over the spherical surface to obtain a one-dimensional external potential, $U_{\text{adsorption}} = U_{\text{wall}}(r)$, where r is the distance between a given monomer and the pore wall. Parameters for the adsorption potential were solid-fluid LJ energy ($\epsilon_{\text{SF}}/k_{\text{B}} = 100.0$ K), solid-fluid LJ diameter ($\sigma_{\text{SF}} = 0.33$ nm), and the surface density of the LJ atoms in the wall (15.3 nm^{-2}), chosen to approximate alkane interactions with an amorphous silica surface. Free chains were simulated in a box with periodic boundary conditions, with volume selected so that density was not greater than 1×10^{-8}. Simulations lengths were at least 10^8 MC steps, with each move an attempted displacement or exchange. This relatively long simulation time ensured equilibration of the chain.

In addition to the monomer insertions and removals from/to the gauge, described by Equations (4.24) and (4.25), the system in the target cell was equilibrated using standard canonical MC moves that included random monomer displacement in the target cell, and chain reptation. On displacement, a randomly selected monomer was displaced in a random direction. Maximum displacement distance was selected so that the move is accepted $\sim$50% of attempts. Reptation, or the "slithering snake" move, involves growing the chain at one end while removing the opposite end [237]. Sampling of polymeric systems is often facilitated using other canonical moves, such as crankshaft type moves [238], partial regrowth via configurational bias [91, 239], and concerted rotation [240], just to name a few. Generally speaking, these additional moves are necessary to ensure reasonable sampling of state space for polymeric systems.

4.1.5 Single Chain in the Zero-Density Limit: Comparison to Previous Methods

Using MDGC, IGGC and MW trial particle technique, we calculated the incremental chemical potential (μ_{inc}) for chains of harmonically bonded LJ beads from $n = 2$ to 30 in a $1000^3\sigma^3$ periodic box to approximate the zero density limit, that is, at the same conditions that were studied by the MW method [11]. These authors considered two temperatures, $k_{\text{B}}T/\epsilon = 2$ and $k_{\text{B}}T/\epsilon = 8$. The higher temperature $k_{\text{B}}T/\epsilon = 8$ is well above the θ-point, which means that the chain accepts a self-avoiding random coil configuration characterized by the $R_{\text{G}} \propto (n-1)^{0.59}$ dependence between the radius of gyration R_{G} and the chain length n. This condition is analogous to the polymer being dissolved in a "good" solvent. On the contrary, $k_{\text{B}}T/\epsilon = 2$ is below the θ-point [115], which means that polymer-polymer interactions are more favorable than polymer-solvent ones; this effective attraction between the beads prevails over the entropic desire for a disordered (highly random) chain, and the polymer "condenses" into a globule with $R_{\text{G}} \propto (n-1)^{1/3}$. At the θ temperature, the entropic contribution (favoring chain expansion into a self-avoiding coil) and enthalpic contribution (favoring chain contraction into a globule) approximately cancel each other, and the polymer effectively behaves like a Gaussian chain with $R_{\text{G}} \propto (n-1)^{1/2}$. The θ temperature for a stiff-jointed LJ polymer was earlier estimated as $k_{\text{B}}T/\epsilon = 3.18$ when considering short range interactions with LJ potential cut off at $r_{\text{cut}} = 2.5\sigma$ [39], and approximately $k_{\text{B}}T/\epsilon$ $= 4$ with $r_{\text{cut}} = \infty$ [115].

Our results for $n < 30$ are displayed in Figure 4.1, along with the original MW results [11]. Our results agree quantitatively with the MW method (a difference of less than $0.01k_{\text{B}}T$ was observed) at $n > 10$. Both MDGC and IGGC results agree when $n > 10$. For shorter chains, the results of the IGGC and MW are in good agreement, while the chemical potentials obtained by MDGC deviate. This problem is similar to the one faced by very small non-polymer systems [10], since Equation (4.23), upon which the MDGC method is based, is itself an approximation truly valid only for large systems approaching the bulk limit. The MDGC method is reliable when the average number of particles in the gauge cell does not differ (within given accuracy) from the most probable number. In practice, this

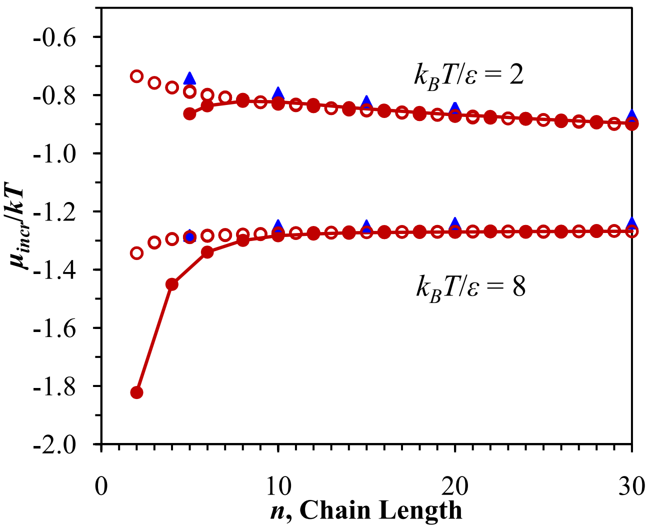


Figure 4.1: Comparison of the gauge cell and MW methods for calculating the incremental chemical potential $\mu_{\rm inc}$ in dilute bulk solution. Triangles are data from Ref. [11], calculated by using MW insertions at the end of the chain (to compare with published data, the term $\ln(V_{\rm ins})$ was subtracted from referenced data). Open circles correspond to IGGC method, and closed circles to MDGC method. The deviations between the IGGC and MW calculations of $\mu_{\rm inc}$ do not exceed $0.01k_{\rm B}T$. Note that IGGC provides at least two values of $\mu_{\rm inc}$ at given n, calculated from the overlapping histograms of IGGC simulations. As expected, $\mu_{\rm inc}$ calculated by MDGC method diverges for short chains of $n < 10$, when $\langle n\rangle < \langle n_{\rm g}\rangle$, and agrees with IGGC method for longer chains. The IGGC method should be used for short chains ($n < 10$), while MDGC method is sufficient at $n > 10$.

occurs when $\langle n \rangle$ (the average chain length) is larger than $\langle n_{\mathrm{g}} \rangle$. If this condition is not met, the ideal gas gauge cell (IGGC) method should be used. The IGGC method calculates μ_{inc} from individual (statistically significant) histogram density bins rather than the average density of the gauge cell and therefore the results do not depend on the gauge size. In this calculation, the gauge volume was selected to have an average particle number of 10 in the gauge cell. Each IGGC simulation, therefore, yields several points (of varying statistical certainty) in a plot of μ_{inc} versus n. In fact, multiple overlapping points in Figure 4.1 were determined from distinct gauge cell particle distributions from IGGC method. With either method, the feature of interest is the dependence of μ_{inc} with the chain length. For the high temperature case, μ_{inc} does not depend on n, for chains $n > 10$. This behavior is expected by the chain increment ansatz [80, 81, 115].At $k_{\mathrm{B}}T/\epsilon = 2$, μ_{inc} monotonically decreases with n, albeit slowly, and the ansatz does not hold.

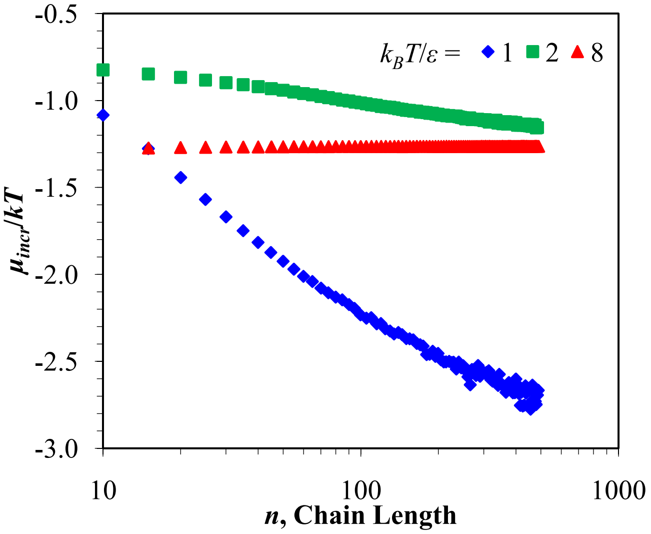


Figure 4.2: Chain length dependence of μ_{inc} at different temperatures, calculated by MDGC method. Larger statistical variation is present for long chains at the lowest temperature due to sampling inefficiencies. At temperatures below the θ-point, μ_{inc} decreases with the chain length, since larger globules have larger surface that favors additional monomer insertions.

To further investigate the chain length dependence, we used the incremental gauge cell method to study the behavior of long chains. Chains from $n = 10$ to 500 were simulated in free space at low ($k_{\mathrm{B}}T/\epsilon = 1$), intermediate ($k_{\mathrm{B}}T/\epsilon = 2$) and high ($k_{\mathrm{B}}T/\epsilon = 8$) temperatures. A semi-log $\mu_{\mathrm{inc}}(n)$ plot is presented in Figure 4.2. It is clear that the chain increment ansatz holds for very long chains in the high temperature (that is, for a self-avoiding coil type of configuration) case with a reasonable precision. The standard deviation in $\mu_{\rm inc}$ from $n =$ 15 to 500 is 1.49×10^{-3}, with a mean value of $-1.265 k_{\rm B}T$, indicating little variation of $\mu_{\rm inc}$ over a long range of chain lengths. At both intermediate and low temperatures, $\mu_{\rm inc}$ decreases monotonically with the length. Since both intermediate and low temperatures are below the θ-point the chain increment ansatz is not expected to hold. As the polymer globule grows, there is an increase in the attractive potential experienced by each new bead inserted. As a result, the incremental chemical potential decreases monotonically, and at a greater rate for a lower temperature (where attractive monomer-monomer interactions are stronger). The situation is similar to that in a liquid droplet of a small-molecule fluid in its equilibrium vapor: as the droplet grows, the fraction of molecules located at the droplet surface decreases, as hence the chemical potential decreases. For both low temperature cases, the incremental activity has an approximate power law dependence on the chain length, $\exp(-\mu_{\rm inc}/k_{\rm B}T) \sim bn^a$; for $k_{\rm B}T/\epsilon = 1$, $b = 0.8593$ and $a = 0.4008$, with correlation coefficient $R^2 = 0.9913$, and for $k_{\rm B}T/\epsilon = 2$, $b = 0.8408$ and $a = 0.0836$, with correlation coefficient $R^2 = 0.9924$. The exponent a diminishes as the temperature increases, however, in order to draw any conclusions on its temperature dependence, additional simulations are needed.

As an additional test of the employed equilibration scheme, and to determine the θ temperature in our model, we studied how the length of free chains affects the coil-globule transition temperature. To this end, we calculated the chain expansion ratio at various temperatures, $k_{\rm B}T/\epsilon = 1$ to 100, to investigate the globular to self-avoiding random coil transition. The chain expansion ratio is defined as the ratio of the radius of gyration squared and the radius of gyration squared of an ideal Gaussian chain. The results for various chain lengths are displayed in Figure 4.3. Short chains ($n = 20$) are compared to previous work [80]. Finite chain effects are immediately observed, as the transition from poor to good solvents occurs over a range of temperatures, rather than a first-order type transition expected from an infinite length chain [241]. Two longer chains, $n = 50$ and $n = 410$, showed sequentially sharper transitions. Chains at their θ temperature would be realized as exhibiting a chain expansion ratio of unity. From the three chain lengths

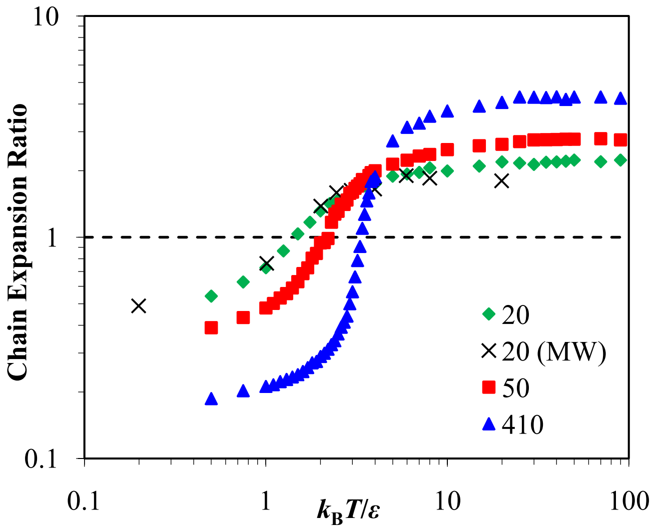


Figure 4.3: Temperature dependence of the chain expansion ratio, R_G/R_G^θ (R_G^θ is the gyration radius of an ideal chain), for various chain lengths. The incremental gauge cell method replicates previous MW calculations for the shortest chain [80]. As the chain length increases, the transition between globular and random coils becomes sharper. Note that, even for the longest chain of 410 monomers, this transition is far from a stepwise one.

studied, we observe the temperature at which the expansion ratio is equal to unity increase with chain length. For the last case of $n = 410$, this temperature is $\sim 3.3\epsilon/k_B$, which is within the expected range for the θ transition of finite chains [80, 115, 242]. If their trend is extrapolated, the curves seem to approach the infinite length θ temperature value of $\sim 4\epsilon/k_B$ [115].

4.1.6 Single Chain Confined to 7.5σ Nanopore

To observe the balance of entropic and enthalpic effects, we varied the temperature from $T^* = 1$ to $T^* = 8$. For single free chains, it is well known that below the θ temperature, monomer-monomer interactions dominate and chain collapses into a condensed form called a globule. Similarly, we know that above this temperature, thermal motion dominates and the chain behaves as a random coil [37]. The theta temperature of an LJ chain is between $T_\theta^* = 3$ and 4, depending the potential cut-off radius [39, 86] and chain length. Therefore, we selected cases below ($T^* = 1$), near ($T^* = 2$), in ($T^* = 3.2$), and above ($T^* = 8$) the θ transition temperature. To compare the structure of each system, we calculate the radius of gyration, R_{G},

$$R_{\mathrm{G}}^2 = \sum_i^n (\mathbf{r}_i - \mathbf{r}_{\mathrm{cm}})^2 \tag{4.27}$$

All values of R_{G} discussed are reduced by σ. To compare chains from all systems, the figures below plot all curves in terms of monomer density, that is, $\rho\sigma^3 = n\sigma^3/V_{\mathrm{acc}}$, including the free chain, which is actually calculated in the limit of zero density. However, because they are all reduced by the same volume, chain length effects are numerical comparable.

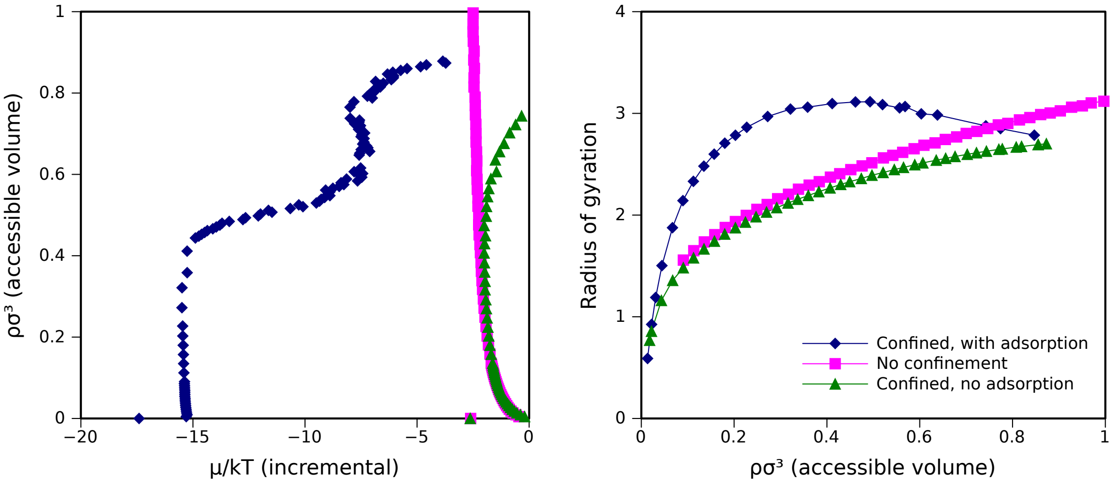


Figure 4.4: Incremental chemical potential (left) and radius of gryation results (right) for $T^* = 8.0$

The first case is the high temperature, $T^* = 8$. The incremental chemical potential for all three confinement cases is presented in Figure 4.4a, and their corresponding radius of gyration in Figure 4.4b. As predicted, μ_{inc} of a free chain (no confinement) does not depend on chain length once the limiting chain length is reached ($n \sim 5$). Once the chain is confined, it encounters an immediate entropic penalty. Where the unconfined chain's incremental chemical potential has no dependence on chain length, it becomes a strong function of pore filling when confined. The incremental chemical potential increases monotonically with density (chain length). The inclusion of an attractive adsorption potential does not significantly alter the behavior of the curve, suggesting that the entropic effects at such a temperature outweigh both the internal and external attractive potentials. The relatively constant shift between the confined hard wall and confined adsorbing walls is indicative of the strength of adsorption potential. The structure, as described by the radius of gyration, of the free chain is that of a random, self-avoiding walk. We found that for the free chain, $R_G \propto n^{0.590}$ (tested up to $n = 500$). This scaling exponent is very close the most accurate literature result of 0.5877 by Li et al. [243], further validating the algorithm. The R_G of the confined, hard-wall pore show that only for very short chains ($n < 15$) does confinement not impact the R_G of the polymer. After that, the RG approaches a value representative of the filled pore. When the adsorption potential is present, the R_G reaches a plateau quickly ($n \sim 50$). This indicates that the chain is evenly distributed in the pore volume, and increasing the density does not change the distribution of mass in the pore. Essentially, this is the equivalent to the adsorption of a super-critical fluid.

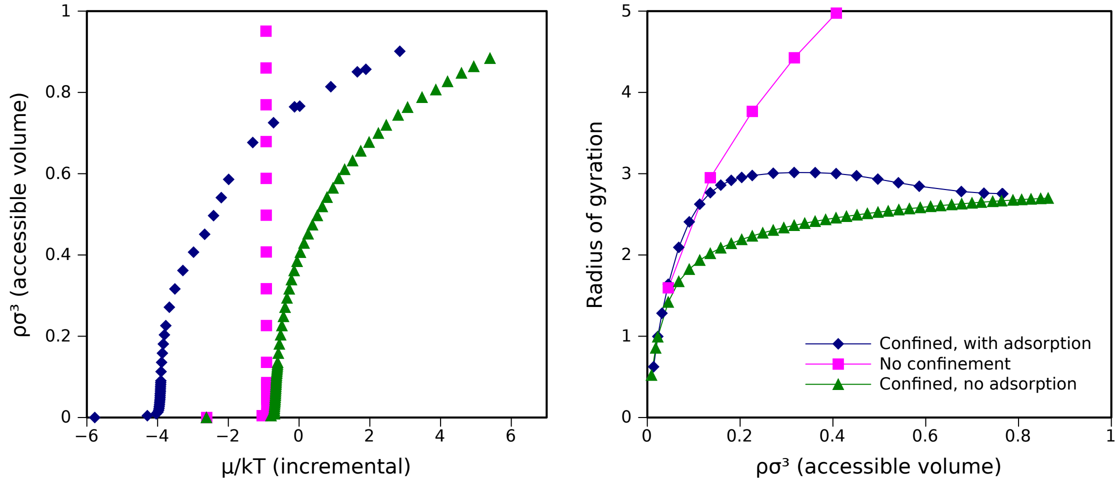


Figure 4.5: Incremental chemical potential (left) and radius of gryation results (right) for $T^* = 3.2$

The next temperature, $T^* = 3.2$ is close to the θ temperature calculated by Graessley [39] using a cut-off radius of 2.5 and a shifted potential, which was $T^* = 3.18$. Our results for this system are displayed in Figure 4.5. The incremental chemical potential of the free chain and hard-wall confined chain are qualitatively similar to the higher temperature case. The radius of gyrations are quite similar to the $T^* = 8$ case, with the free chain following a power scaling and the confined case following a logarithm-type scaling. This suggests that the thermal motion is still the dominant force. However, with the inclusion of the adsorptive force, two interesting effects can be observed. First, the incremental chemical potential is no longer a smooth exponential curve like the hard-walled confined chain, but has a weak inflection around a density of 0.4. Second, the R_{G} of the confined with adsorption chain now has a maximum, where the high temperature R_{G} had a plateau. The maximum indicates that there is a tendency (albeit a slight one) for the chains to exist closer to the radius of the pore. While no distinct layering is observed, this tells us that the average density of monomers is higher closer to the adsorbing wall.

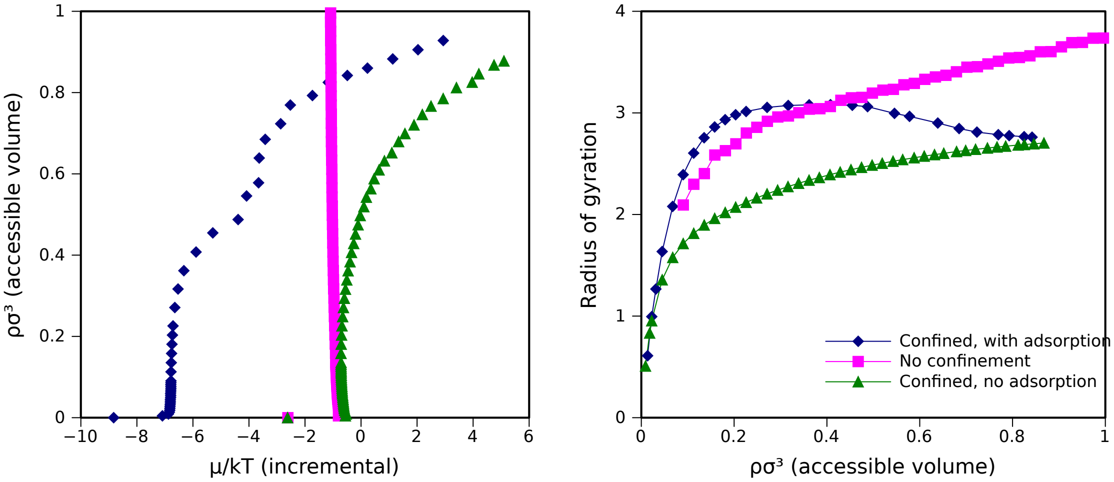


Figure 4.6: Incremental chemical potential (left) and radius of gryation results (right) for $T^* = 2.0$

Below the θ temperature, we have calculated two systems at $T^* = 2$ and 1. The former is closer to the transition point. Figure 4.6 shows the results for $T^* = 2$. The free chain incremental chemical potential is now clearly dependant on chain length. This dependence is logarithm in nature, and results from the net attractive potential of the growing globule in space. When the chain is placed in the hard-walled pore, confinement effects are observed for chains $\sim$50 monomers long. After this point, the incremental potential increases quick as in the previous cases. The RG of free and confined chains is now comparable, as they both exhibit a logarithm characteristic, but the free chain is shifted to higher R_{G} (the free chain is above 3 times larger at $n = 200$). However, the adsorbing chain now begins to show distinct regions in incremental chemical as a function of chain length. First, at low densities, a monolayer adsorbs onto the pore wall, and μ_{inc} is relatively constant. This is also observed as a clear maximum of R_{G} with respect to chain length, reflecting the tendency of the mass in the system to be located near the radius of the pore (the attractive pore wall).

As the chain length grows, it must fill the available volume of the pore, and RG decreases and approaches the value of the hard-wall confined polymer. Second, after the monolayer is formed, another transition is observed as the chain fills the pore. Once the pore is filled, adding additional monomers has a large energetic penalty, and the incremental chemical potential again increases rapidly. The shape of the incremental chemical potential curve is similar to an isotherm of an adsorbing critical fluid.

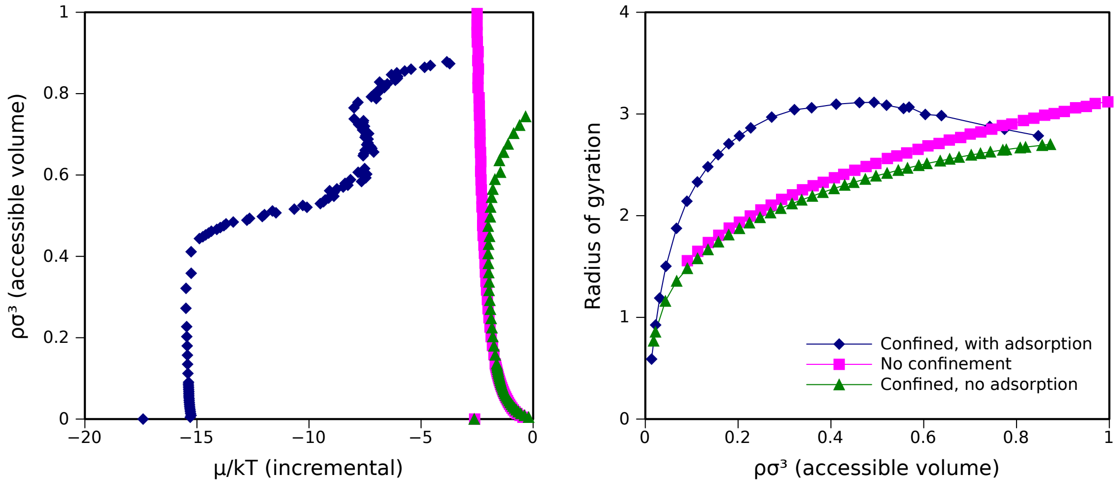


Figure 4.7: Incremental chemical potential (left) and radius of gryation results (right) for $T^* = 1.0$

The final case studied in this work is well below the θ temperature, where the bulk chain behaves as tightly condensed globule. The results for this case ($T^* = 1$) are presented in Figure 4.7. The confinement penalty for such a condensed chain is low—both the incremental chemical potential and the R_{G} of the chain confined to a hard-walled pore and the free chain are very similar to moderate lengths of $n \sim 100$. Even for long chains, when the pore is nearly filled at $n = 190$ ($\rho\sigma^3 = 0.86$), the R_{G} of the confined pore is only fractionally smaller than an unconfined pore (2.7 and 3.0, respectively). As before, when the confined chain is subjected to an adsorption potential, a peak in the R_{G} is observed, relating to the formation of a monolayer of monomers on the pore wall. A regime of nearly constant μ_{inc} is again observed from $n = 2$ to ~80. Unlike the previous case, the transition to a filled pore appears to have a sigmoidal shape reminiscent of a van der Waals loop in a canonical isotherm. The physical phenomenon associated with this characteristic is capillary condensation. As with physisorption of fluids, the lower branch is a layered fluid adsorbed onto the pore wall (having zero density at the center of the pore), and the upper branch has a nonzero density from the pore center to the pore wall. It's important to remember we are observing one single molecule in these systems, and what interesting is that many of the same physical insights from adsorption isotherms (μ vs N) are present in these single chain systems (chain length n vs μ_{inc}).

4.1.7 Single Chain Confined to 10σ Nanopore

The incremental gauge cell method was developed with further studies of adsorbed and confined chains in mind. Accurate prediction of partitioning (such as in polymer chromatography) and chain translocation through an opening (e.g. DNA/RNA transport) are examples of situations where the thermodynamics of confined polymers is important. To illustrate the potential uses of our method for confined polymers, a series of single chains were simulated in the nanopores with and without adsorption potential. As an instructive example, the confinement was chosen as a spherical pore of 10σ in diameter that may accommodate approximately 525 LJ particles either as a dense fluid or connected in the chain. For the rest of this section, 'confined' refers to a single LJ chain in a pore with no adsorption potential (i.e. only the hardcore repulsion between the beads and the wall), and 'adsorbed' refers to a chain subjected to the mean-field spherically integrated LJ potential [27] exerted by the pore wall. High, intermediate, and low temperatures, ($k_\mathrm{B}T/\epsilon = 8$, 2, and 1), were examined, representing the conditions of good and poor solvents. The dependence of μ_inc on chain length for free, confined, and adsorbed polymers at all three temperatures is presented in Figure 4.8. As mentioned before, confinement has a double influence on the free energy and behavior of the chain in that (1) it imposes limitations on the possible conformations of the chain, which diminishes the entropy; this effect becomes more important as the temperature increases, and (2) the adsorption field reduces the configurational energy of the polymer as its monomers are adsorbed on the wall; this effect becomes less important as the temperature increases. As Figure 4.8a shows, at $k_\mathrm{B}T/\epsilon = 8$ the confinement severely constricts conformations, as μ_inc increases dramatically with n when confined. The inclusion of an adsorption potential reduces μ_inc, but does not change its overall behavior, as μ_{inc} increases exponentially with the chain length. This is also reflected in the density profiles (Figure 4.9), as the monomer density $\rho\sigma^3$ remains relatively constant in the pore, with only a slight increase near the adsorbing wall. Note that while the chain increment ansatz holds for the free chains it fails for confined and adsorbed chains, for which the incremental chemical potential increases with the chain length in a non-linear fashion similar to an adsorption isotherm at supercritical (for confined LJ fluid) conditions that would be measured for non-bonded monomers.

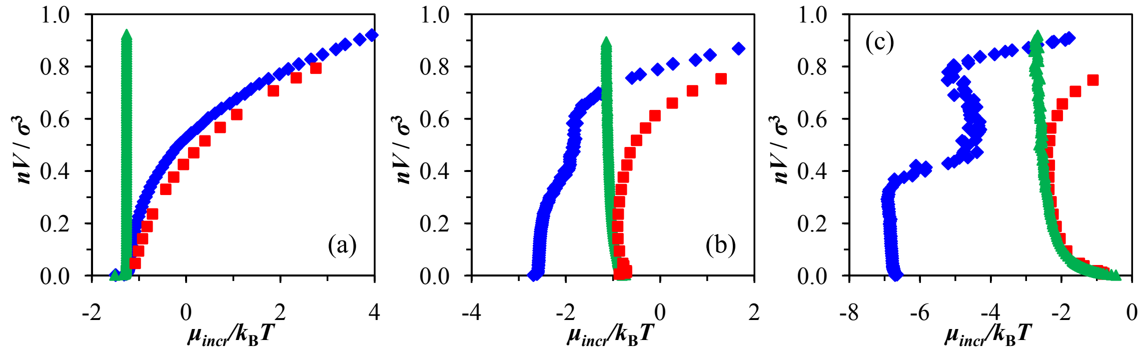


Figure 4.8: Chain length dependence of the incremental chemical potential of free, confined, and adsorbed chains at different temperatures, k_BT/ϵ = (a) 8, (b) 2, (c) 1. Green triangles (▲)—free chain (no confinement), red squares (■)—chain confined to the spherical pore of volume $V/\sigma^3 = 523$ with hard non-adsorbing walls, and blue diamonds (◆)—chain in the same pore, but with adsorbing LJ walls.

When the temperature of this system is reduced below the θ temperature to $k_BT/\epsilon = 2$, the behavior of the incremental chemical potential in confinement becomes more complex; see Figure 4.8b. In the case of hard wall potential, μ_{inc}, is similar to the free chain up a certain length. After this point, confinement effects manifest as steric restrictions, and μ_{inc} begins to increase exponentially as in the high temperature case. Because the incremental chemical potential for free chains decreases while below the θ temperature, μ_{inc} for confined chains has a minimum at some chain length. This effect is more pronounced at lower temperate of $k_BT/\epsilon = 2$ where the minimum is achieved at $n \sim 100$. When the adsorption potential is applied, a clear shift of μ_{inc} is observed. Unlike the high temperature case, below the θ temperature the adsorption force mostly overcomes the entropic effects of confinement. A large shift, $\Delta\mu_{\text{inc}} \approx -3k_BT$, occurs at $k_BT/\epsilon = 2$ for short chains as they are predominately adsorbed onto the pore walls. As chain length increases, the thickness of the adsorbed layer grows and μ_{inc} increases gradually. An inflection point is observed at $n = 275$, at approximately $\mu_{\text{inc}} = -3.6 k_{\text{B}}T$. Near this density, the pore becomes filled as can be seen from the increase in density as well as the local density profiles, see Figure 4.9. The polymer chain now occupies all available volume in the pore. This is analogous to phase change in a pore filled with a simple fluid [244]. When a pore is filling with a simple fluid, a transition is observed from a vapor-like state to a liquid-like state at the vapor-liquid equilibrium pressure. In this study, a single LJ chain is observed transitioning from layering to filling, which occurs at some characteristic incremental chemical potential. To be clear, this is not a phase change, but a conformation change. Although adsorbed layers are sharp, all mass is not tightly bound in this layer. The selected density profiles (Figure 4.9) show a slight 'tail' towards the center of the pore, suggesting the system contains enough thermal energy for loops or pending ends of the chain to exist unadsorbed for an appreciable amount of simulation time.

The low temperature case was simulated at $k_{\text{B}}T/\epsilon = 1$, a temperature well below the θ temperature of the globule transition point in the bulk. A plot of μ_{inc} versus $\langle n \rangle$ is presented in Figure 4.8c. Confinement without adsorption potential has little effect on μ_{inc} until high densities; the chain exists as a tight globule with monomer interactions dominating whether confined or free. Confinement effects are not 'felt' until the globule's radius of gyration approaches the size of the pore. In the adsorption case, layering is strong, as the polymer chain strongly adsorbs to the pore wall. Unlike the previous thermal cases, μ_{inc} as a function of chain length is not monotonic increasing. At small chain lengths, it is nearly constant, decreasing from -6.7 to $-6.9 k_{\text{B}}T$, from $n = 5$ to 170. In this range, the chain is adsorbed in a growing monolayer. At $n \sim 170$, μ_{inc} begins to increase, and the second adsorbed layer starts to grow. A transition region, from $n = 250$ to $n = 410$, exhibits a negative trend of μ_{inc}. In this region, the second and third layers grow until transition to a filled pore is observed. Plots of the density profile (Figure 4.10) confirm what is observed in behavior of μ_{inc}; sharp peaks of high local densities indicate distinct layering occurring in the pore.

The influence of adsorption can also be noticed when comparing the radius of gyration R_{G} of free and confined chains. Figure 4.9 displays R_{G} as a function of the chain length for

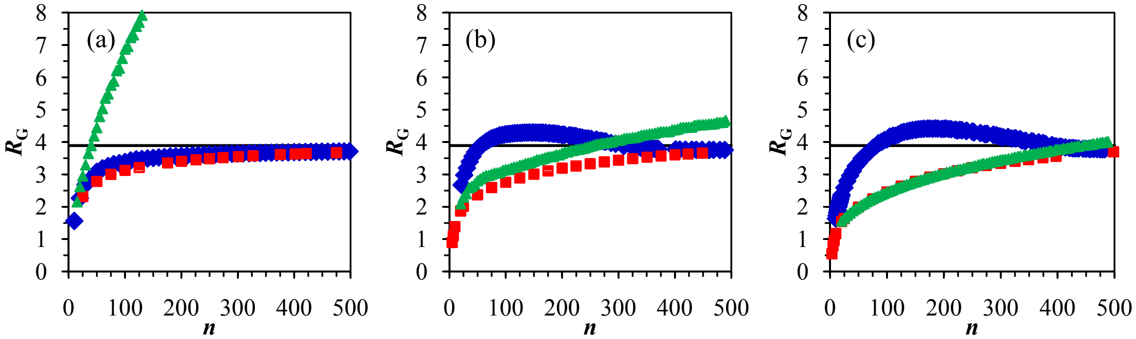


Figure 4.9: Chain length dependence of the radius of gyration of free, confined without adsorption, and confined with adsorption chains at temperatures, k_BT/ϵ = (a) 8, (b) 2, (c) 1. Green triangles (▲)—free chain (no confinement), red squares (■)—chain confined to the spherical pore of volume $V/\sigma^3 = 523$ with hard non-adsorbing walls, and blue diamonds (♦)—chain in the same pore, but with adsorbing LJ walls. The solid horizontal line indicates the radius of gyration for a hypothetical case of the uniform mass distribution within the pore of radius R, $R_G = 3R^2/5$.

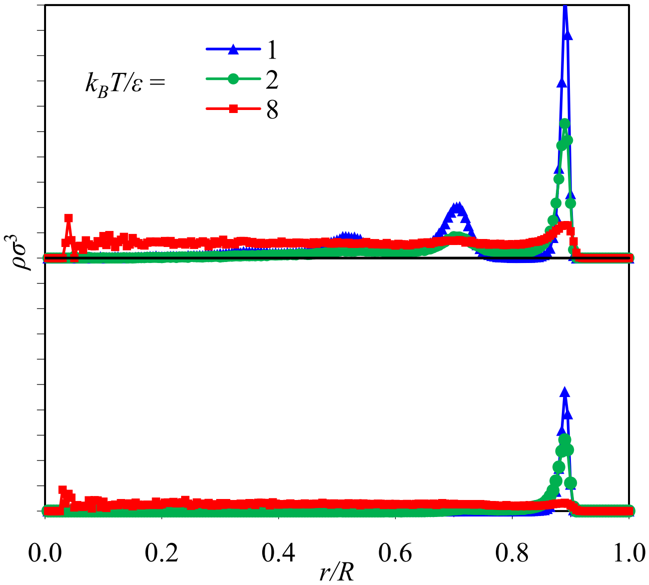


Figure 4.10: Density profiles of chains in adsorbing pores at different temperatures. Top plot—chains of $n = 260$ (pore filling density of ∼0.5); bottom plot—chains of n = 105 (pore filling density of ∼0.2). The maximum density ($\rho\sigma^3$) for this pore is ∼10. Note a distinct layering at low temperatures.

all three temperatures, and for each type of confinement. In the high temperature case, μ_{inc} for both the adsorbing pore and hard-wall pore shows an asymptotic approach to the limiting value of R_{G} for a uniform density distribution, $3R^2/5$, in a spherical pore with radius R. The value of the adsorbing and hard-wall nearly coincide, which strongly suggests no appreciable adsorption and thus negligible effect of adsorption potential. However, when temperature is decreased, two important effects are noticed: (1) R_{G} of the free chain approaches that of hard-wall confined chain, and (2) R_{G} of the adsorbing chain has a maximum, with a value larger than the limiting case of uniform density, at a relatively small chain length. The first point has already been made above; as the temperature is reduced, the polymer globule is more tightly condensed, and can be smaller than the pore containing it, resulting in no or little confinement effects. The second point reflects the creation of an adsorbed layer at the wall; in this case, the adsorbed chain exhibits R_{G} larger than that of the uniform density. The location of the maximum of R_{G} helps to identify transitions from growing adsorbed film to volume pore filling. The adsorption curve on the low temperature subplot Figure 4.9c has a broader curve than the intermediate case of Figure 4.9b. This suggests that at the lower temperature, the chain remains in an adsorbed, film-like conformation for longer chain lengths then that of the middle temperature, before transitioning to a pore volume filling conformation. The evolution of the chain conformation from adsorbed film to pore filling is illustrated on selected snapshots presented in Figure 4.11.

When in contact with an attractive substrate, many features of $\mu_{\mathrm{inc}}(\mathrm{n})$ are analogous to adsorption isotherms of simple fluids confined to nanopores. For comparison with the chain adsorption isotherms, $\mu = \mu(N)$, the canonical isotherms of non-bonded LJ particles were calculated using the gauge cell method and are displayed in Figure 4.12. All parameters (e.g. the fluid-fluid and fluid-solid interaction) were the same as for the chains, with the exception of the absence of the harmonic bond potential (Equation (4.26)). Panel A of Figure 4.12 gives an example of typical fluid behavior when confined to nanopores. The left-most curve, $k_{\mathrm{B}}T/\epsilon = 0.7$, is a subcritical fluid that would exhibit hysteresis on absorption and desorption. It is characterized by an S-shaped, van der Waals type loop [50, 167]. The lower branch ($\rho\sigma^3 < 0.4$) indicates a vapor-like phase, while the upper branch ($\rho\sigma^3 > 0.7$) shows a liquid-like phase. They are joined at their respective spinodal points by an unstable

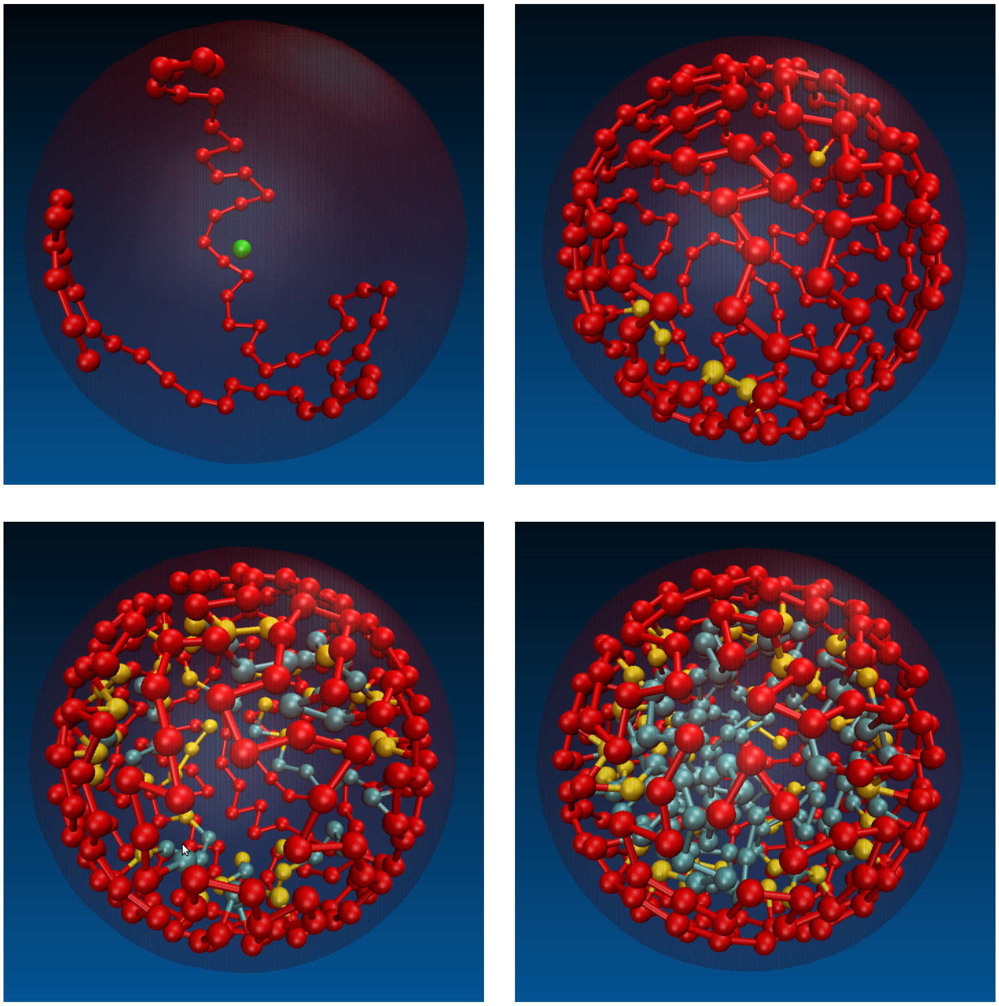


Figure 4.11: Selected snapshots of chain conformations for an adsorbing chain at $k_{\mathrm{B}}T/\epsilon = 1$. Red balls indicate the monomers in contact with the pore wall (i.e., monolayer adsorption), yellow indicates the second adsorbed layer, and the remaining monomers are colored blue (green ball is indicator of pore center). From left to right, $n = 55$, a short chain fully adsorbed; $n = 209$, near the monolayer capacity (several short 'loops' are visible); $n = 295$, full monolayer plus a more weakly adsorbed second layer; $n = 392$, a pore volume filling conformation. Rendered using VMD software [245].

branch. Thermodynamic integration of the isotherm and application of Maxwell's rule yield the position of the vapor-liquid equilibrium. The grand canonical isotherm of this system would exhibit hysteresis between the spinodal pressures of condensation and evaporation. The next isotherm, $k_\mathrm{B}T/\epsilon = 1.0$, is close to a critical fluid (the critical temperature of an LJ fluid is $\sim 1.3\epsilon/k_\mathrm{B}$) [246]. In this case, the pore contains a growing film at low chemical potentials, and exhibits a sharp yet reversible transition to a liquid-like state at $\mu \sim 5.2 k_\mathrm{B}T$. The last two temperatures considered are supercritical, and the adsorption isotherms reflect a pore volume filling with fluid density increasing monotonically with μ.

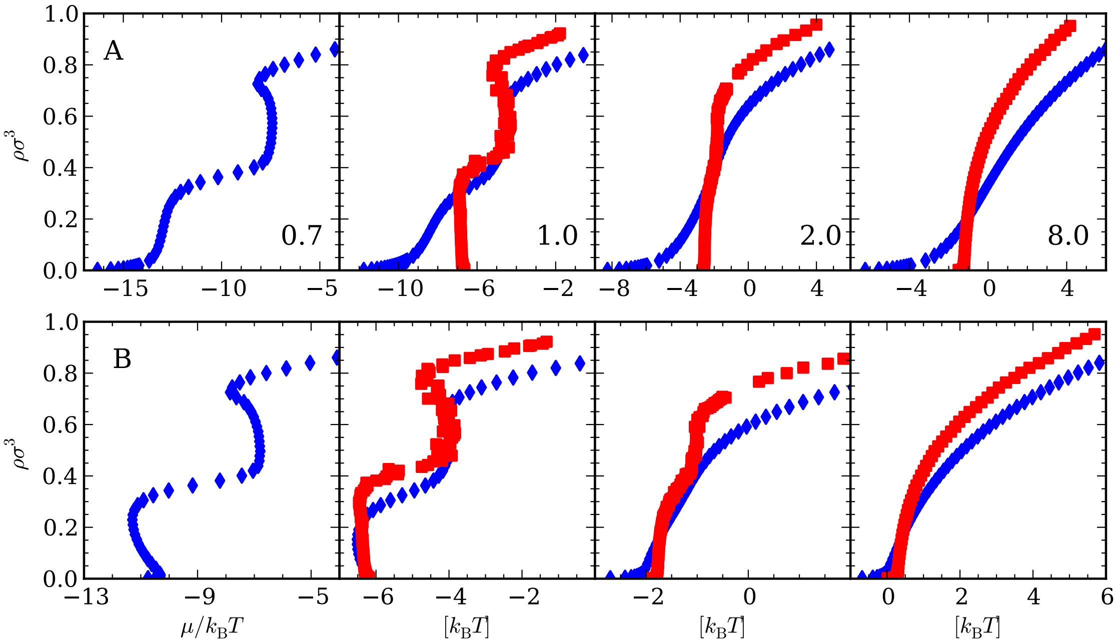


Figure 4.12: Comparison of the monomer isotherms and the incremental chemical potential of LJ chains in a 10σ adsorbing pore at various temperatures. Panel A: Red squares (■) report the incremental chemical potential of the LJ chain, using Equation (4.10′). Blue diamonds (♦) represent the isotherm of a monomer fluid. Panel B: Red squares report the incremental chemical potential subtracted by the average contribution of a harmonic bond, Equation (4.28). Blue diamonds represent the excess isotherm by excluding the ideal term, $k_\mathrm{B}T \ln[(N+1)/V]$ from the "true" isotherm. At the lowest temperatures (leftmost subplot), the isotherm is subcritical, with metastable and labile states connecting the stable vapor branch to the stable liquid branch. Increasing temperature forces a transition to a critical fluid, and finally, to a supercritical fluid.

Except for significantly more gradual formation of the monolayer, the monomer adsorption isotherm resembles the chain length dependence of the incremental chemical potential.

The apparent distinction prior to the monolayer formation is not surprising, since there are two factors affecting the difference of free energies between free and bonded monomers. The main difference between the monomer and incremental chemical potentials comes from the loss of translational degrees of freedom of bonded monomers due compared to free monomers. There is also a loss of free energy by removal of the harmonic bond between consecutive monomers. Panel B shows the same data as panel A, but the monomer adsorption isotherms are presented as functions of the reduced chemical potential with the characteristic contributions from these two factors subtracted: the monomer chemical potential is reduced by the ideal contribution, $k_\mathrm{B}T\ln[(N+1)/V]$ and the incremental chemical potential is reduced by the average bond energy contribution per monomer,

$$\mu_\mathrm{inr}^\mathrm{bond} = -k_\mathrm{B}T\ln\left[\frac{4\sqrt{2\pi^3}(\kappa_\mathrm{b}r_0^2+k_\mathrm{B}T)}{\kappa_\mathrm{b}\sqrt{\kappa_\mathrm{b}/k_\mathrm{B}T}}\right] \tag{4.28}$$

In doing so, one can quantitatively compare the monomer isotherm to the chain incremental chemical potential isotherm and find the two strikingly similar, albeit somewhat superficially. While one can distinguish on the S-shaped isotherms the regions of metastable and labile states separated by turn-over points of "spinodals", and also indentify the "critical" temperature above which the isotherms are monotonic, this physical picture is applicable only for the phase behavior of simple fluids. Such behavior of the chain incremental chemical potential cannot be treated in terms of phase equilibrium and criticality, yet it clearly reflects the variation of the chain conformations from an adsorbed state to a pore filling state as the chain length increases. A similar behavior is expected for the chain of given length with the decrease of the adsorption potential at fixed temperature or with the increase of temperature at fixed adsorption potential.

4.1.8 Computational Efficiency

We tested the efficiency of incremental gauge cell relative to that of the MW method by comparing the length of simulation time of each method to arrive at a statistically equivalent average of incremental chemical potential. The simulation was ended when a desired level of precision was obtained in the averaging of chemical potential. To estimate the standard deviation of such correlated samples, we applied the standard blocking method of Flyvberg and Petersen [247]. In this method, the trajectory is divided into smaller and smaller "blocks," and the results for each block are compared. We implemented the algorithm of Kent et al. [248] to calculate these quantities as the simulation progressed. The simulation was considered converged in μ when two conditions were met: first, if the blocking procedure produces a valid estimate of variance (where variance is approximately constant over blocks with more than 10 samples), and second, if the estimated standard deviation was less than a specified value and within an acceptable error range. For the MW method, the Boltzmann factor of the trial insertions was sampled. Once the first criterion above was met, the resulting standard deviation was used to calculate the precision in chemical potential. The same method was used with the gauge cell, but the monitored quantity was the number of particles in the gauge. In both cases, the simulation was stopped after a certainty of $\pm 0.01 k_{\mathrm{B}}T$ in chemical potential was obtained. The system considered was the same model as described above, with chain length varying from $n = 10$ to 300, confined in a hard wall sphere of $d = 10\sigma$, at temperature $k_{\mathrm{B}}T/\epsilon = 1$, calculated using either the canonical ensemble with Widom insertions, or the mesocanonical ensemble using the gauge cell method. Each MC step corresponds to one attempted monomer displacement and either one trial monomer insertion or attempted gauge monomer exchange, depending on the method of chemical potential calculation. Therefore each step (in both canonical and mesocanonical ensembles) requires $O(2n)$ calculations to complete.

The results of this approach are displayed in Figure 4.13. We normalized the number of steps in each simulation by

$$t_{\text{eff}} = t_{\text{actual}} \left(\frac{\delta_{\text{actual}}}{\delta_{\text{specified}}} \right)^2 \tag{4.29}$$

where t is the number of MC steps, either "effective" or actual, and δ is the error in chemical potential. This was necessary because of the nature of equilibration of long chains using only local moves. In some cases, it takes many local moves for the transition to the next global conformation, and many unique conformations are required to achieve a suitably converged average. In such a case, the resulting error becomes much lower than initially

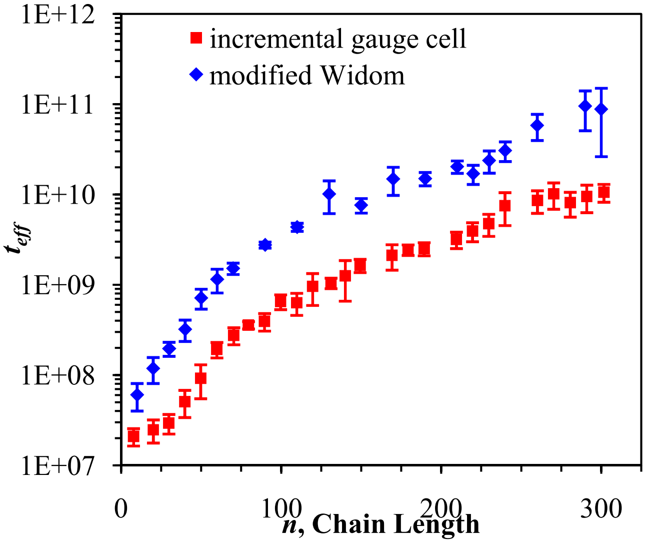


Figure 4.13: Comparison of the computational efficiency between the incremental gauge cell and MW methods for a single chain confined in a hard wall sphere of diameter 10σ at $k_{\rm B}T/\epsilon = 1$. The vertical axis represents normalized length of simulation required to reach an acceptable level of error in the chemical potential calculation, chosen to be $\pm 0.01 k_{\rm B}T$. The relative efficiency of the gauge cell method increases with the chain length.

specified. The error bars in Figure 4.13 were calculated simply by averaging the estimated error of the standard deviation in the blocking plateau region. Using the gauge cell method brings approximately an order of magnitude decrease in the number of steps required to equilibrate and collect a significant average. By allowing the exchange of particles, the configuration is allowed more degrees of freedom than a constant length chain, where the trial Widom insertion by definition cannot influence system. The exchange move combines both equilibration and “measurement” of chemical potential in one MC move, and as such it facilitates further mixing of the system. This results in significantly shorter simulations than using the conventional Widom trial particle insertion to calculate chemical potential.

It is worth noting that the most efficient algorithm for calculation of this particular system (a single LJ chain) may be Grassberger’s PERM [85]. However, the incremental gauge cell method is intended to be a general methodology for calculation of chemical potential in the complex and confined systems, which require the traditional Markov chain MC approach. PERM is a static method not capable of calculating complex systems (e.g. multiple chains or chains in an explicit solvent), whereas the simple fluid gauge cell method has already been extended to mixtures, and generally only requires an additional gauge cell per extra component [49].

4.1.9 Discussion and Conclusion

The gauge cell method suggested in this work offers substantial advantages in calculating the chain chemical potential over the two most popular strategies, namely the insertion of the entire chain via various configurational bias methods [72, 91, 93] and the incremental insertion according to the MW scheme [11]. Following the work of Kumar et al. [11], we exploited the concept of the incremental chemical potential as the increase of the chain free energy upon its growth by one monomer, as shown by Equation (4.8). This strategy avoids relying on the insertions of the entire chain in one MC move, whose probability even using customized bias schemes becomes very rare as the chain length increases. The use of the gauge cell offers several advantages over the trial monomer insertion strategy of the MW method: first, several points of the μ_{inc}–n dependence may be calculated in a single simulation via the IGGC framework (Equation (4.22)), with the size of the gauge cell dictating the level of density fluctuations in the system; and second, replacement of the trial chain growth by real increment and decrement of the chain by one monomer provides additional "mixing," thereby facilitating the equilibration of the system and making the simulation more efficient, especially for inhomogeneous and dense systems. We quantitatively validated our method against the published data of the MW method. At the same time, we estimated the sampling efficiency gain as approximately one order of magnitude when utilizing the gauge cell method over the MW method. Further progress in efficiency in dense environments may be achieved by using advanced MC move types developed for chain molecules within the configurational bias technique [97, 231, 239, 249].

The suggested method was applied for calculations of the chemical potentials of free chains in a wider range of the chain lengths, up to $n = 500$, that was studied before by traditional MC. We confirmed that above the θ temperature, the incremental chemical potential of sufficiently long random coils is constant in accord with the chain increment ansatz [81]. For globular chains below the θ temperature, the chain increment ansatz progressively fails and requires a logarithmic correction. This dependence hinders the main advantage of the chain increment ansatz that the chain chemical potential can be calculated from few simulations with short chains. As such, the computational advantages of the incremental gauge cell method, combined with interpolation opportunities provided by the established logarithmic correction, seem to be especially important in studies of globular polymers, for which the chain increment ansatz is not valid.

We explored the effects of confinement on the incremental chemical potential of a single chain, focusing our attention on the competition of steric restrictions imposed by confinement of the polymer and attractive adsorption forces between individual monomers and the substrate. These two factors determine the free energy difference of confined polymers and thus govern polymer sorption from dilute solution, which is of significant practical importance in polymer chromatography. At good solvent conditions, the steric restrictions are considerable even for relatively short chains. As the chain length increases, sorption of a polymer quickly becomes unfavorable due to the loss of entropy, which overcomes the gain of enthalpy due to the adsorption attraction that is limited to the chain fragments being in the immediate vicinity of the pore walls. Entropic effects are leveled and may be even overcome by enthalpic adsorption effects. At sufficiently low temperatures, entropic effects are completely masked by adsorption, and short polymer chains are strongly adsorbed to the pore wall forming a monolayer film. The incremental chemical potential decreases slightly as a monolayer is formed. Only when the chain length exceeds the monolayer capacity do some fragments of the chain protrude into the pore volume forming floating loops. Upon further increase of the chain length, the whole pore volume becomes gradually filled, however the monomer density distribution at the pore walls is always larger than in the pore center.

The dependence of the incremental chemical potential on the chain length resembles the canonical ensemble isotherm of capillary condensation of a fluid of free monomers and has a characteristic S-shape of van der Waals isotherms. After the formation of the monolayer, μ_{inc} as the function of the chain length first increases up to a certain maximum and then decreases down a certain minimum and further monotonically increases. This behavior is limited by a certain temperature, above which the incremental chemical potential monotonically yet non-linearly increases with the chain length. The turnover points of S-shape

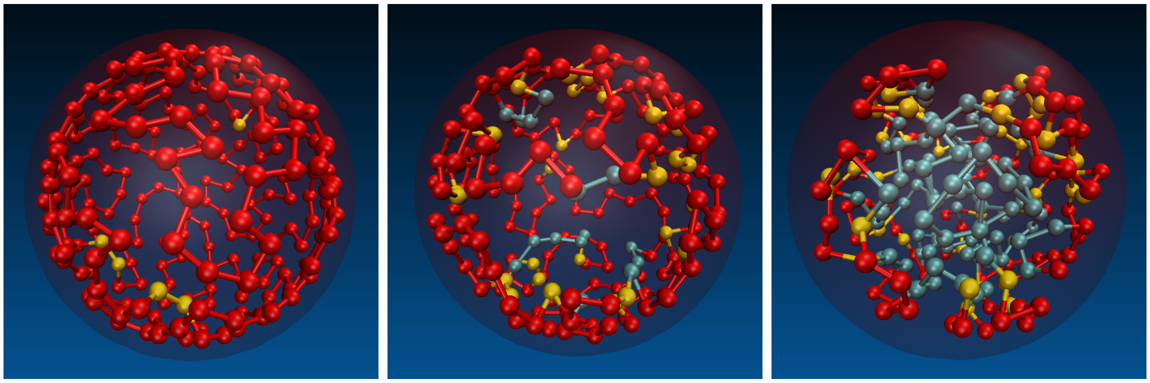


Figure 4.14: Selected snapshots of chain conformations for the adsorbing chain of length n = 200, illustrating the transition from adsorbed film to pore filling conformation upon the increase of temperature, $k_{\mathrm{B}}T/\epsilon = 1$ (left), 2 (center) and 8 (right). Red balls indicate the monomers in contact with the pore wall (i.e., monolayer adsorption), yellow indicates the second adsorbed layer, and the remaining monomers are colored blue. Rendered using VMD software [245].

$\mu_{\mathrm{inc}}(n)$ dependencies are apparently analogous to the spinodals on the vapor-liquid phase diagram of subcritical fluids, as the limiting temperature of S-shape behavior is to the critical temperature. However, this analogy should not be extended to the description of the chain length dependence of the incremental chemical potential as characteristic to phase transition. At the same time, we envision that the adsorption behavior of the chain of a given length upon increase of the temperature, or upon the decrease of adsorption potential, can exhibit transitions between adsorbed film and pore filing conformations. In this case, the analogy with the vapor-liquid transition may be useful. A series of snapshots illustrating the transition from adsorbed film to pore filling conformation of the chain of length n = 200 with the temperature increase is given in Figure 4.14.

We provided a rigorous statistical mechanical derivation of the definition of the incremental chemical potential in the general case of a polymer melt consisting of a mixture of chain lengths, as well as the foundations of the gauge cell MC scheme. Although the validation of the proposed method and the presented applications are limited to single chain systems, this general approach can be further advanced to include multi-component, copolymers, and polymer/solvent systems. Immediate practical applications of the proposed method worth noting are the problem of polymer partitioning on porous substrates between size-exclusion, critical adsorption, and adsorption chromatography, and the dynamics of translocation of chain molecules through nanopores in biological and solid-state membranes.

4.2 Translocation

4.2.1 Introduction

Polymer translocation is the process of chain movement from one compartment to another through a narrow opening that is significantly smaller than the chain itself [250]. Translocation is implicated in many physical phenomena of scientific interest, including diffusion of DNA out of the cell nucleus [251], injection of viral genetic material into a host cell [252], transport of proteins and polypeptides across openings [253], and drug delivery [254]. Recent interest in computational translocation studies has been fueled partly by the desire for fast and accurate nucleotide sequencing using biological and solid-state membrane nanopores [255]. Translocation may also be an important mechanism in chromatography of macromolecules. Nanoporous adsorbents are widely employed for separation of synthetic and biological polymers [12].

Translocation has been a classic problem in polymer physics for some time. Polymer translocation can be either unforced or forced. Unforced translocation is due to diffusion of constituent monomers through the opening. An entropic barrier that is associated with the reduction of chain conformations as the polymer threads the opening must be overcome for translocation to be successful. Forced translocation is facilitated by applied driving forces, such as electrostatic [256, 257], hydrodynamic [258], or adsorption fields [259–262]. The existence of free energy barriers makes the translocation problem reminiscent to the classical problem of nucleation that is solved employing the Fokker-Plank (FP) formalism [65, 250]. The pioneering works of Park and Sung [259, 263] utilized the Gaussian chain model to obtain the free energy landscape during the translocation of ideal chains permeating a pore within a non-interacting membrane. It was assumed that the free energy landscape could parameterized by the single coordinate, the degree of translocation defined as the number of monomers that have successfully translocated from one side of the membrane to the other. The translocation dynamics was modeled by the FP equation, more specifically, by the 1d Smoluchowski equation of diffusion along this coordinate, with the free energy gradient acting as a driving or resisting force depending on its sign. Muthukumar [65] has combined the FP approach with the scaling theory for the free energy landscape of translocating chains. The primary assumption of the FP approach is that translocation proceeds significantly slower than the relaxation time of the chain on either side of the membrane. Although this has been questioned using scaling arguments [264, 265], the approach captures the essential physics of the translocation process and the main results compare favorable with experiment [250]. Recently, Mirigian et al. found good agreement between the FP approach and detailed Langevin Dynamics (LD) studies, even in complex polyelectrolyte systems [266]. Kong and Muthukumar [267] extended the FP approach to unforced translocation of non-ideal polymers using self consistent field theory (SCFT). Several attempts to better fit the one-dimension dynamical approach to observed anomalous diffusion behavior has led to application of the fractional Fokker-Planck equation [268]. Translocation dynamics of both forced and unforced systems have been studied directly in simulations using LD [269–271], Brownian Dynamics [272, 273], Molecular Dynamics (MD) [274], Dissipative Particle Dynamics [275], and dynamical Monte Carlo (MC) techniques [262, 276–278]. Such studies range from relatively simple, lattice models to large-scale MD systems modeling the translocation of DNA through biological pores [279]. These simulation studies suggest a rich variety of environments where translocation is relevant. For a recent review of these theoretical and simulation methods, see ref [280].

Previous studies suggest that forced translocation encompasses weak and strong regimes, with different scaling relationships [65, 262, 281]. A strong driving force masks entropic confinement effects, thus the translocation can be a function of the applied force alone [65]. Conversely, in the weak regime, the driving force is on the same order as the entropic resistance, and the translocation times depend on both factors. A system of interest that exemplifies this interplay is translocation into an adsorbing pore. The attractive adsorption potential favors translocation, yet the chain suffers an entropic penalty due to confinement in the pore. The balance of these enthalpic and entropic free energy contributions is critical in many polymer systems, such as theta chain transitions [282] and the critical point of adsorption in polymer chromatography [13]. Adsorption effects on translocation through a membrane pore have been studied in several papers [260, 262]. Recently, Yang and Neimark

[261] employed the FP approach combined with the SCFT calculations of the free energy landscape for studies of translocation into adsorbing pores and performed a detailed study of the competition of surface adsorption and confinement effects in the process of translocation into a pore. This work increases the level of details from the mean field SCFT resolution to the molecular level of Monte Carlo (MC) simulations. Although the MC simulation operates with entirely different and more realistic chain models, the results obtained below generally confirm the conclusions drawn from SCFT. We found a non-monotonic dependence on translocation times as a function of adsorption potential that can be divided into three regimes: weak potentials garner fast but improbable translocation, moderate potentials give slow translocation, and large potentials yield fast translocation.

In our MC simulations, we employ a freely jointed chain model with Lennard-Jones (LJ) non-bonded interactions that is standard in molecular modeling of polymers. As a reference, the ideal freely jointed chain model is used, in which the absence of non-bonded interactions allows for the monomer overlap. The MC simulation similarly to SCFT enables calculations of the free energies of equilibrated states at given thermodynamic conditions and constraints. The translocation dynamics is not monitored explicitly; rather it is studied by solution of the FP equation of diffusion along the free energy landscape determined in MC simulation as a function of the degree of translocation. To calculate the free energy, we apply the incremental gauge cell (IGC) method suggested by us recently [208]. This method is based on the mesocanonical ensemble, which considers the system of interest in thermodynamic equilibrium with a finite reservoir called the gauge cell [9, 10]. IGC "measures" the incremental chemical potential [11] (the difference of the chemical potential between an n-mer and an $(n+1)$-mer) by allowing the chain to grow or shrink by exchanging the terminal monomers of the chain with the free monomers in the gauge cell. The chain free energy represents the sum of the incremental chemical potentials. The IGC method was found [208] to be an order of magnitude more efficient than the modified Widom approach [11]. It has been used to study adsorption effects of confined polymers [207, 208].

The rest of this section is structured as follows. In Section 4.2.2, we describe the systems considered, the molecular model and simulation parameters, and the FP approach to translocation dynamics. Section 4.2.3 presents the results of free energy landscape calculations from MC simulation, and discusses its implication on translocation. Section 4.2.4 details the analysis of translocation dynamics obtained from the integration of the FP equation. In Section 4.2.5, we present our conclusions and critical analysis of the suggested approach.

4.2.2 Model

We model the translocation of a homopolymer chain into an adsorbing spherical pore (*trans* compartment) through a narrow window from a dilute solution (*cis* compartment) using off-lattice Monte Carlo simulations. The progress of the translocation process is characterized by the degree of translocation s representing the number of chain segments in the *trans* compartment. The goal of MC simulation is to determine the variation of free energy of the translocating chain $\mathcal{F}$ as a function of the degree of translocation s. The *cis* and *trans* compartments are modeled as independent subsystems (Figure 4.15), following the classical approach [65, 263]. The *cis* compartment represents a semi-infinite space limited by the hard non-adsorbing wall, and *cis* sub-chain of length $N - s$ is considered as tethered to the wall at the pore opening, whose size is neglected. The *trans* compartment represents a spherical pore of radius R with adsorbing walls, and *trans* sub-chain of length s is tethered to the pore wall at the pore opening. The initial condition is chosen as the chain of length N in the *cis* compartment tethered to the pore opening. As the translocation progresses the chain is modeled as a composite of *cis* and *trans* sub-chains tethered to the pore opening. The behavior of the chain within the pore opening is not considered, since the chain fragment in the pore is assumed to contribute a constant part into the total chain free energy. Within these assumptions, the free energy of the translocating chain is the sum of the free energies of the tethered chains in the *cis* and *trans* compartments,

$$\mathcal{F}(N, s) = F_{cis}(N - s) + F_{trans}(s) \tag{4.30}$$

As such, the simulation problem is reduced to modeling and calculating the free energy of tethered chains in *cis* and *trans* compartments independently, as shown in Figure 4.15. Equation (4.30) determines the driving force for translocation, which is proportional to the gradient of $\mathcal{F}(N, s)$ with respect to the translocation coordinate and is supplied to the FP equation as the external potential.

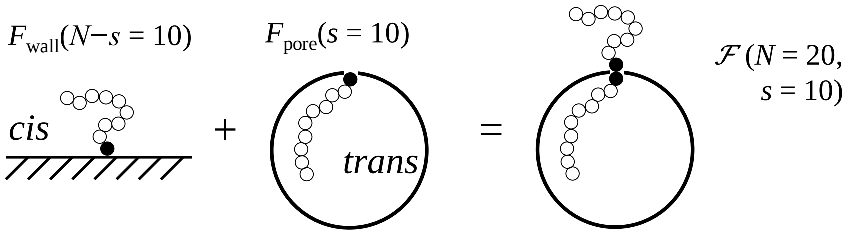

$F_{\text{wall}}(N - s = 10)$
$F_{\text{pore}}(s = 10)$
$\mathcal{F}(N = 20, s = 10)$
cis
trans
+
=

Figure 4.15: Schematic of free energy calculations. Two subsystems are simulated separately—a chain tethered to a hard wall, and a chain tethered in a spherical pore. The free energy of the translocating chain (of length N, and translocation coordinate s) is the combination of the free energy of the two sub-chains. Filled circles indicate the tethered, immovable monomer.

Simulations were performed in the mesocanonical ensemble (MCMC) [9, 10] using the incremental gauge cell method [208]. MCMC introduces a finite volume reservoir of non-bonded monomers (called the gauge cell) that exchange with the system cell by addition and deletion of monomers at the free end of the tethered chain. Use of the gauge cell method brings about two main advantages: accurate determination of the chemical potential (and thus the free energy), and more efficient sampling of the phase space. Additionally, the gauge cell limits fluctuations that would otherwise result in phase changes characteristic to an open system with unconstrained fluctuations. As such, a continuous trajectory of metastable and labile states can be stabilized, and the free energy difference can be obtained by integrating along this trajectory [1]. For this reason, MCMC has become a useful tool for studying nucleation phenomena [1, 53–55, 209], confined fluid phase behavior [50–52, 167], and adsorption deformation [56]. Chains are equilibrated with three types of MC moves: local monomer displacement, configurational bias regrow [91], and exchange with the gauge cell. The simulation scheme consisted of 400 discarded equilibration sets and 500 averaged production sets, each of 850,000 attempted MC moves. Convergence was tested by monitoring error estimates in the chemical potential using blocked statistics [247] with a runtime algorithm [248]. A detailed description of the incremental gauge cell method is given in our recent paper [208].

The polymer is modeled as a freely-jointed chain. Sequential monomers are bonded with a harmonic potential,

$$U_{\mathrm{bond}}(r) = \begin{cases} \frac{1}{2}\kappa_{\mathrm{b}}(r-r_0)^2 & \text{for } 0.5 \leq r/\sigma \leq 1.5 \\ \infty & \text{otherwise} \end{cases} \tag{4.31}$$

where r is the distance between bonded monomers, r_0 is the equilibrium bond length set to σ_{MM}, and κ_{b} is the spring constant, taken as $400\epsilon/\sigma^2$ [236]. The excluded volume effect is modeled by setting non-bonded monomers to interact via the LJ potential with monomer-monomer parameters $\epsilon_{\mathrm{MM}}/k_{\mathrm{B}} = 49.3$ K, $\sigma_{\mathrm{MM}} = 0.394$ nm, and no cut-off distance. In order to mimic good solvent conditions [80], the simulations were performed at $k_{\mathrm{B}}T/\epsilon_{\mathrm{MM}} = T^* = 8$, which is well above the theta transition temperature of LJ chains (of about $5\epsilon/k_{\mathrm{B}}$) [115]. To demonstrate the effects of confinement on the chains of maximum length of 200 monomers the pore diameter was set to $2R = 10\sigma_{\mathrm{MM}}$, which is approximately the radius of gyration of a free chain of the maximum length considered in this work. To restrict the chain's conformations in the *cis* compartment, the terminal monomer was fixed at (0, 0, $0.5\sigma_{\mathrm{MM}}$), and a hard wall repulsion is implemented by rejecting the moves where monomers crossed the plane $z = 0$. Adsorption in the *trans* compartment is captured using the site averaged solid-monomer LJ potential integrated over the spherical layer of adsorption centers [205], with $\sigma_{\mathrm{SM}} = 0.33$ nm. The interaction strength between the pore wall and the monomer units was varied as the ratio of solid-monomer and monomer-monomer interaction energies $\xi = \epsilon_{\mathrm{SM}}/\epsilon_{\mathrm{MM}} = 0$, 0.5, 1, 1.1, 1.3, 1.6, 2.0, 2.5, and 3.0. This range covers the regions of steric repulsion, weak adsorption, and strong adsorption. At $\xi \sim 2$, the interaction parameters roughly correspond to alkane adsorption on a silica surface [81, 164]. Because the free energy is the sum of the incremental chemical values, it is sensitive to these values for short chains. For this reason, a more detailed potential was used for short chains, $n =$ 1 to 20. This potential explicitly accounts for the translocation opening in the pore wall by partially integrating over the pore spherical surface [2]. An opening of $1\sigma_{\mathrm{MM}}$ was created, and the terminal bead of the chain tethered a distance of $0.5\sigma_{\mathrm{MM}}$ from the system boundary. This potential is shown as an xz-plane projection in Figure 4.16. The value of incremental chemical potential calculated with the more detailed and simple solid-monomer potentials are asymptotically identical for $n > 10$, thus the more detailed potential implements short chain "corrections" to the free energy.

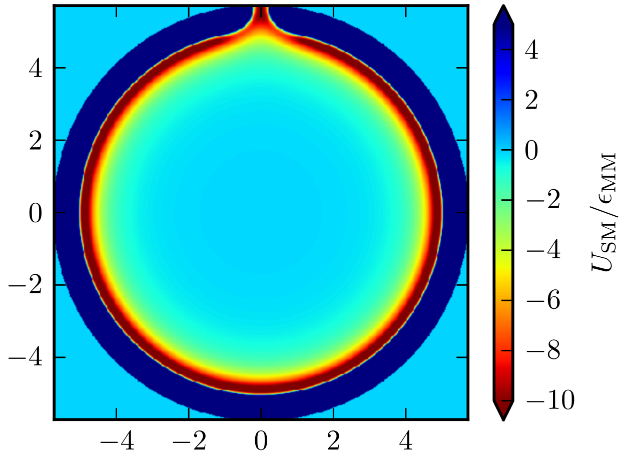


Figure 4.16: Map of solid-monomer potential energy on the $y = 0$ plane, for $\xi = 1.6$. One end of the chain is tethered at (0, 0, $R/\sigma - 0.5$), which corresponds to the small attractive opening at the top of this plot.

Translocation dynamics are studied using the FP formalism following the approach of Sung and Park [263], and Muthukumar [65], as implemented in our recent theoretical work [261]. We assume that chain relaxation times are shorter than translocation times (i.e. that the *cis* and *trans* sub-chains can be considered as independent, equilibrated chains). In this case, we assume that the chain diffuses between the *cis* and *trans* compartments with the drift term proportional to the free energy gradient. The FP equation governing the translocation dynamics is

$$\frac{\partial}{\partial \tau} W(s,\tau) = \frac{\partial}{\partial s}\left[\frac{\partial \mathcal{F}(N,s)}{\partial s} W(s,\tau) + \frac{\partial}{\partial s} W(s,\tau)\right] \tag{4.32}$$

where $W(s,\tau)$ is the probability of a polymer chain of length N, with one initial segment in the *trans* compartment (and $N-1$ segments in *cis* compartment) at time $\tau = 0$, to have s segments in *trans* compartment at time τ. Note that τ is the dimensionless time; it is proportional to the local friction coefficient k_0, which for a homopolymer is assumed to be independent of the degree of translocation [263]. For a complete discussion of Equation (4.32), see Ref. [261].

4.2.3 Results: Free Energy

The incremental chemical potential, μ_{inc}, of two subsystems: a chain tethered to a hard wall and a chain tethered in a spherical adsorbing pore (Figure 4.15), was calculated using the IGC method [208]. The adsorption potential of the pore was varied from weak to strong interaction. The values of μ_{inc} as a function of chain length are shown in Figure 4.17 (top). These values were reduced by the reference state, an ideal chain with only harmonic bonds in the same system (i.e. tethered in *cis*/*trans* compartment), to better illustrate confinement effects on free energy. The dashed line indicates the hard wall *cis* sub-chain. For long chains of $n > 10$, μ_{inc} is constant, as is expected for expanded coil chains well above the theta temperature [80]. The adsorption potential of was varied from $\xi = 0$ to 3.0. The incremental chemical potential of the *trans* chains rises exponentially with chain length because of increasing steric hindrance. The absolute values of μ_{inc} shift to more negative values at adsorption strength increases, as favorable adsorption interactions offset the confinement entropy loss. To obtain the free energy of the tethered chain, the incremental chemical potential is summed over the chain length n [208]. The result is shown in Figure (4.17) (bottom). The *cis* sub-chain reduced free energy profile is increasing and linear, because of its positive and constant incremental chemical potential. The shape of the reduced free energy of the tethered chain in the adsorbing pore is more complex: depending on the adsorption potential strength, the free energy can be an increasing, decreasing, or non-monotonic (i.e. with a minimum) function of the chain length. Since the incremental chemical potentials always increases with the chain length, the minimum is due to the negative contribution from short chains. For the weak adsorption potentials up to ξ = 0.5 the free energy is increasing, as the respective incremental chemical potentials are all positive. As the adsorption potential increases, the free energy becomes non-monotonic: most of monomers of the short tethered chains are adsorbed at the pore wall, and the gain in enthalpy compensates for the loss of entropy. For chains shorter than 200, this regime is seen in the range of adsorption potentials 1.0 to 2.5. At the strongest potential of ξ = 3.0 considered, the free energy is still decreasing at n = 200, but it would achieve a minimum and then increase as the chain grows further and the pore becomes crowded with monomers. One of the important qualitative conclusions derived from these calculations is the observation that in contrast to the chains tethered to non-adsorbing hard wall, the free energy of chains confined to the adsorbing and confining pore is a non-linear function of the chain length.

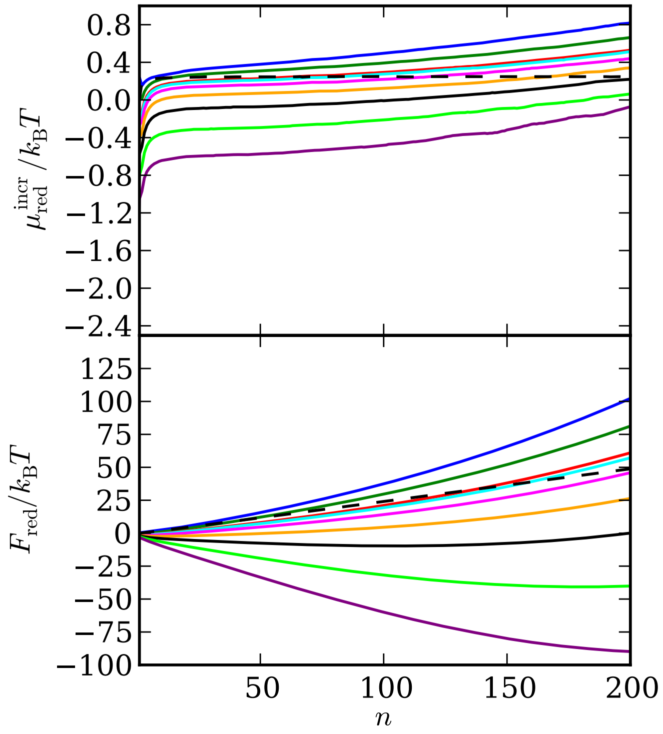


Figure 4.17: (Top) The reduced incremental chemical potential of tethered chains in pores of varying adsorptive strength, as a function of chain length n. (Bottom) The reduced free energy of the same chains. Adsorptive strength is described as ξ, which is the ratio of solid-monomer LJ parameter to the monomer-monomer parameter. Solids lines indicate chains in pores with $\xi = 0$, 0.5, 1, 1.1, 1.3, 1.6, 2, 2.5, and 3, from most positive to most negative, respectively. Dashed lines indicate a chain tethered to a hard wall (i.e. the *cis* subsystem).

The free energy landscape of the translocating chain is calculated according to Equation (4.30) as the sum of *cis*- and *trans*- subchain free energies shown in Figure 4.17. The free energy landscape of the chain with $N = 200$ total monomers is plotted as function of the degree of translocation in Figure 4.18. Note that the free energy terms F_{cis} and F_{trans} of Equation (4.30) include the contribution from the harmonic bonds; that is, they are not the reduced values. As such, the values of $\mathcal{F}$ in Figure 4.18 were shifted upwards by $F_{cis}(N)$ so that the depth of the free energy wells can be clearly discerned. For the weakest confinement interactions of 0 and 0.5, the free energy is positive and monotonically increases forming an uphill landscape. This suggests unfavorable conditions for translocation, which is intuitively obvious ($\xi = 0$ refers to a spherical pore with no favorable enthalpic interactions at all; a chain translocating into this pore would lose entropy and thus be unfavorable). Increasing the interaction potential to the range of 1.0 to 1.6, the free energy landscape becomes broader, and exhibits a minimum. Minima of $\xi = 1$ and 1.1 occur for short *trans* sub-chains (s ¡ 20), while 1.3 and 1.6 have a more broad landscape, and significantly more negative minima. The minima correspond to a metastable chain configuration composed of *trans*- and *cis*- fragments, which we call a flower configuration. In such a configuration the enthalpic gain and entropic loss balance each other. As we show below, these metastable configurations represent the stall points and play an important role in translocation dynamics by fsignificantly increasing the translocation time. Above this range ($2 \leq \xi \leq 3$), the favorable adsorption interactions are dominant and the free energy monotonically decreases forming a downhill landscape, which favors translocation, up to the length of 200 tested in our simulations. However, additional loading would begin to increase chemical potential as steric hindrance rises.

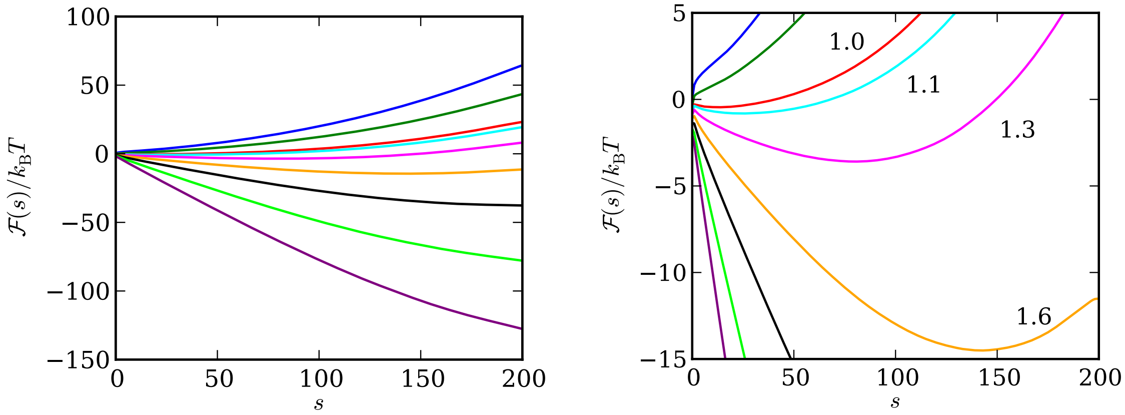


Figure 4.18: (Left) The free energy landscape for an $N = 200$ chain translocating into the pore, as function of length of chain on the *trans* side s, calculated from Equation (4.30). The free energy here includes the harmonic bond contribution, and is shifted by $F_{cis}(N)$ for clarity. Lines correspond to $\xi = 0$ to 3 from top to bottom. (Right) Scaled to highlight free energy minima.

4.2.4 Results: Translocation Dynamics

The dynamics of translocation was determined by numerical solution of Equation (4.32). Several relevant quantities can be extracted from Equation (4.32). Of particular interest is the translocation time probability distribution. The distribution of translocation times is a measurable experimental quantity, and can be interpreted to gain knowledge of the nature of translocation [283]. When using the FP formalism, it is found from the corresponding probability flux,

$$P_{\text{in}}(\tau) = J|_{s=N} = -\left[k_0 \frac{\partial \mathcal{F}(N,s)}{\partial s} W(s,\tau) + k_0 \frac{\partial}{\partial s} W(s,\tau)\right]_{s=N} \tag{4.33}$$

where k_0 is the local friction coefficient. Similarly, the probability distribution of failed translocation events, when the chain comes out of the pore, is represented by the negative flux at $s = 0$. The normalized probability distributions are displayed in Figure 4.19. The normalizing factor is the total probability of successful translocation into the pore, given by

$$P_{\text{in}}^{\text{total}} = \int_0^\infty P_{\text{in}}(\tau)\,\mathrm{d}\tau \tag{4.34}$$

In practice, this is calculated by summation of the resulting series, multiplied by the selected $d\tau$. The lengths of the series were chosen so that final value can be approximated as zero. The first and sharpest peak is that of the strong adsorption case, $\xi = 3$, followed by 2.5. This is expected, as the strong adsorption potential forces quick translocation. However, the next two peaks are the weakest adsorption cases of $\xi = 0$ and 0.5, followed by the strong 2.0, and then by very broad distributions of the intermediate cases. This behavior is clearly governed by the shape of free energy profile (Figure 4.18). The broad distributions are the result of the minima, or stall points, found in the free energy profile. At the translocation coordinate where the minimum is found, the chain is in a metastable state, partially threaded through the translocation pore with s monomers in the adsorbing pore and $N - s$ monomers outside. With no driving force, a successful translocation must rely on stochastic fluctuations to escape the metastable position; the time required is proportional to $exp(-\Delta\mathcal{F}/k_{\text{B}}T)$, where here $\Delta\mathcal{F} = \mathcal{F}_{\text{min}} - \mathcal{F}(s = N)$. Thus, the deeper the free energy well, the longer the polymer will take to escape it and complete translocation. This results in the very broad time distributions for potentials where a minimum is found, namely $\xi =$ 1, 1.1, 1.3, and 1.6. Interestingly, we find that when the adsorption potential is nonexistent or very weak, successful translocation still occurs fairly quickly (the peaks of $\xi = 0$ and 0.5 are both at a smaller time than $\xi = 2$). This effect was also observed in our previous theoretical work [261]. These weak potentials both present a monotonically increasing free energy profile; translocation occurs quickly because there is no energetic advantage to slow translocation. Even though the unfavorable translocation is fast, its total probability is quite low (Figure 4.20). It is worth to note that the solution to the FP equation (4.32) is the same for monotonically decreasing and monotonically increasing free energy, save for the prefactor [261]. This produces probability distributions of the same shape for the weak potential cases of $\xi = 0$, 0.5 and the strong potentials $\xi = 2$, 2.5, 3, albeit with a much different prefactor.

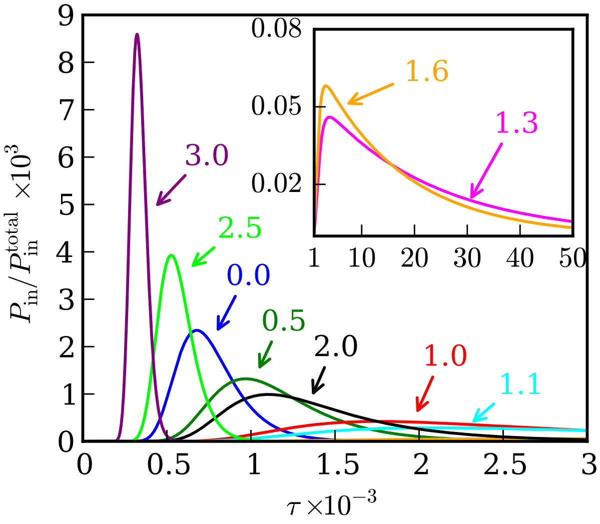


Figure 4.19: The normalized probability distribution function of 200-mer chain translocating into the adsorbing pore. The inset is a magnified view of two systems: $\xi = 1.3$ and 1.6, which have extremely broad distributions.

It is important to note that the distributions presented in Figure 4.19 are normalized by the total translocation probability (given in Equation (4.34)). These values are given as a function of the length of the translocating chain for the various adsorption potentials in Figure 4.20. Short chains have a high probability of translocation for all potentials.

Similarly, increasing adsorption potential always increases translocation probability. As the chain length increases for translocation into pores with weak or no potential, the total probability of transition approaches zero. Chains driven by intermediate adsorption potentials (1.6, 2) have approximately constant total probability with respect to chain length, while the strongest potentials' (2.5, 3.0) total probability increases with chain length. This effect is found when the adsorption enthalpy is the dominate contribution to the free energy, provided the availability an excess of adsorption sites per monomer [207]. In this case, the adsorption force is strong enough that adding additional monomers (i.e. increasing the length of translocating chain) decreases the total free energy, resulting in a more likely translocation. A similar effect is found in interaction polymer chromatography, where retention increases with chain length because of increasing favorable interaction with the substrate [284]. For chains with broad time distributions at $N = 200$ (Figure 4.19), most have a total probability of success of zero. The exception is $\xi = 1.6$, whose overall probability of translocation is ∼0.14. This suggests that broad translocation time distributions do not necessarily prevent a successful translocation event.

The most experimentally relevant property of a translocation process is the average time of successful translocation. This is given by

$$\langle \tau_{\text{in}} \rangle = \frac{\int_0^\infty \tau P_{\text{in}}(\tau)\, \mathrm{d}\tau}{\int_0^\infty P_{\text{in}}(\tau)\, \mathrm{d}\tau} \tag{4.35}$$

The average translocation times of the LJ chain moving into an adsorbing pore are shown in Figure 4.21. The three regimes of adsorption strength are reflected. The strongly adsorbing pores exhibit translocation times that are exponential in nature, with a scaling exponent of ∼1.37. Similarly detailed LD studies found nearly identical scaling exponents for translocation with strong forcing [285]. It is important to realize that if simulations of longer chains were performed, the scaling relation would no longer apply. For a $10\sigma_{\text{MM}}$ diameter pore, full surface coverage is achieved with about 220 adsorbed monomers [208]. If a chain longer than this were translocating into the strongly adsorbing pore, the additional beads would experience a higher incremental chemical potential as they fill the pore volume, as they no longer directly interact with the adsorptive walls. Translocation into a weakly adsorbing pore ex-

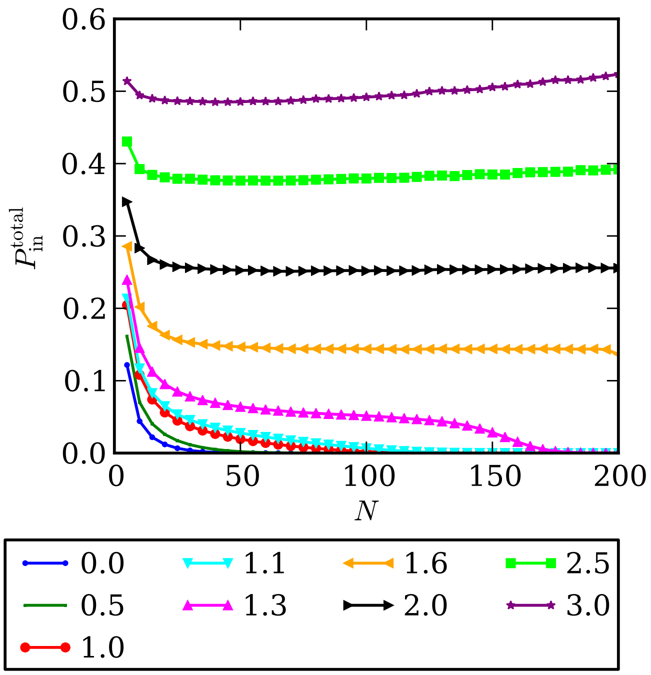


Figure 4.20: The total probability of success for N-mers translocating into the adsorbing pore (Equation (4.34)). Note that for long chains with unfavorable adsorption potentials, the probability tends to zero. Legend indicates adsorptive strength of spherical pore, ξ.

hibits a similar scaling behavior for the range of chain lengths tested, albeit with a smaller exponent. In addition to the faster per-monomer translocation in these weakly adsorbing systems, the prefactor is the same order as the strongly adsorbing systems. As Figure 4.19 also indicates, unforced chains may translocate faster than strongly forced chains. This apparent contradiction can be explained with two points: first, it is important to remember that the data presented represents the average times of successful translocations, and that the probability of such an event is still quite low for unforced or weakly forced systems (see Figure 4.20). Second, as time increases, there are significantly more chances for the chain to escape back to the lower energy *cis* state. In other words, if a successful translocation is to occur, it would have the best odds if it proceeded quickly. For the intermediate systems of ξ = 1.3 and 1.6, i.e. the ones exhibiting a significant free energy minimum for the N = 200 chain, the average translocation time shows a slow then a fast regime (as measure of translocation time per monomer scaling). The slow regime corresponds to a downhill free energy profile, akin to ξ = 2–3 in Figure 4.18. At these shorter chain lengths, there are no minima present on the free energy landscape, and translocation proceeds as with the strongly forced chains. This is followed by the fast regime plateau at larger N, representing translocation against an entropic barrier. The transition between these regimes is marked by even slower translocation times (between $N = 115$–160 for $\xi = 1.3$ and $N = 170$–200 for $\xi = 1.6$). In this range, a surmountable free energy minimum is present. In other words, the chain reached a length where a minimum is present; however the probability of random fluctuations pushing the translocation to competition is nonzero. The chain is metastable at the minimum, and translocation times go up accordingly. The plateau in average translocation times begins were the total probability of successful translocation (Figure 4.20) goes to zero, and the previous 'uphill' arguments apply. The two smaller values of the intermediate range ($\xi = 1$, 1.1) appear similar to the 'uphill' translocations times of $\xi = 0$, 0.5, but with a larger prefactor. The minima in these two systems are relatively small, and occur at short chain length (Figure 4.18, right). In all cases of the 'fast' translocation against a large free barrier, the probability of a successful translocation approaches zero (see Figure 4.20).

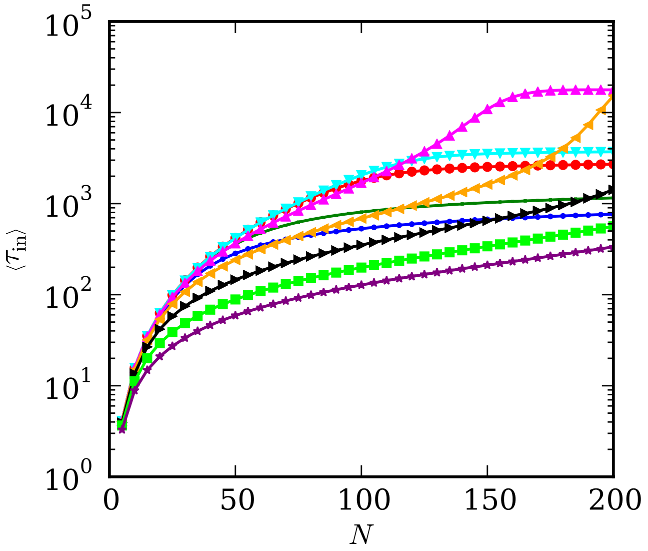


Figure 4.21: Average translocation time into adsorbing pore of varying adsorption strength ξ, as function of chain length N. Refer to the legend of Figure 4.20.

The non-monotonic behavior of translocation times with respect to adsorption strength is an interesting observation of this work. Figure 4.22 plots the average translocation time as a function of adsorption strength for several lengths of chains. A prominent peak in translocation time is clear for long chains in pores in the intermediate range of potentials, where the free energy profile has a local minima corresponding to a transient metastable state. The effect is more pronounced for longer chains, since such chains must move further from the metastable state to fully translocate. Short chains are not affected because they successfully translocate before they reach the minimum. The peak arises due to the long time needed to escape the stall point at the free energy well. It falls as the adsorption potential increases, as expected. For the longest simulated chain of $N = 200$, the strongest metastable state (i.e. with the deepest free energy well along the translocation coordinate s) is found between $1.3 < \xi < 1.6$. As the chain length decreases, the position of the minima shifts to lower ξ, with $N = 150$ exhibiting a maxima near $\xi = 1.3$, and $N = 100$ at $\xi = 1.3$. The shift in the maximum for τ with changing chain length is due the additional entropic penalty from confinement of longer chains. In other words, for a given chain to reach its partially threaded metastable state, increasing length must be compensated by increasing adsorption potential. The collapse of the long chain length translocation-in times at similar values occurs where conditions are unfavorable to translocation, $\xi < 1.0$. Figure 4.23 shows the probability of successful translocation for the same chains as shown in Figure 4.22. We see that the probabilities of successful translocation into weakly adsorbing pores are mere zero for all but the shortest chains. As the likelihood of translocation begins to increase, so does the time required for successful translocation. After the adsorption potential reaches a certain strength (which depends on chain length), translocation times fall as the potential increases, as expected. This is the strong adsorption region.

4.2.5 Conclusions

We studied the dynamics of a single chain translocating into an adsorbing pore by Monte Carlo calculation of its free energy landscape using the incremental gauge cell method [208], and application of the Fokker-Planck equation that mimics the translocation process as the diffusion along the free energy landscape. The respective free energy landscape for partially translocated chains was obtained by combining two independent *cis*- and *trans*- tethered chains. It was found that the free energy landscape of a translocating chain exhibits three characteristic behaviors: ‘uphill,’ or monotonically increasing free energy as translocation

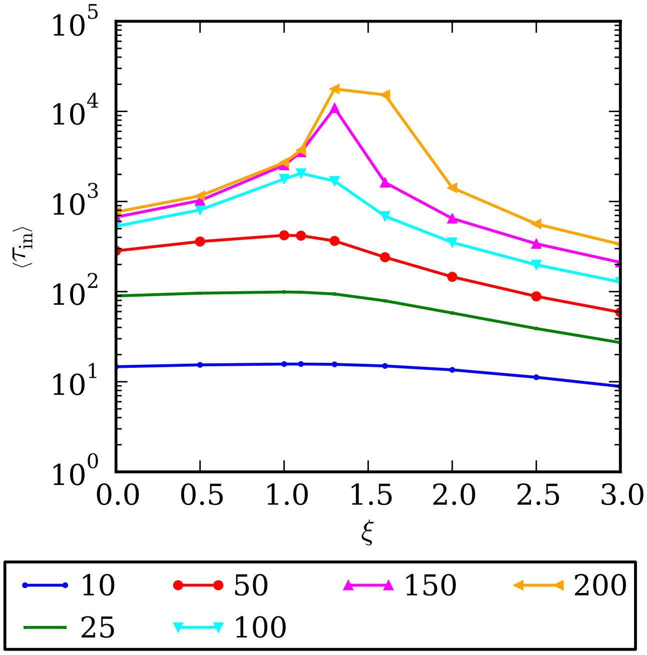


Figure 4.22: Average translocation time into adsorbing pore, as function of adsorption strength for several fixed lengths N. Legend indicates chain length of translocation, N.

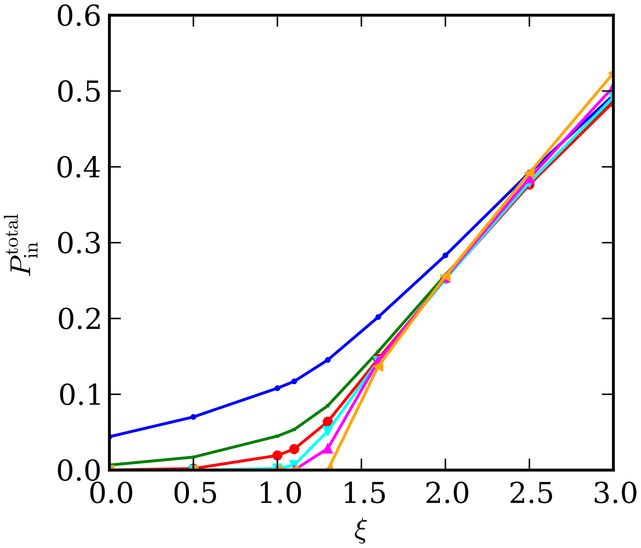


Figure 4.23: The total probability of successful translocation for various chain lengths, as a function of adsorption strength.

proceeds, ‘downhill,’ or decreasing free energy, and “concave” with a local minimum. This free energy minimum corresponds to a transient metastable state, where the chain comfortably balances adsorption and confinement in a flower configuration being partially adsorbed in the pore. These three energy regimes translate into very different translocation behaviors. Uphill diffusion, being unfavorable, has a very low probability of completing translocation, except for the shortest chains. However, translocation proceeds quickly if it does occur. To conceptualize this phenomenon, we can invoke a momentum analogy. One can imagine a steep hill (the energy barrier) with two climbers, one fast and one slow. The faster climber is more likely to succeed, as the slower climber has more chances to fall backwards. Downhill diffusion results in more intuitive results, with translocation times increasing with adsorption strength. This region of strong forcing was predicted quantitatively by Muthukumar [65]. The work of Krasilnikov et al. [286] finds experimental evidence of the non-monotonic dependence of residence times on molecular weight: they found that polyethylene glycol (PEG) residence times in an α-hemolysin pore increased with chain length, up to a molecular weight of 3000, and then decreased.

Concave translocation landscapes with free energy minima result in a very broad time distribution, up to several orders of magnitude more broad than a simple uphill or downhill case (see Figure 4.19, inset). When a chain that is favorably moving towards the pore reaches its free energy minimum, forward progress effectively stops as the chain fluctuates within its free energy well. At this point, successful translocation means overcoming the energy barrier from the minimum to $\mathcal{F}(s = N)$. Time for successful translocation slows accordingly. Though slow, broad translocation time distributions were found to have a non-zero probability, suggesting that addition complexities may arise in experimental systems. These phenomena highlight the importance of the interplay between enthalpy and entropy during the adsorption of polymers.

Recently the validity of the FP method (and its assumptions) has been questioned [264, 265, 273, 278]. The main concern is the result that unforced FP translocation scales as N^2, the same as (or less than) the Rouse time (the characteristic time for a free ideal chain to diffuse a distance of the order of its radius of gyration) [287], which scales as $N^{2\nu+1}$, with ν being the Flory exponent (0.5 for an ideal chain and 0.588 for a random coil). We don't believe this scaling interpretation is particularly useful for the system considered in this work. First, chain lengths considered only span two orders of magnitude, hardly the length scales relevant for true scaling analysis. Where scaling laws could fit translocation data, i.e. the downhill free energy regime created by strong adsorption potential, the free energy cannot scale with N, as continued pore loading will result in a dramatic increase in free energy. In addition, previous work has found the FP approach [65] to accurately describe forced translocation values in experiment [283, 288, 289].

The results of this MC study are qualitatively similar to previous work where the free energy landscape was calculating using self-consistent field theory (SCFT) [261]. Most notably, the prediction of free energy minima, and the corresponding increase of translocation times, was found using SCFT. One notable difference is our previous work with SCFT did not probe overly strong adsorption potentials, where translocation times of strongly-adsorbing systems are faster than the translocation time of the weakly-adsorbing systems. While SCFT can make many useful predictions of ideal and excluded volume chains, the real strength of the current approach is the use of a chemically accurate model, which can be upgraded for more complex pore geometries and inhomogeneous polymers, including random and block copolymers. Many factors influence translocation in experiment: temperature, chemical and substrate heterogeneities, solvent composition, hydrodynamic and electrostatic effects, and so on. Many of these variables can be incorporated with relative ease into a molecular simulation. Use of such accurate and realistic potentials may help to resolve the many still open questions about polymer translocation.

The main methodological outcome of this work is the demonstration of the efficiency of the suggested simulation technique (a combination of MC calculation of the free energy landscape by use of the incremental gauge cell method, coupled with application of the FP equation) for modeling translocation dynamics. This technique can be applied not only to adsorption driven translocation, but also to any forced or diffusive translocation process in various confining geometries, including the escape of initially confined chain from a small cis compartment to a large trans compartment. This is enforced by the fact that two entirely different simulation techniques, SCFT in Ref. [261], and incremental gauge cell MC simulation in this work, bring about similar quantitative conclusions about the specifics of the adsorption-driven translocation is by itself a valuable theoretical result.

Finally, these results suggest an interesting complication to the traditional picture of polymer chromatography. The retention time of a polymer in a column, and thus the quality of chromatographic separation, is found by determining the partition coefficient for a given polymer/solvent and substrate. Traditionally, the partition coefficient is modeled as a function of the difference of free energy upon adsorption from the mobile "free" state to the adsorbed state. When this difference is zero, steric repulsion is balanced by attractive enthalpic interactions, and retention becomes independent of chain length. This condition is called the critical point of adsorption and important in the separation of structured polymers (e.g. functionalized polymers and copolymers) [284]. This work suggests that an intermediate state, the partially adsorbed polymer, exist. If adsorption interactions are favorable and the pore imposes a steric penalty as the polymer fills the pore, a metastable partially adsorbed state is reached. This has clear implications in the separation of high weight polymers. We intend to examine this scenario in future work.

4.3 Gradient Elution Polymer Chromatography

4.3.1 Introduction

The retention factor for polymeric solutes usually exponentially increases with the molar mass as a result of multiple attachment mechanism of polymer adsorption (the so-called Martin rule [290]). As the result, gradient elution should be a preferred way for separation of such molecules. Nevertheless, until recently, two isocratic methods, SEC and liquid chromatography at critical conditions (LCCC), have been the most popular liquid chromatography techniques employed for polymer characterization [12]. There are two main reasons for such exception. First, both techniques do not follow the general rule: in SEC, elution time decreases with molar mass (or, more precisely, molecular size), while in LCCC elution time is molar mass-independent. Secondly, both approaches are supported by molecular-statistical theory which provides a clear understanding of the separation mechanism and can explain and sometimes even predict the selectivity of separation [291]. Thus, in SEC, steric interaction alone produces a separation by molecular size and can be described by the conformational entropy loss of a polymer chain within a pore of a comparable size. In LCCC, such a loss is exactly compensated by enthalpy gain of the chain subjected to attractive interaction with a wall of the pore, which produces molar mass independent retention [284]. Isocratic elution in any chromatographic mode including those employed in SEC or LCCC, in principle, can be quantitatively described by a molecular-statistical model of Gaussian polymer chain penetrating a pore with a simple geometry (e.g. slit-like or cylindrical) without (SEC) or with (LCCC) attractive interaction with the walls [292]. Thus, the theory of LCCC relates the thermodynamic distribution coefficient K describing such elution, to the molecular parameters of the polymer chain, geometry of the pore and the chromatographic conditions affecting the energy of polymer-stationary phase interaction. Such a theory was developed originally for homopolymers [293], and then extended for polymers with more complex molecular structures such as telechelic linear [294] and star-shaped [295, 296] polymers, block-copolymers [295–297], statistical copolymers [298] and macrocycles with various topologies [299].

In spite of significant success in numerous applications (see [284] and references therein), both isocratic techniques have obvious limitations. Thus, separation by molecular size (SEC) does not provide information about various molecular heterogeneities of complex polymers and copolymers, e.g. the distributions by chemical composition or functional groups, while LCCC is effective mostly in characterization of oligomers. The generalization of the theory to gradient elution would significantly extend the capability of liquid chromatography in characterization of polymers with complex molecular structure. It was done for the first time in papers [13–15], where a simplified asymptotic equation for distribution coefficient K was used in mass balance equation for gradient elution. The most important result from the theory was the conclusion that in gradient mode polymer with high enough molar mass can elute close to critical conditions, i.e., with eluent composition close to its critical point of adsorption (CPA). In other words, a unique feature of LCCC, molar mass-independent elution, can be inherent also in gradient mode, so that the benefits of LCCC can be achieved without the limitation of the isocratic elution. This theoretical conclusion was confirmed experimentally for homopolymers and statistical copolymers [13, 298]. A similar result was obtained later in papers [300–302] using another asymptotic expression for K, which is valid only in case of wide pores (i.e. when a pore is wider than the size of a polymer chain). In fact, this infers a conclusion that gradient elution at CPA (if such a point exists for a given system) for high molar mass polymers has a general nature and does not depend on a specific model of polymer retention.

So far, the efforts to extend the molecular-statistical theory to gradient elution have been made only for the case of homopolymer and statistical copolymer which is described by the same equations [298]. The goal of this section is to extend the theory to polymers with other molecular structures, such as telechelic polymers (Section 4.3.3) and block copolymers (Section 4.3.4). In the first section (4.3.2), we consider the most general approach to the theory of gradient elution of homopolymers and analyze possible asymptotic approximations. The theoretical results are compared with the experimental data for homopolymers.

4.3.2 General Approach for Homopolymers

Separations in polymer chromatography are governed by a simple "master" equation [303],

$$V_R = V_i + KV_p \tag{4.36}$$

where V_R is the retention volume, V_i is the interstitial volume, V_p is the pore volume available for adsorption and K is the partition (distribution) coefficient. The value of K determines the chromatographic regime: if $K < 1$, there is a free energy penalty for confinement of the solute polymer, thus the solute elutes close to the solvent band and the column operates in size-exclusion mode, if $K = 1$, the column is in the critical mode where exclusion and adsorption are balanced, and if $K > 1$, the adsorption interactions are dominate and the solute is retained longer [303]. Dividing Equation (4.36) by the eluent flow rate F yields the analogous equation in time,

$$t_R = t_i + Kt_p \tag{4.37}$$

Together with an expression for K, Equation (4.36) or (4.37) is all that is needed to predict or interpret isocratic separations.

Several models have been proposed for modeling gradient elution chromatography. The linear solvent strength model [304] and quadratic solvent strength model [305] assume a direct dependence of the partition coefficient K on the solvent strength Φ (here, Φ is the percentage of the strong solvent in a binary elution). More sound models are based on the statistical theory of ideal chains (that is, random flight statistics). In these cases, one can obtain a function for the partition coefficient that can be integrated over the chromatographic column to find the time a solute takes to elute under a solvent gradient. For a column operating with a linear solvent gradient, the velocity of a solute (whose position is x in a column of L length) can be described using a simple balance equation,

$$\frac{dx}{dt} = \frac{L}{t_R} \tag{4.38}$$

Equation (4.38) simply states that the retention time of a given solute with be reached at the end of the column. Substituting in Equation (4.37) and taking the inverse, Equation (4.38) becomes,

$$\frac{dt}{d\tilde{x}} = t_i + K(t, \tilde{x})t_p \tag{4.39}$$

where the partition coefficient is shown as a function of time and position and $\tilde{x} = x/L$ is the dimensionless distance along the column. This equation can be solved for as an ordinary differential equation (ODE) where t goes from 0 to t_G (the time of elution for a solute in a gradient-mode column), and $\tilde{x}$ goes from 0 to 1. Equation (4.39) is well-suited to numerical integration.

An expression for K can be obtained by selecting an appropriate model for the polymer/solvent and polymer/substrate interaction. If we model the homopolymer as an ideal (Gaussian) chain, the change of conformational energy can be determined by solving the diffusion equation with appropriate boundary conditions [291]. Confinement and adsorption effects are captured by applying "sticky walls" boundary condition. This model is advantageous because is it an analog of the heat equation, and thus can utilize the considerable amount of work done. Skvortsov and Gorbunov [293] obtained a general solution for this model. This expression for K is given as

$$K = \sum_{m=1(m\text{ odd})}^{\infty} \frac{2\lambda^2 \exp\left(-\alpha_m^2 g^2\right)}{\alpha_m^2 \left[\lambda(\lambda+1) + \alpha_m^2\right]} \tag{4.40}$$

where α_m are the characteristic roots of

$$\alpha_m = \arctan\left(\lambda/\alpha_m\right) + (m-1)\pi/2 \tag{4.41}$$

Here, $g = 2R_\mathrm{G}/D$, with R_G the radius of gyration of the solute and D the mean pore diameter, is the dimensionless size parameter; when $g > 1$, the polymer is larger than the pore (e.g. in the narrow pore regime), and conversely, when $g < 1$, the polymer is smaller than the pore and in the wide pore regime. λ is a dimensionless adsorption interaction parameter; a negative value of λ indicates that the repeat polymer units are attracted to the pore walls; and conversely repulsed when λ is positive. Thus, critical conditions occur at $\lambda = 0$. Because of the relative complexity of Equation (4.53), asymptotic values in the limits of strong adsorption, wide pores, or narrow pores are often used in analysis. In order to apply Equation (4.53) to gradient elution chromatography, we must integrate the balance ODE (Equation (4.39)) over the length of the column. Changing solvent conditions imply that λ will vary over the length of the column. The choice of the solvent model is a complex one; as the parameter λ captures the combined solvent/solute/substrate interaction and is thus a function of many variables. The simplest approach is to assume a linear dependence near the critical composition,

$$\lambda(\Phi) \approx \lambda(\Phi_\mathrm{cr}) + \left.\frac{d\lambda}{d\Phi}\right|_{\Phi_\mathrm{cr}} (\Phi - \Phi_\mathrm{cr}) \tag{4.42}$$

By definition, $\lambda(\Phi_\mathrm{cr}) = 0$, so the only remaining parameter is the derivative $d\lambda/d\Phi$ taken at $\Phi = \Phi_\mathrm{cr}$. Finally, we account for the solvent concentration difference along the length of the column during the gradient elution,

$$\Phi = \Phi_0 + \frac{d\Phi}{dt}\left[\left(t - \left(t_i + t_p\right)\tilde{x}\right)\right] \tag{4.43}$$

Here, Φ_0 is the initial solvent concentration at the start of the gradient and $d\Phi/dt$ is simply the gradient rate. Together, Equations (4.39), (4.53), (4.42), and (4.43) can be solved numerically to obtain the time of gradient elution, tG. Any standard ODE solver can be used; we selected the Python extension SciPy and its integration pack [306]. A mesh of 500 points on the *tildex* coordinate was used. When integrating, the general solution for K is solved to a precision of $\leq 10^{-4}$ for each step.

The benefits to solving the general equation directly are clear. Asymptotic equations are useful as can be used to solve Equation (4.39) analytically. However, they are subject to specific constraints. This is of concern during a gradient elution, as the polymer analyte will transition from adsorption to exclusion regime. Brun [14, 15] proposed an asymptotic expression for the partition coefficient as a first-order approximation of the statistical ideal chain model of above, for polymer near the critical point of adsorption $\lambda = 0$. It can be expressed as

$$K \approxeq \exp\left(-\lambda g^2\right) \tag{4.44}$$

When solved for a gradient elution using Equations (4.42) and (4.43), a simple analytical equation with two parameters, the critical composition and a lumped parameter that describes how close a polymer is to the critical point, was obtained for the gradient time of elution. This solution was found to accurately capture the elution times of polystyrene and PMMA [13], as well as statistical copolymers when assuming a single effective critical composition exists [298].

Recently, Bashir and coworkers obtained an analytical solution for adsorbing wide pores [300]. Their solution is based on the partition coefficient for wide pores, $R_{\mathrm{G}} < D/2$, found by Gorbunov and Skvortsov [295],

$$K \approxeq 1 - \frac{2}{\sqrt{\pi}} g + \frac{1}{\lambda}\left[1 - Y(\lambda g)\right] \tag{4.45}$$

$$Y(x) = \exp\left(x^2\right) \operatorname{erfc}\left(-x\right) \tag{4.46}$$

A solution to the ODE using this expression for K was found, although it is not a simple expression. Note that both asymptotic cases above, near-critical (4.44) and wide-pore (4.45) can be solved numerical in the same fashion as the general solution (4.53); in both cases the numerical approach was found to be identical to their respective analytical solutions. This direct approach is a strength of this method; any expression for K, as long as it's a function of solvent strength, can be simply integrated. This approach of direct integration of the general equation gives the most accurate prediction of t_G, as its only limitations are those of the model itself.

Additionally, an extra term can be included in (4.44). Gorbunov and Skvortsov [292] determined this asymptotic condition, and is valid for near-critical adsorption, narrow pore cases,

$$K \approxeq \exp\left(\frac{\lambda^2 g^2}{3} - \lambda g^2\right) \tag{4.47}$$

This equation was investigated for the possibility it may produce an analytical solution when integrating Equation (4.39), since it is relatively simple, and increases accuracy over the near-critical expression (4.44) (see Figure 4.25). However, a simple closed-form expression could not be obtained.

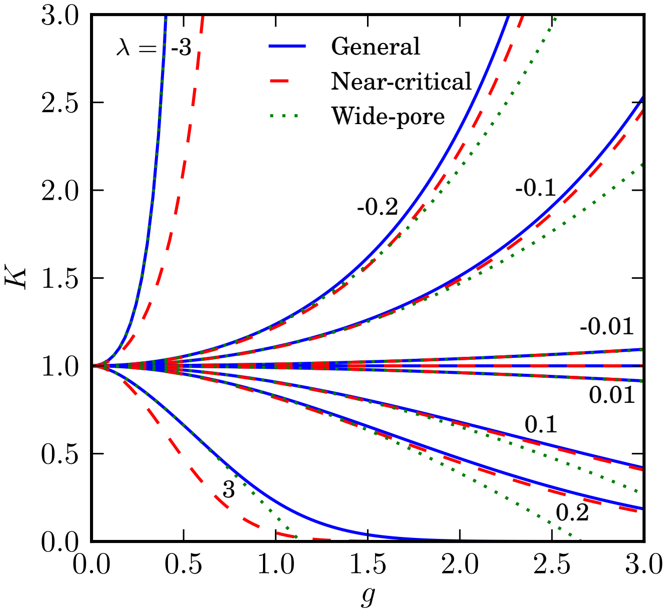


Figure 4.24: Dependence of partition coefficient K on polymer size g, calculated using the near-critical expression (4.44), wide-pore expression (4.45), and the Skvortsov and Gorbunov's general solution (4.53), for various values of λ. The approximate values are found to coincide in the vicinity of $\lambda \approx 0$.

We compared the near-critical and wide-pore asymptotic expressions for the partition coefficient to the accurate general solution of Skvortsov and Gorbunov, over a range of pore sizes and adsorption strength in Figure 4.24. The wide-pore expression is found to coincide with the general solution for most $g < 1$. As expected, it becomes progressively less accurate as g increases (i.e. transitioning to a narrow-pore regime). The near-critical expression also behaves as expected; it is least accurate in the wide pore regime, but becomes progressively more accurate as λ approaches the critical point of adsorption. Its accuracy is much less affected by pore size regime than the wide-pore expression, and remains close to the general expression in the narrow-pore regime where $\lambda \approx 0 \pm 0.2$.

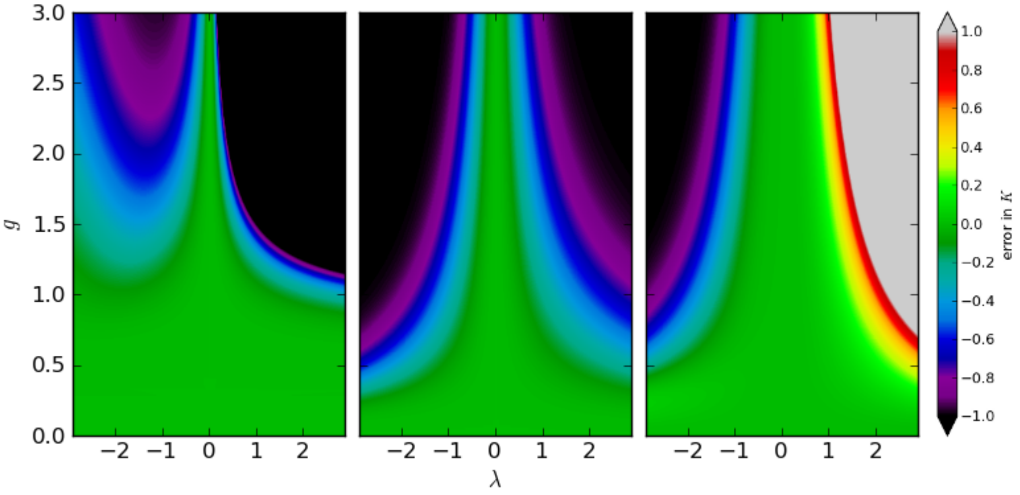
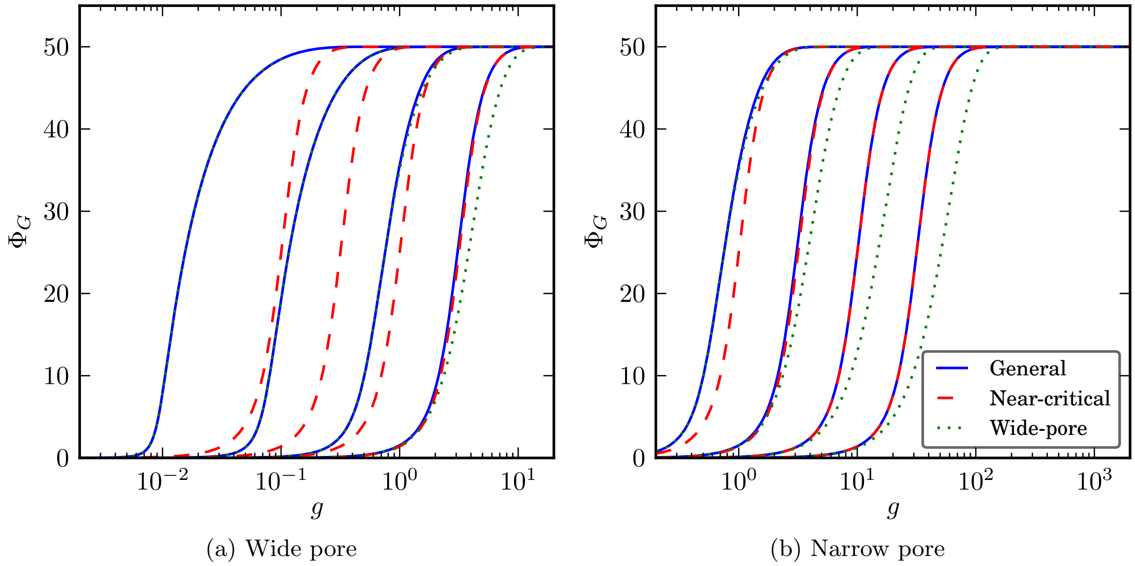


Figure 4.25: Calculated error in asymptotic expressions for K (4.53). (Left) Wide-pore regime (4.45), (center) near-critical regime (4.44), and (right) critical-narrow-pore (4.47).

To further highlight the differences between the three asymptotic equations with the general solution for K, the value of the error was calculated on a two dimensional grid. This value,

$$K_{\text{err}} = \frac{K_i - K_{\text{gen}}}{K_{\text{gen}}} \tag{4.48}$$

where i is one of the three asymptotic equations, near-critical (4.44), wide-pore (4.45), or critical-narrow-pore (4.47), and K_{gen} is Equation (4.53). Green represents quantitative agreement with the general solution, gray represents an over-prediction of K, and black an under-prediction. For gradient separations, we are most concerned with the agreement in the vicinity of the critical point, $\lambda = 0$. As expected, the wide pore model is the best performing asymptotic function where $g < 1$. However, in a narrow pore situation, it diverges rapidly away from the true value of K, as is seen from large gradient of K with increasing λ. The near-critical approximate functions are quantitatively accurate in the vicinity of the critical point, as well as having a smaller gradient in λ. We should expect the two latter expressions to perform better in a gradient chromatography model where narrow pores may be present.



(a) Wide pore

(b) Narrow pore

Figure 4.26: Solutions of the ODE using approximate, analytical expressions, and accurate numerical integration, for various values of $d\lambda/d\Phi$. From left to right, $d\lambda/d\Phi = 5\times10^{-2}, 5\times 10^{-3}, 5\times10^{-4}$, and 5×10^{-5}. Left, mean pore diameter is 1000, representing a wide-pore case. Right, mean pore diameter is 10, representing a narrow-pore case.

To evaluate the effect of these differences between the approximate and accurate expressions for K, the ODE describing gradient elution (4.39) was solved for using each expression. That is, the GE-CPA performance of the various expressions for K were tested against K_{gen}. These results are displayed in Figure 4.26. The y-axis is the composition of the elutant at time of gradient elution of a polymer of a given weight (the x-axis). The concentration can be converted to time of gradient elution with Equation (4.43). The curves represent the separation potential for a given solvent-change parameter ($d\lambda/d\Phi$), as a function of molecular mass. The horizontal plateau indicates the CPA, as all polymers elute at the same concentration, regardless of weight. The accuracy of the two asymptotic approaches appears to depend only on the pore size regime. The near-critical expression fails considerably in the wide-pore regime, over-predicting the weight of eluting polymers by an order of magnitude. However, at $g \geq 1$, its predictions agree quantitatively with the general solution. The wide-pore expression does not fail with such a large error, but appreciable error is found when $g \geq 1$.

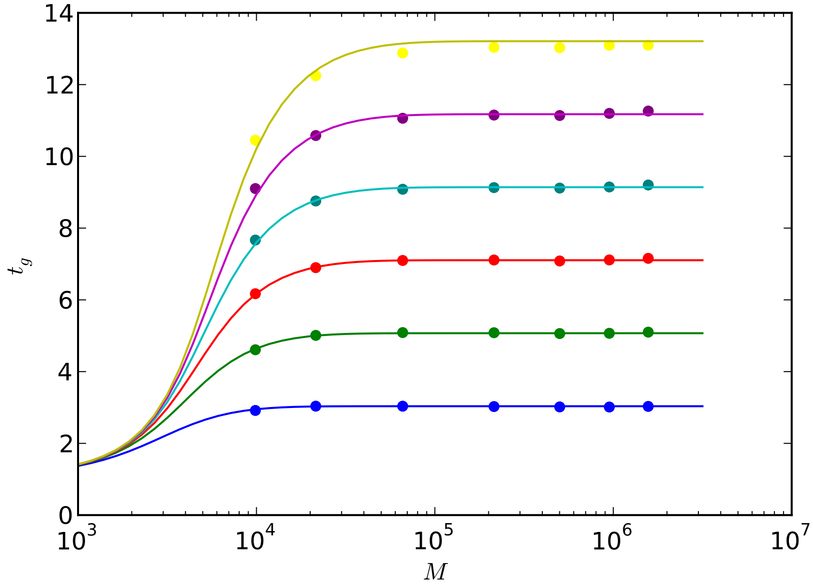


Figure 4.27: Comparison of general model to experimental data for various gradient rates. From top to bottom, $d\Phi/dt$ = 30, 25, 20, 15, 10, and 5 minutes (0–100% THF). Only one adjustable parameter ($d\lambda/d\Phi$) is used for the entire figure. See text for details experiment details.

The performance of the GE-CPA model was tested against experimental data. A range of polystrene (PS) standards from Polymer Laboratories was used in normal-phase chromatographic with Waters 150 × 3.9 mm, 4 μm particle size Nova-Pak® silica column with 60-Å media pore diameter. Various gradients of n-hexane–tetrachloroethane (THF) were examined. The results of the 10 minute (0–100% THF) gradient were used to calculate the only adjustable parameter in the model, $d\lambda/d\Phi$. This parameter was then used to predict elution at other gradient rates of 30, 25, 20, 15, and 5 minutes. The results are displayed in Figure 4.27. Good agreement is found for all gradients. This suggests that the GE-CPA model could be used to predict optimal separation by varying the gradient rate, or even shape, given a known polymer/solvent/column interaction.

4.3.3 Extention to Functionalized Polymers

To model a polymer with one or two terminal functional groups, we assumed the ideal chain/slit pore model with point interaction groups that differ in adsorption characteristic from the repeat monomer of the parent chain. This model was solved for exactly by Gorbunov and Vakhrushev [299],

$$K_{\text{mono}} = K_{\text{gen}} + q_a p_a \tag{4.49}$$

$$K_{\text{di}} = K_{\text{gen}} + q_a p_a + q_a p_a + q_a q_b p_{ab} \tag{4.50}$$

where mono and di subscripts indicate one or two terminal groups, respectfully, q_a and q_b are the reduced functional group interaction parameters, and p_a, p_b and p_{ab} are the one and two point contact probabilities of the functional group with the pore wall. The value q_i is the only new parameter introduced in this model (relative to the homopolymer model); if $q_i = 0$, the functional groups have no effect, and the solution for a homopolymer is recovered. If $q_i > 0$, the functional groups adsorb more strongly than the repeat monomers, and conversely if $q_i < 0$, the functional groups adsorb less strongly than the repeat units. K_{gen} represents the partition function of a chain without functional groups (e.g. the homopolymer case, Equation (4.53)).

The solutions for the contact probabilities are

$$p_a = p_b = \sum_{m=1(m\text{ odd})}^{\infty} \frac{\lambda A_m}{\alpha_m^2} \exp\left(-\alpha_m^2 g^2\right) \tag{4.51}$$

$$p_{ab} = \sum_{m=1(m\text{ odd})}^{\infty} A_m \exp\left(-\alpha_m^2 g^2\right) \tag{4.52}$$

$$\tag{4.53}$$

where

$$A_m = \frac{2\alpha_m^2}{\lambda(\lambda+1)+\alpha_m^2} \tag{4.54}$$

and α_m are the characteristic roots of Equation (4.41).

Changing solvent conditions during the gradient elution imply that λ will vary over the length of the column. The choice of the solvent model is a complex one; as the parameter λ captures the combined solvent/solute/substrate interaction and is thus a complex function. The simplest approach is to assume a linear dependence near the critical composition, as with the model for homopolymers in the previous section (4.3.3). The primary assumption for the extention to functionalized polymers is the the functional group's adsorption affinity does not vary with solvent quality, although the repeat monomers do. In other words, $\lambda = \lambda(\Phi)$, but $q_i \neq q_i(\Phi)$.

Although asymptotic solutions are available for Equations (4.49) and (4.50), even the simplest solutions to (4.39) yield ungainly results, if an analytical solution was possible at all. Thus, we chose to solve Equation (4.39) numerically to obtain the time of gradient elution, t_G. Numerical solution has other benefits as well:

1. Functions for K can be arbitrarily complex, as long as they are continuous in $(t, \tilde{x})$. This allows for more detailed models of K, including the possibility atomistic simulations.

2. More complex functions for the solvent/polymer/pore interaction can be used with current models of K.

3. The ambiguity of asymptotic conditions is avoided.

4. Additional complexities such as dispersion, diffusion, non-linear gradients, etc. can be relatively easily included.

Any standard ODE solver can be used; we selected the Python extension SciPy and its integration pack [306]. A mesh of 500 points on the $\tilde{x}$ coordinate was used. When integrating, the solutions for K_{gen}, p_a, p_b, and p_{ab} was solved to a precision of $\leq 10^{-4}$ for each integration step.

A primary intention of this work is to predict whether functionalized polymer can be separation by the number of identical group for molecules of similar weights. To test this, the parameters from the previous experimental work (Figure 4.25) were used for the repeat units, but one and then two functional groups added with interaction parameters $q_a = q_b = 1.0$. The results are shown in Figure 4.28. Time of gradient elution is plotted

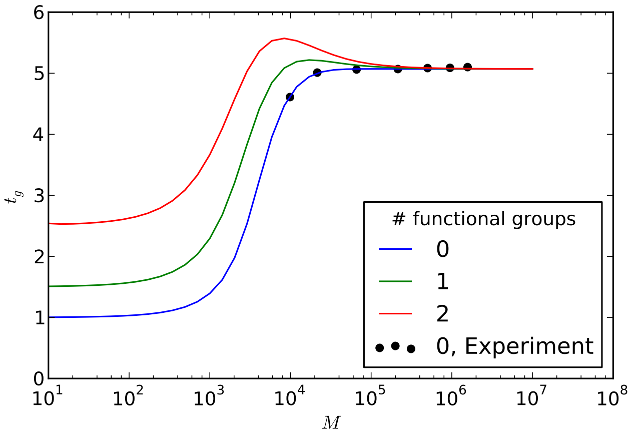


Figure 4.28: Comparison of functional group's affect on the time of gradient elution t_g, versus molecular weight of the solute, with $q_a = q_b = 1.0$. Adsorption functional groups delay elution of low molecular weight solutes. A maximum is found, were elution is later than the critical point. This suggests that the resolution of separation by number of identical functional groups could be increased in gradient mode, for a certain range of molecular weights.

as a function of molecular mass. The addition of the strongly adsorbing functional groups is predicted to increase the rentention time of short chains (MW $< 10^5$), with a longer rentention for two groups rather than one. Interestingly, a maximum is found at $\sim 5 \times 10^4$. In other words, the model predicts the elution of low-weight polymers *after* the critical point is reached and all high weight molecules are eluted. Current models for homopolymers predict that no material will elute after the CPA. The difference between the three lines at lower MW indicates that separation by number of functional groups is indeed possible with GE-CPA.

Figure 4.29 shows the results of the monofunctional expression for K (4.49), after integration in the GE-CPA model (4.39). The strength of the adsorptive functional group q_a is varied from 0 to 3. Zero replications previous homopolymer results (Figure 4.27. Increasing adsorptive strength leads to delayed elution of short chains. At CPA, the remaining polymers are eluted regardless of the functional group's strength. Before CPA is reached, a maximum develops and increases with adsorptive strength.

Having two strongly adsorbing functional makes these effects significantly more pronounced, as one might expect. Figure 4.30 and Figure 4.31 examines polymers with two terminal functional groups, as described by Equation (4.50). Even smaller values of q_i have

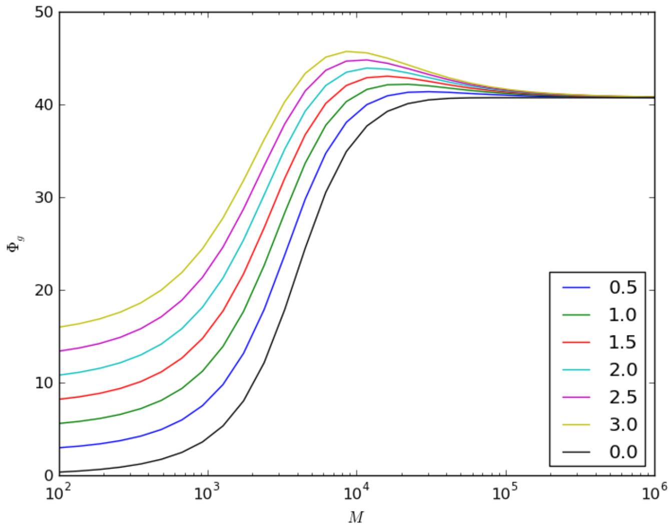


Figure 4.29: Monofunctional polymers in gradient elution, effect of parameter functional group strength

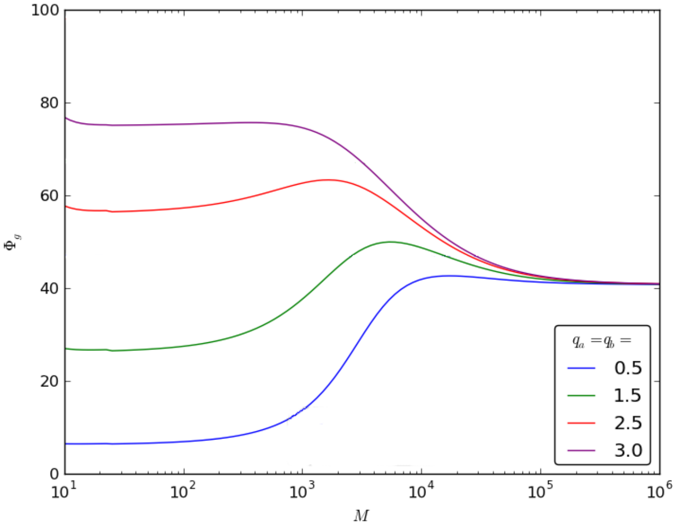


Figure 4.30: Difunctional polymers in gradient elution, effect of parameter functional group strength

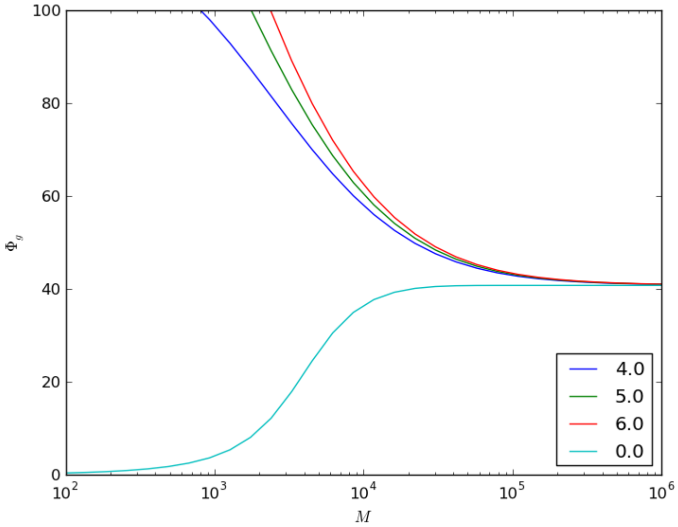


Figure 4.31: Difunctional polymers in gradient elution with strongly adsorbing functional groups. Large values of q force low MW polymers into an adsorbed state that cannot be overcome by increasing Φ

a strong effect (Figure 4.30); at $q_a = q_b = 3$, the model predicts a complete reversal of elution order, with no eluted polymers until the critical point, high weight mixture at CPA, followed by descending MW. Further increasing q_i predicts (Figure 4.31) that short chains will become strongly adsorbed, and will not elute from the column even at 100% favorable solvent.

To test the performance of the functional group model, and the primary assumption that q does not vary with solvent Φ, two chromatography experiments were performed with a polyethylene glycol (PEG) standard and mono-brominated PEG (PEG-Br) of the same weight, in isocratic and gradient modes. 10-μL of a 20,000 Da PEG standard dissolved at 1 mg/mL in H_2O was injected into a Symmetry C4 (4.6×150mm, 3.5-μm, 300Å pore) column, operating at a flow rate of 1 mL/min. The isocratic column was operated at 50% acetonitrile/H_2O, very close the critical point for PEG. A second experiment with the same operating procedure followed for PEG-Br. The isocratic chromatograms are shown in Figure 4.32. If we assume that the standard elutes at the CPA, the value for q_a is simply $\Delta t = 0.2$ (since K_{gen} and p_a in Equation (4.49) are equal to unity at the critical point).

Gradient elution of PEG and PEG-Br was predicted by integrating the balance equa-

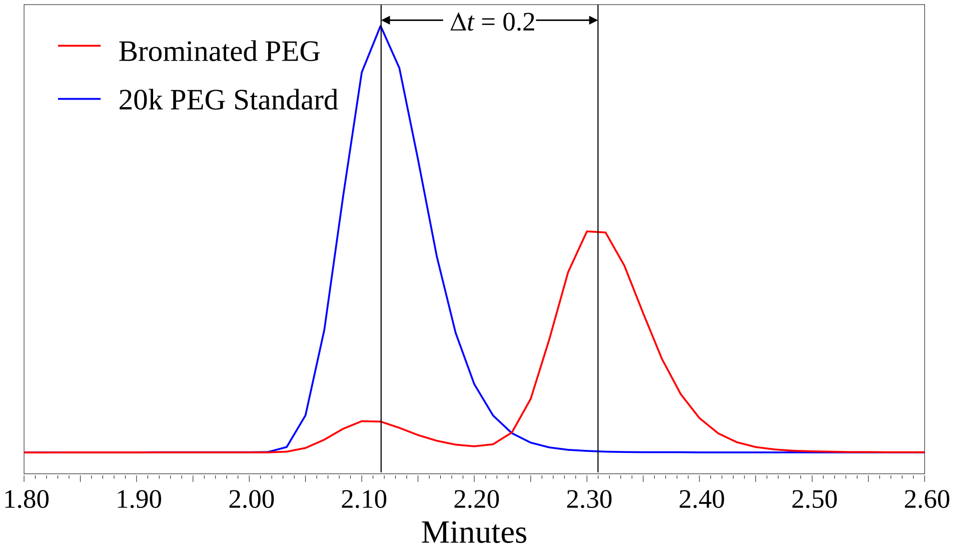


Figure 4.32: Experimental results for PEG and PEG-Br at LCCC

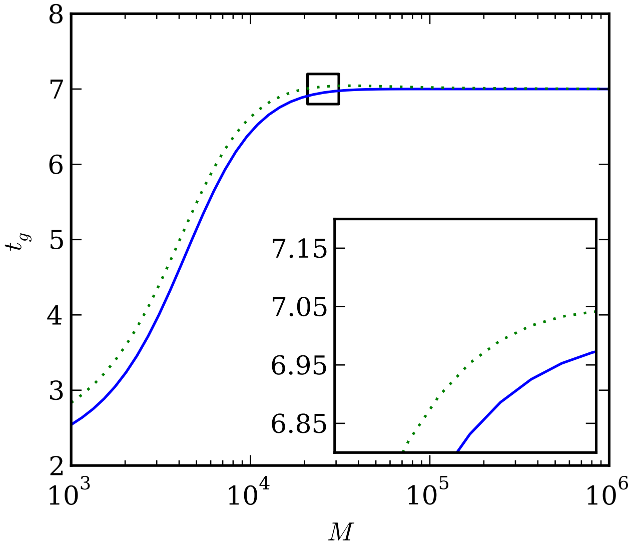


Figure 4.33: Gradient model prediction for PEG (blue line) and PEG-Br (green dotted line). Inset is zoomed for clarity.

tion (4.39) using the monofunctional expression for the partition coefficient (4.49). The results for PEG and PEG-Br are displayed in Figure 4.33. Since the bromine group interacts very weakly, the results are similar. However, before the critical point of elution, a slight maximum in elution time is observed for PEG-Br. The inset of Figure 4.33 shows this in detail. The gradient elution model predicts an elution delay for PEG-Br of about $\Delta\Phi = 1.0\%$, which for a 10 min gradient, corresponds to a $\sim$0.1 minute delay. This was tested by performing the chromatography runs again, but in gradient mode. Similarly prepared samples were injected into the same column, but with 100% H_2O mobile phase from $t = 0$–7 minutes, then a 10 minute linear gradient of acetonitrile (0–100%, from $t = 7$–17 minutes). The resulting chromatograms are shown in Figure 4.34. The brominated sample elutes later than the standard, by $\Delta t_g \approx 0.1$ minutes, exactly as the GE-CPA model had predicted. The interaction parameter $d\lambda/d\Phi$ was adjusted to 2×10^{-3} so that $M = 20,000$ Da was near the maxima in t_g. This is reasonable upon interpretation of the gradient-mode chromatogram; if $d\lambda/d\Phi$ is large, both PEG and PEG-Br are predicted to elute at same (critical-point) time, while a lower value predicts that PEG will elute well before the critical point, which is also not observed in experiment.

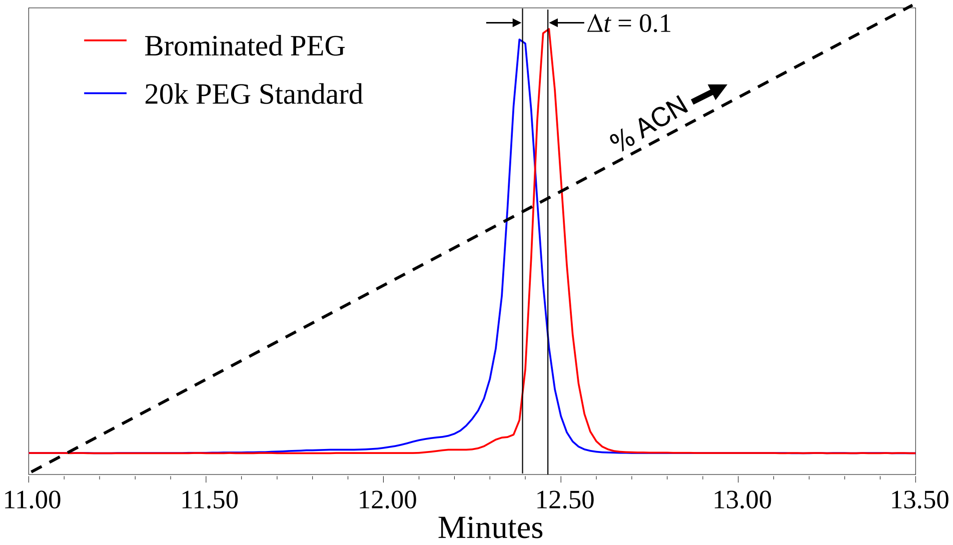


Figure 4.34: Experimental results for PEG and PEG-Br at GE-CPA. The delay in elution of PEG-Br was quantitatively predicted by the GE-CPA model. See text for experimental details.

4.3.4 Isocratic Results for Copolymers

The separation of copolymers is central to the emerging field of nanotechnology. Diblock copolymers have become a basic building block for templating chemistry, drug packaging and delivery, and many more. Separation and characterization of copolymers using chromatography is difficult. Often, components in copolymer mixtures have identical mass and chemical composition. Separation is needed as a function of heterogeneity, or the relative order of comonomers. This subtle difference limits the use of standard chromatographic approaches. This section explores the possible approaches to copolymer separation.

To model copolymer elution, an expression for K using the statical theory of interaction chromatography is found in the literature. The expression is a particular case of the general expression of a long chain with independent monomers, found by Ennis and Jönsson [307],

$$K = 2^m \sum_{i_1=1}^{\infty} \cdots \sum_{i_m=1}^{\infty} \exp\left(\sum_{j=1}^{m} -g_j^2 \alpha_{i_j,j}^2 \right) \times \\ \prod_{j=1}^{m} \frac{1}{\lambda_j^2 + \lambda_j + \alpha_{i_j,j}^2} \prod_{k=1}^{m-1} \frac{\lambda_k - \lambda_{k-1}}{\alpha_{i_k,k}^2 - \alpha_{i_{k+1},k+1}^2} \prod_{p=2}^{m-1} \alpha_{i_p,p}^2 \quad (4.55)$$

Here, m is total number of blocks of size g_i and with interaction parameter λ_i, and $\alpha_{i,j}$ are the corresponding eigenvalues of the characteristic equation (4.41). In theory, this enables us to test any copolymer. Unfortunately, this equation scales as the following: $O(m)$ for the sums of the first eigenvalue, $O(2^m)$ for the sums of the second, $O(3^m)$ for the third, and so on. Luckily, in most cases, the first eigenvalue is the only significant contribution. The accuracy of Equation (4.55) with limited number of eigenvalues was tested by calculating the error for converged (err $\leq 10^{-2}$), and with using only the first eigenvalue. The results are shown in Figure 4.35. Significant error was found only in the wide pore case. When $g \geq 1$, there is very little error introduced. Thus, the wide pore case requires the summation of the second eigenvalue. Inclusion of the second eigenvalue reduced error to nearly zero. Only very blocky (AB repeating), long chains in the wide-pore regime are unfeasible for calculation by this method.

To understand the dependence of K on blockiness, an algorithm was developed to gen-

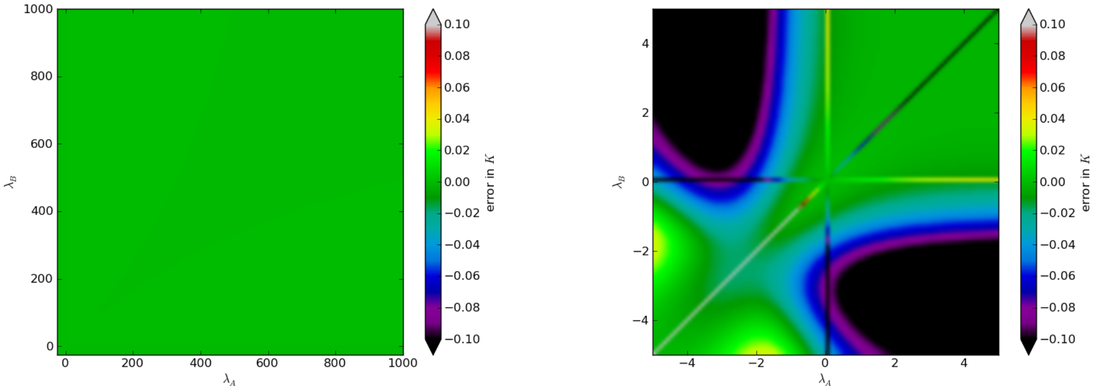


Figure 4.35: Error in calculation of K by using only summations of first eigenvalue, for (left) $g = 1$, and (right) $g = 0.5$. Note that black represents at least 10% error. The error is reduced to approximately zero if the summations are completed for the second eigenvalue.

erate a random chain using given probability distributions, based on a Markov method described in Ref. [298]. Assuming a constant total ratio of monomers ($P_\text{A} = P_\text{B} = 50\%$), the probability of a certain type of monomer following another given type is assigned to v_{ij} (e.g. v_AB is the probability of a B monomer following an A monomer); this is the transition matrix describing the chain. For a chain comprised of 2 species, only 2 of the 4 matrix entries are independent, and the blockiness of the chain can be characterized as a function of the remaining probabilities, $v = v_\text{AB} + v_\text{BA}$, the so called sequence heterogeneity index. A value of $v = 0$ indicates a diblock copolymer, $v = 1$ a statistically random distribution of monomers (so-called Bernoulli statistics [REF]), and $v = 2$ an alternating-type copolymer (alternating monomers of A and B).

The following figures were calculated using a chain of 200 monomers (so blocks m is always less than 200). This may not be sufficient to exceed the chemical correlation length, but at least gives a first order approximation of retention time versus blockiness v. Additionally, only one randomly-generated chain was sampled for a given point. Statistical variability could be reduced by averaging over several generated chains for the point.

The above results predict that isocratic separation by blockiness is possible, if block B is chromatographically invisible and a solvent/column is chosen to be in the adsorption mode. In the narrow pore regime, retention becomes independent of blockiness, but not adsorption

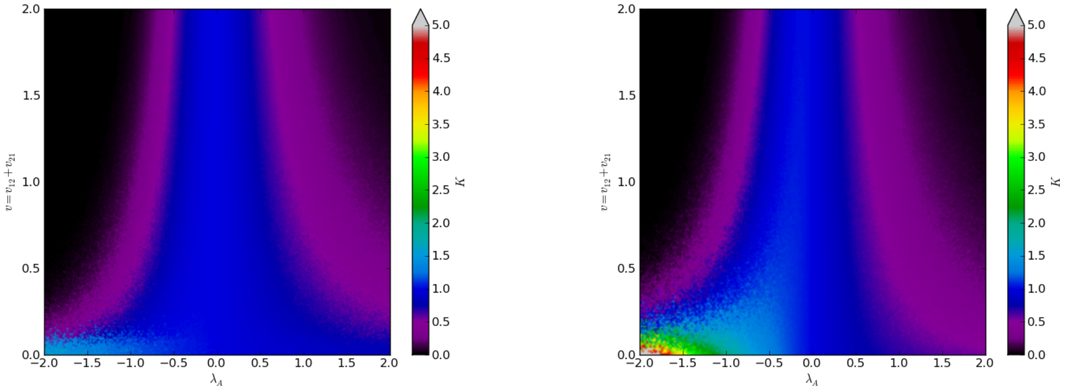


Figure 4.36: Dependence of chemical heterogeneity on λ_{A}, with $\lambda_{\mathrm{B}} = 0$, for (left) $g = 0.5$ and (right) $g = 1$

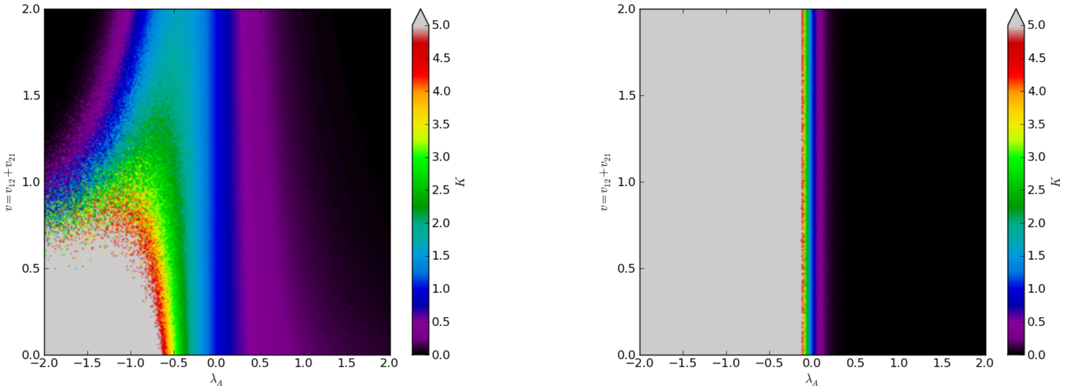


Figure 4.37: Dependence of chemical heterogeneity on λ_{A}, with $\lambda_{\mathrm{B}} = 0$, for (left) $g = 2$ and (right) $g = 5$

interaction parameter. In fact, a steep gradient near the critical point of adsorption suggests gradient elution would have good separation results, while isocratic separation would not.

4.3.5 Conclusions

A model for chromatography mode gradient-elution at critical conditions was developed. The accuracy of competing models for the partition coefficient K was tested for homopolymers by comparison to the general solution. We found the simplest expression, the "near-critical" asymptotic expression of Brun [14, 15] (which allows for a closed-form analytical expression of retention time in GE-CPA) is accurate for cases where the polymer's radius

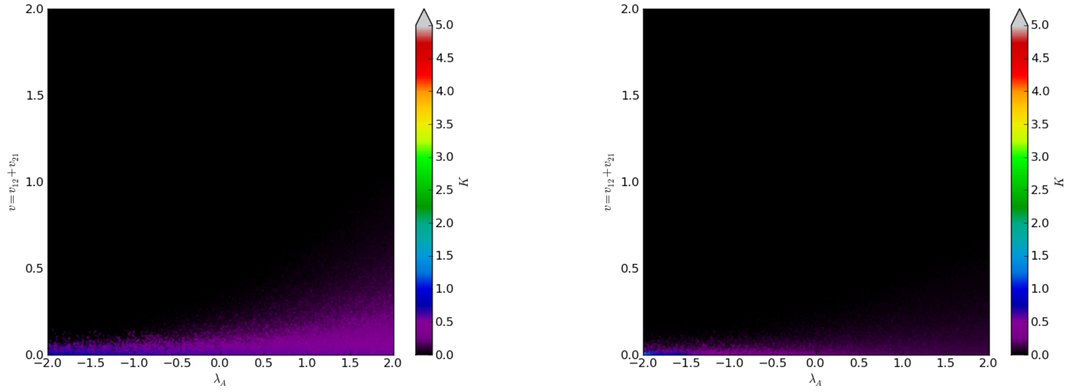


Figure 4.38: Dependence of chemical heterogeneity on λ_{A}, with $\lambda_{\mathrm{B}} = 1000$, for (left) $g = 0.5$ and (right) $g = 1$

of gyration is close to or larger than the mean pore size. The GE-CPA model has one adjustable interaction parameter, and was used to quantitatively predict elution times for various gradient rates.

The GE-CPA model was extended to functionalized polymers and copolymers, using appropriate models for the partition coefficient from the literature. Functionalized polymers are predicted to elute after the CPA for strongly-adsorbing functional groups. Experimental comparison showed that the model produces quantitative retention times, once the functional group parameter is fit from isocratic experiment. Application of the copolymer model showed that separation by blockiness is possible, for a certain size range of polymers.

Chapter 5

Conclusions

Conclusions

This dissertation has covered many areas where improved knowledge of the thermodynamics of fluids at the nanoscale are useful. This was achieved mainly through the tools of molecular simulation. Through the strategic application of grand canonical, gauge cell, and incremental gauge cell Monte Carlo, a clear picture of the systems under consideration was developed.

The first part of the work examined the role of cavitation in the desorption of fluids in nanoporous media. Experimental collaborators used detailed, highly-resolved adsorption measurements to determine the pressure where evaporation occurs. Tabulated by primary pore size, a dependence of the cavitation pressure on the pore size is found. A weak dependence for pore diameter $d < 10$ nm is followed by a flat plateau for $10 < d < 35$ nm, suggesting an extent to adsorption influences on cavitation. This is confirmed by MC simulation of similarly sized model systems, where the nucleation barrier is found to be approximately constant for $d > 10$ nm. This implies homogeneous nucleation in bulk-like conditions. A follow-up study used grand canonical and gauge cell MC to investigate the role of a pore wall defect on the nucleation mechanism. A theoretical model based on classical nucleation theory was developed, and predicted a transition from homogeneous to heterogeneous nucleation for a certain defect size. This was qualitatively confirmed with detailed MC studies that tracked the formation of the critical nucleus that precedes cavitation. Finally, the influence of pore structure was studied by overlapping two spherical pores. Depending on the degree of overlap, desorption occurred via cavitation (for a small overlap that restricts mass transfer between the pores), or via a receding meniscus (for a large overlap). Scanning isotherms were calculated and suggest a viable method to study porous material with similar morphology via experiment.

The next chapter of this dissertation focuses on adsorption of polymer chains. A new method to calculate the chemical potential of polymers in MC, the incremental gauge cell, was developed and implemented. It was shown to be approximately one order of magnitude faster than the previous method to reach the same statistical certainty in chemical potential. The method was applied to long, single LJ chains in confinement and adsorption environments. Adsorbing chains at low temperature were found to exhibit characteristics similar to confined simple fluids—monolayer formation and a capillary condensation transition to a volume-filling conformation. The dynamics of polymer translocation was studied by using the incremental gauge cell to calculate the free energy landscape. This was applied as the driving force in the Fokker-Planck equation to calculate dynamics. A non-monotonic dependence of translocation times on adsorption potential was found; strong potentials yield fast and probable translocation, while in systems exhibiting a free energy minima, the chain reaches a partially-adsorbed metastable state and translocation slows considerably. For the systems with an 'uphill' free energy, translocation was not likely, but proceeded quickly if at all.

Finally, a model for gradient-elution critical point of adsorption chromatography was developed and tested against experiment. Suitable models for the partition coefficient were found and applied to complex polymers, such as homopolymers with functional groups, and copolymers ranging from diblocks to statistical copolymers. The application of this model has great practical value, as gradient elution is a key tool in the separation and characterization of such complex polymers, and there exists very little theoretical interpretation of experimental results.

Continued and Future Work

There is great potential in continuation of the some of the work initiated in this dissertation. We revealed that homogeneous cavitation can be experimentally observed at a reasonable temperature and pressure, with no special experimental setup. Besides the obvious use in characterizing porous materials, this work offers an approach to study the nature of homogeneous cavitation itself. Homogeneous cavitation in the bulk is notoriously difficult to study, as it requires supercooling metastable fluids. Observation of cavitation in mesopores may lead to new a approach to study homogeneous nucleation.

Our work on heterogeneous cavitation showed that a large, non-wetting defect can significantly increase the observed cavitation pressure. However, fundamental questions regarding the nature of fluid behavior in porous silicon still exists, since defects of that nature are not possible for liquid nitrogen on porous silicon. Continued work in this area offers a path toward better understanding of nanoscale physics.

The incremental gauge cell method has been shown to be a powerful tool in the simulation of polymers. The problems studied in this work, namely the critical point of adsorption and translocation, can benefit greatly from new, detailed models. The critical point of adsorption has a great practical application in the separation of complex polymer molecules; however design and interpretation of experiments at the critical point are difficult and costly due to the lack of an adequate model. Detailed simulations using the incremental gauge cell can help eliminate many unnecessary experiments. In addition, theoretical study of the critical point itself may challenge some long held assumptions in the chromatography field. Similarly, study of translocation has been hampered by numerous conflicting models. Accurate simulation of specific cases may help to answer questions regarding fundamental translocation physics.

The model developed for gradient elution chromatography has a great potential for further development. Many interesting polymer morphologies can be easily added, including stars, theta-shaped loops, and functionalized stars. Additional physics can be included, such as diffusion along the column, better solvent models, and excluded volume. Predictive algorithms can change variables such as gradient rate and shape to give practitioners operating parameters for optimal separation of a given sample. Finally, a detailed investigation of isocratic and gradient separation by blockiness can shed light on new methods for purification of copolymers by minute structural details.

Appendix

A Derivations of incremental chemical potential

Calculation of the incremental chemical potential (ICP) in Monte Carlo simulation was first suggested by Kumar et al. [11] as a possible remedy to inefficient MC methods of the day (such as Rosenbluth insertions). Physically, a system of chains is equilibrated and and a test "ghost" monomer is inserted onto the end of chain. This procedure is reminencent of the particle insertion method proposed by Widom [8]. To find the chemical potential of an n-mer macromolecule, n individual simulations must be performed, although evidence suggests that for chains in good solvent conditions, the incremental chemical potential has little or no dependence on chain length [80, 81, 115]. While Kumar's "modified Widom insertion" is an improvement over primitive Rosenbluth insertions, the same drawbacks that plague standard Widom insertions will affect this chain increment method. Specifically, dense and/or inhomogeneous systems are not able to be sampled effectively. The gauge cell method [9, 10, 49] provides an alternate means to calculating the chemical potential in canonical systems. This is implemented by placing a gauge cell in contact with the system cell. The gauge cell contains monomers which are allowed to be added or removed from a target chain in the system cell. Thermodynamically, this system will minimize Helmholtz free energy, with the implication that the system and gauge cell will have comparable chemical potentials. The rest of this section will introduce the basic statistical mechanical equations as well as notation used through out the document.

A.1 Modified Widom approach

To derive the expressions for general polymeric MC systems, we begin with the canonical (constant NVT) partition function,

$$Q(\{N,n\},V,T) = \frac{1}{\Lambda^{3Nn}N!}\int\limits_V d\mathbf{r}^{Nn}\exp\left[-\beta\Phi(\mathbf{r}^{Nn})\right], \tag{A.1}$$

where N is the number of indistinguishable macromolecules (polymer chains), each n monomers in length, V and T are the system's volume and temperature respectfully, $\beta = 1/k_\mathrm{B}T$ is the inverse temperature, Λ is the thermal de Broglie wavelength of the monomers (assumed identical to each other), Φ is the sum of the various potential energy contributions (such as intermolecular, intramolecular and external [attractive pore walls]), and $\mathbf{r}$ is the set of position vectors for the system in phase space. This equation fully describes the system and makes no assumptions regarding the interconnectivity of the chain. The so called "bridge" equation links the microscopic partition function to the macroscopic quantity Helmholtz free energy,

$$F(\{N,n\},V,T) = -k_\mathrm{B}T \ln Q(\{N,n\},V,T). \tag{A.2}$$

From the fundamental equation of thermodynamics, the chemical potential of a system is

$$\mu(N,n) = \left(\frac{\partial F}{\partial N}\right)_{V,T} \tag{A.3}$$

at constant V, T and in the thermodynamic limit $N \to \infty$. For large N, this value can be approximated by a finite difference:

$$\mu(N,n) = F(\{N+1,n\},V,T) - F(\{N,n\},V,T). \tag{A.4}$$

Note that either a forward or reverse difference is technically correct, however the choice has serious practical consequences. In the derivations below, certain choices of forward or reverse differences for chemical potential or incremental chemical potential may result in non-cancelling terms that complicate practical calculations of these quantities.

Approach of Kumar et al.

The following approach calculation of ICP was proposed by Kumar et al. in [11]. First, the chemical potential (Eq. A.4) is considered as a reverse finite difference rather than forward,

$$\mu(N,n) = F(\{N,n\},V,T) - F(\{N-1,n\},V,T). \tag{A.5}$$

Now consider the system described by Eq. A.1, that is N chains of n monomers. If one of the chains in this system is incremented to $n+1$ monomers, the chemical potential of said chain would be:

$$\mu(N-1,n;1,n+1) = F(\{N-1,n;1,n+1\},V,T) - F(\{N-1,n\},V,T). \tag{A.6}$$

We can now define the incremental chemical potential as the difference of chemical potentials between identical chains with a single monomer difference in length, and can be found by subtracting Eq. A.5 from Eq. A.6,

$$\mu_{incr} \equiv \mu(N-1,n;1,n+1) - \mu(N,n) = F(\{N-1,n;1,n+1\},V,T) - F(\{N,n\},V,T). \tag{A.7}$$

With this definition in hand, we can relate the Widom particle insertion method [8] to incremental chemical potentials. Realizing that

$$Z(\{N-1,n;1,n+1\},V,T) = \int_V \cdots \int_V d\mathbf{r}_1 \cdots d\mathbf{r}_{Nn} d\mathbf{r}_{n+1} \exp\left(-\beta\left[U(\mathbf{r}_1,\cdots,\mathbf{r}_{Nn}) + U(\mathbf{r}_{n+1})\right]\right), \tag{A.8}$$

where Z is the configurational integral of the canonical partition function,

$$Z(\{N,n\},V,T) = \Lambda^{3Nn} N!\; Q(\{N,n\},V,T). \tag{A.9}$$

In Eq. A.8, $U(\mathbf{r}_{n+1})$ is the interaction energy experienced by the $(n+1)$th bead inserted onto the end of a chain in the system. Combining Eqs. A.7 with A.1 and A.8, we obtain

$$\exp(-\beta\mu_{incr}^{ex}) = \frac{\int_V \cdots \int_V d\mathbf{r}_1 \cdots d\mathbf{r}_{Nn} \exp\left[-\beta U(\mathbf{r}_1,\cdots,\mathbf{r}_{Nn})\right] \int_V d\mathbf{r}_{n+1} \exp\left[-\beta U(\mathbf{r}_{n+1})\right]}{Z(\{N,n\},V,T)}. \tag{A.10}$$

The ICP considered is an excess value, as we are only considering the configuration part of the partition function. Equation A.10 contains a statistical average of the Boltzman factor for the $n+1$ monomer, and simplifies to

$$-\beta\mu_{incr}^{ex} = \ln\left\langle \exp\left[-\beta U(\mathbf{r}_{n+1})\right]\right\rangle_{\{N,n\},V,T}. \tag{A.11}$$

This is the main equation the work of Ref. [11] is based on. Chemical potential of the entire chain is simply the sum of its incremental values. The incremental chemical potential is a well-defined thermodynamic quantity, with no restricting assumptions used in its derivation. This can be illustrated [232] by expanding terms in the definition of the chemical potential of a chain, Eq. A.5, [232]

$$\exp\left[-\beta\mu(N,n)\right] = \frac{Z(N,n)}{Z(N-1,n;1,n-1)}\frac{Z(N-1,n;1,n-1)}{Z(N-1,n;1,n-2)}\cdots\frac{Z(N-1,n,1,1)}{Z(N-1,n)}. \quad (A.12)$$

Applying the definition of ICP (Eq. A.7) to this expansion, we see that

$$\mu(n) = \mu_{incr}(n-1) + \mu_{incr}(n-2) + \ldots + \mu_{incr}(0) \quad (A.13)$$

where n is the length of a varying chain in an equilibriated system of $N-1$ chains with n monomers each, and $\mu_{incr}(0)$ is the chemical potential of inserting a single monomer into the entire system volume.

Alternative Approach

Another approach to calculation of the incremental gauge cell is to utilize the forward finite-difference definition of chemical potential, Eq. A.4, by calculating the chemical potential of an additional n-mer. We begin with the same system of N chains of n monomers. We insert a designed "test" chain of n_t monomers. This configuration is described by the partition function $Q(N,n;1,n_t,V,T)$. The chemical potential of the addition n_t chain we included is

$$\mu(n_t) = F(\{N,n;1,n_t\},V,T) - F(\{N,n\},V,T). \quad (A.14)$$

Similarly, the chemical potential of an incremented test chain, n_t+1, is

$$\mu(n_t+1) = F(\{N,n;1,n_t+1\},V,T) - F(\{N,n\},V,T). \quad (A.15)$$

We now define the incremental chemical potential as the difference in chemical potential of two chains, one of which with an additional monomer,

$$\mu_{incr}(n_t) \equiv \mu(n_t+1) - \mu(n_t) = F(\{N,n;1,n_t+1\},V,T) - F(\{N,n;1,n_t\},V,T). \quad \text{(A.16)}$$

This is equilivent in principle to Eq. A.7. Using the definition of the Helmholtz free energy (Eq. A.2), we obtain a ratio of partition functions,

$$\begin{aligned}
\mu_{incr}(n_t) &= -k_{\mathrm{B}}T \ln \left[\frac{Q(\{N,n;1,n_t+1\},V,T)}{Q(\{N,n;1,n_t\},V,T)} \right] && \text{(A.17)}\\
&= -k_{\mathrm{B}}T \ln \left[\frac{\frac{1}{\Lambda^{3Nn+3(n_t+1)}(N+1)} \int\limits_V d\mathbf{r}^{Nn+n_t+1} \exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t+1})\right]}{\frac{1}{\Lambda^{3Nn+3n_t}(N+1)} \int\limits_V d\mathbf{r}^{Nn+n_t} \exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t})\right]} \right] && \text{(A.18)}\\
&= -k_{\mathrm{B}}T \ln \left[\frac{1}{\Lambda^3} \frac{\int\limits_V d\mathbf{r}^{Nn+n_t} \exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t})\right] \int\limits_V d\mathbf{r}_{n_t+1} \exp\left[-\beta\phi(\mathbf{r}_{n_t+1})\right]}{\int\limits_V d\mathbf{r}^{Nn+n_t} \exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t})\right]} \right], && \text{(A.19)}
\end{aligned}$$

where $\phi(\mathbf{r}_{n_t+1})$ is the interaction energy of the n_t+1 inserted monomer at the end of the chain. Note Eq. A.17 is only valid if the number of molecule is the same (i.e. $n_t > 0$). Equation (A.19) is a thermal average of the partition function of the inserted monomer, similar to the traditional Widom's method. Thus, the average of the n_t+1 integral is computed over configuration of $\{N,n;1,n_t\}$ particles and volume V,

$$\begin{aligned}
\mu_{incr}(n_t) &= -k_{\mathrm{B}}T \ln \left[\frac{1}{\Lambda^3}\right] - k_{\mathrm{B}}T \ln \left[\int\limits_V d\mathbf{r}_{n_t+1} \left\langle \exp -\beta\phi(\mathbf{r}_{n_t+1}) \right\rangle_{Nn+n_t} \right] && \text{(A.20)}\\
&= -k_{\mathrm{B}}T \ln \left[\frac{1}{\Lambda^3}\right] - k_{\mathrm{B}}T \ln \left[V_{ins} \left\langle \exp -\beta\phi(\mathbf{r}_{n_t+1}) \right\rangle_{Nn+n_t,V_{ins}} \right] && \text{(A.21)}
\end{aligned}$$

where V_{ins} is a small volume at the end of the test chain where further insertions are allowed. To arrive at a full expression for the chemical potential of the system, we must expand the ratio presented in Eq. (A.4)

$$-k_{\mathrm{B}}T \ln \left[\frac{Q(\{N,n;1,n_t\},V,T)}{Q(\{N,n;1,n_t-1\},V,T)} \frac{Q(\{N,n;1,n_t-1\},V,T)}{Q(\{N,n;1,n_t-2\},V,T)} \cdots \frac{Q(\{N,n;1,1\},V,T)}{Q(\{N,n\},V,T)} \right], \quad \text{(A.22)}$$

which becomes the expression for chemical potential if we set n_t to n. Applying the definition of incremental chemical potential above,

$$\mu(N,n) = \mu_{incr}(n-1) + \mu_{incr}(n-2) + \cdots + \mu_{monomer}, \tag{A.23}$$

where $\mu_{monomer}$ is the chemical potential of inserting a monomer into the system of chains. It can be shown that this is simply Widom's insertion method on the system of N, n chains. Applying this, and the derivation of μ_{incr} above, Equation (A.23) becomes

$$\begin{aligned}-\frac{\mu(N,n)}{k_\mathrm{B}T} = \sum_{n_t=1}^{n-1} \ln\left[\frac{V_{ins}}{\Lambda^3}\left\langle \exp\left(-\frac{\phi(\mathbf{r}_{n_t+1})}{k_\mathrm{B}T}\right)\right\rangle_{Nn+n_t,V_{ins}}\right] + \\ \ln\left[\frac{V}{\Lambda^3(N+1)}\left\langle \exp\left(-\frac{\phi(\mathbf{r}_1)}{k_\mathrm{B}T}\right)\right\rangle_{Nn,V}\right],\end{aligned} \tag{A.24}$$

where $\mathbf{r}_{n_t+1}$ and $\mathbf{r}_1$ are the position vectors for trial particles inserted into the biased insertion volume or the system volume, respectfully. Seperating and rearranging,

$$\begin{aligned}-\frac{\mu(N,n)}{k_\mathrm{B}T} = \ln\left[\frac{V}{\Lambda^{3n}(N+1)}\right] + \ln\left[\left\langle \exp\left(-\frac{\phi(\mathbf{r}_1)}{k_\mathrm{B}T}\right)\right\rangle_{Nn,V}\right] + \\ \sum_{n_t=1}^{n-1} \ln\left[V_{ins}\left\langle \exp\left(-\frac{\phi(\mathbf{r}_{n_t+1})}{k_\mathrm{B}T}\right)\right\rangle_{Nn+n_t,V_{ins}}\right].\end{aligned} \tag{A.25}$$

We've defined the reference state as an ideal gas of non-selfinteracting n-mers. Alternatively, we can include internal interactions and only report excess chemical potentials with external (non-bonded) interactions (see Section A.4 for derivation). The final expression for the chemical potential of a fluid composed of N n-mers is:

$$\mu(N,n) = \mu_{igc} + \mu^{ex}_{monomer} + \sum_{n_t=1}^{n-1} \mu^{ex}_{incr}(n_t) \tag{A.26}$$

Note the first and second terms can usually be calculated analytically. For chains well above their θ point, the μ_{incr} is essentially constant with respect to n_t, thus signifcantly reducing computation needs.

Polydisperse fluid

A more general approach is to consider a distribution of chain lengths rather than one fixed length. Let N_i be the number of chains of length i in volume V, in a mixture composed of $\{N_\alpha\}$, where the $\{\dots\}$ brackets represent the set of chain lengths, $\{N_\alpha\} = N_1, N_2, N_3, \cdots, N_\infty$. That is, N_1 is number of monomers in the mixture, N_2 is the number of dimers, and so on. Let the subscript α represent *all* lengths of chains in the mixture. The total number of molecules is

$$M = \sum_{i=1}^{\infty} N_i. \tag{A.27}$$

Similarly, the total number of monomers is

$$m = \sum_{i=1}^{\infty} iN_i. \tag{A.28}$$

The partition function for the mixture is then

$$Q(\{N_\alpha\}, V, T) = \frac{1}{\Lambda^{3m}\prod_{i=1}^{\infty}(N_i!)} \int\limits_V d\mathbf{r}^m \exp\left[-\frac{\Phi(\mathbf{r}^m)}{k_\mathrm{B}T}\right]. \tag{A.29}$$

If an additional chain of length ν monomers were inserted into this system, the partition function would then be

$$Q(\{N_1, N_2, \cdots, N_\nu{+}1, \cdots\}, V, T) = \frac{1}{\Lambda^{3(m+\nu)}(N_\nu+1)\prod_{i=1}^{\infty}(N_\alpha!)} \int\limits_V d\mathbf{r}^{m+\nu} \exp\left[-\frac{\Phi(\mathbf{r}^{m+\nu})}{k_\mathrm{B}T}\right], \tag{A.30}$$

And thus the chemical potential of the ν component of the mixture would be the difference of free energy upon inserting the $N_\nu + 1$ chain,

$$\mu_\nu(\{N_\alpha\}) = \left(\frac{\partial F}{\partial N_\nu}\right)_{V,T,\{N_{\alpha\neq\nu}\}}, \tag{A.31}$$

and its finite difference,

$$\mu_\nu(\{N_\alpha\}) \approx F(\{N_1, \cdots, N_\nu+1, \cdots\}, V, T) - F(\{N_1, \cdots, N_\nu, \cdots\}, V, T). \tag{A.32}$$

Substituting in the bridge equation (Eq. A.2) for the Helmholtz free energy, we obtain a ratio of partition functions,

$$\exp\left[-\frac{\mu_\nu(\{N_1,\cdots,N_\nu,\cdots\})}{k_\mathrm{B}T}\right]=\frac{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu,\cdots\},V,T)}= \\ \frac{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_{\nu-1}+1,\cdots\},V,T)}\frac{Q(\{N_1,\cdots,N_{\nu-1}+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_{\nu-2}+1,\cdots\},V,T)}\cdots \\ \frac{Q(\{N_1+1,N_2,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu,\cdots\},V,T)}. \tag{A.33}$$

Incremental chemical potential can now be defined as the contribution of each ratio between systems with test chains of ν and $\nu-1$ beads,

$$\mu_\nu(\{N_\alpha\})=\mu_{incr}(\nu-1,\{N_\alpha\})+\mu_{incr}(\nu-2,\{N_\alpha\})+\cdots+\mu_0(\{N_\alpha\}) \tag{A.34}$$

$$=\sum_{i=1}^{\nu-1}\mu_{incr}(i,\{N_\alpha\})+\mu_0(\{N_\alpha\}), \tag{A.35}$$

where μ_0 is equivalent to the chemical potential of a monomer inserted into the mixture of $\{N_\alpha\}$ chains, and μ_{incr} is defined as

$$\mu_{incr}(\nu,\{N_\alpha\})=-k_\mathrm{B}T\ln\left[\frac{Q(\{N_1,\cdots,N_\nu,N_{\nu+1}+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu+1,N_{\nu+1},\cdots\},V,T)}\right]\quad(\nu>0) \tag{A.36}$$

Note that this definition is the same as if we define $\mu_{incr}=\mu_{\nu+1}-\mu_\nu$.

We will treat the chemical potential of the monomer first, following the standard approach for calculation of chemical potentials via Widom insertions,

$$\mu_0(\{N_\alpha\})=-k_\mathrm{B}T\ln\left[\frac{Q(\{N_1,\cdots,N_1+1,\cdots\},V,T)}{Q(\{N_\alpha\},V,T)}\right] \tag{A.37a}$$

$$=-k_\mathrm{B}T\ln\left[\frac{\Lambda^{3m}\prod_{i=1}^{\infty}(N_i!)\int_V d\mathbf{r}^m d\mathbf{r}_0\exp\left[-\Phi(\mathbf{r}^m)/k_\mathrm{B}T\right]\exp\left[-\phi(\mathbf{r}_0)/k_\mathrm{B}T\right]}{\Lambda^{3(m+1)}(N_1+1)\prod_{i=1}^{\infty}(N_i!)\int_V d\mathbf{r}^m\exp\left[-\Phi(\mathbf{r}^m)/k_\mathrm{B}T\right]}\right] \tag{A.37b}$$

$$=-k_\mathrm{B}T\ln\left[\frac{1}{\Lambda^3(N_1+1)}\int_V d\mathbf{r}_0\left\langle\exp\left[-\phi(\mathbf{r}_0)/k_\mathrm{B}T\right]\right\rangle_{\{N_\alpha\}}\right], \tag{A.37c}$$

where $\mathbf{r}^m$ is a set of position vectors for all m particles, $\mathbf{r}_0$ is the position vector for a particle inserted at random into the system volume V, ϕ is the interaction potential energy the particle at $\mathbf{r}_0$ experiences, and $\langle\cdots\rangle$ denotes the canonical average over a configuration of $\{N_\alpha\}$ chains of all lengths α. Computing this average over the volume V gives

$$\mu_0(\{N_\alpha\}) = -k_{\mathrm{B}}T \ln\left[\frac{V}{\Lambda^3(N_1+1)}\right] - k_{\mathrm{B}}T \ln\left[\langle \exp\left[-\phi(\mathbf{r}_0)/k_{\mathrm{B}}T\right]\rangle_{\{N_\alpha\},V}\right] \tag{A.38}$$

$$= \mu_0^{id} + \mu_0^{ex}. \tag{A.39}$$

This is equivalent to the standard Widom insertion equation.

The incremental chemical potential for each successive particle inserted is defined in Eq. (A.36). To calculate a practical equation, we will follow the same approach as above.

$$\exp\left[-\frac{\mu_{incr}(\nu,\{N_\alpha\})}{k_{\mathrm{B}}T}\right] = \tag{A.40a}$$

$$\frac{\Lambda^{3(m+\nu)}(N_\nu+1)\prod_{i=1}^{\infty}(N_i!)\int_V d\mathbf{r}^{m+\nu}d\mathbf{r}_{incr}\exp\left[-\Phi(\mathbf{r}^{m+\nu})/k_{\mathrm{B}}T\right]\exp\left[-\phi(\mathbf{r}_{incr})/k_{\mathrm{B}}T\right]}{\Lambda^{3(m+\nu+1)}(N_{\nu+1}+1)\prod_{i=1}^{\infty}(N_i!)\int_V d\mathbf{r}^{m+\nu}\exp\left[-\Phi(\mathbf{r}^{m+\nu})/k_{\mathrm{B}}T\right]} \tag{A.40b}$$

$$= \frac{V(N_\nu+1)}{\Lambda^3(N_{\nu+1}+1)}\frac{1}{V}\int_V d\mathbf{r}_{incr}\,\langle\exp\left[-\phi(\mathbf{r}_{incr})/k_{\mathrm{B}}T\right]\rangle_{\{N_1,\cdots,N_\nu+1,\cdots\}}, \tag{A.40c}$$

where $\phi(\mathbf{r}_{incr})$ is the interaction energy of a trial monomer inserted at the end of a ν-mer chain (e.g., the $\nu+1$ monomer), in a system composed of $\{N_\alpha\}$ chains with one additional ν-mer chain. Because of the bond potential, only a small volume at the end of a chain is available for insertion. The rest of the volume of the system makes no contribution to the Boltzmann average. Therefore, we will only average contributions from this biased volume, V_{ins}. The expression for incremental chemical potential becomes

$$\mu_{incr}(\nu,\{N_\alpha\}) = -k_{\mathrm{B}}T\ln\left[\frac{V_{ins}(N_\nu+1)}{\Lambda^3(N_{\nu+1}+1)}\langle\exp\left[-\phi(\mathbf{r}_{incr})/k_{\mathrm{B}}T\right]\rangle_{\{N_1,\cdots,N_\nu+1,\cdots\},V_{ins}}\right] \tag{A.41}$$

There are several important points to make regarding Eq. (A.40c). First, if a chain of ν monomers is inserted incrementally into $\{N_\alpha\}$ chains, the terms in Eq. (A.41) refer to the original system $\{N_\alpha\}$, not the individual incremental system. Second, although the V_{ins}

appears explicitly, there is no dependence on it, provided it is selected small enough to not exclude potentially favorable confirmations, and large enough to allow adequate sampling.

Applying the results above to Eq. (A.35) and rearranging terms, the full expression for the chemical potential of a chain of ν monomers in a mixture of $\{N_\alpha\}$ chains is

$$\begin{aligned}\mu_\nu(\{N_\alpha\}) = -k_{\mathrm{B}}T\ln\left[\frac{V}{\Lambda^{3\nu}(N_\nu+1)}\right] - k_{\mathrm{B}}T\ln\left[\left\langle\exp\left[-\phi(\mathbf{r}_0)/k_{\mathrm{B}}T\right]\right\rangle_{\{N_\alpha\},V}\right] \\ -\sum_{i=1}^{\nu-1}\left\{k_{\mathrm{B}}T\ln\left[V_{ins}\left\langle\exp\left[-\phi(\mathbf{r}_{incr})/k_{\mathrm{B}}T\right]\right\rangle_{\{N_1,\cdots,N_i+1,\cdots\},V_{ins}}\right]\right\}\end{aligned} \tag{A.42}$$

$$\mu_\nu(\{N_\alpha\}) = \mu_\nu^{id}(\{N_\alpha\}) + \mu_0^{ex}(\{N_\alpha\}) + \sum_{i=1}^{\nu-1}\mu_{incr}^{ex}(i,\{N_\alpha\}) \tag{A.43}$$

where μ_ν^{id} is an ideal reference state for the ν component, composed of $N_\nu+1$ chains with no external or internal interactions. Note that once the entire ν-mer chain is inserted, the $\log[(N_\nu+1)/(N_{\nu+1}+1)]$ terms from each incremental chemical potential cancel, except for the final $\log(1/N_\nu)$ term which remains in the reference state.

A.2 Extension to the Mesocanonical Ensemble

Statistical Mechanical Background

Suppose a gauge cell of volume V_g is placed in contact with the system of volume V described above. Both are held at a constant temperature T. The main cell contains N n-mers, plus one test chain of n_t particles, while the gauge contains a fluid of ideal *monomers*, with n_g particles. We allow the gauge to exchange particles with the terminal positions of the test chain. The partition function for the main cell (Q_{NVT}) and the gauge cell (Q_g) are

$$Q_{NVT}(\{N,n;1,n_t\},V,T) = \frac{1}{\Lambda^{3(Nn+n_t)}(N+1)!}\int_V d\mathbf{r}^{Nn+n_t}\exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t})\right], \tag{A.44}$$

$$Q_g(n_g,V_g,T) = \frac{V_g^{n_g}}{\Lambda^{3n_g}n_g!}. \tag{A.45}$$

It is important to note that the gauge cell fluid does not need to be ideal; the follow derivations are completely valid for gauge fluids with intermolecular potentials. However, the ideal case is later applied for the ideal gas gauge cell (IGGC) method, and supposing it here simplifies the expressions significantly. If we denote the composite system with subscript Σ, and realizing that $n_\Sigma = n_t + n_g$, we obtain the partition function for the composite system,

$$\begin{aligned} Q_\Sigma(\{N,n;1,n_\Sigma\},V,V_g,T) &= \sum_{n_t=0}^{n_\Sigma} Q_{NVT}(\{N,n;1,n_t\},V,T)Q_g(n_\Sigma - n_t,V_g,T) \\ &= \sum_{n_t=0}^{n_\Sigma} \frac{V_g^{n_\Sigma - n_t}}{\Lambda^{3(Nn+n_\Sigma)}(N+1)!(n_\Sigma - n_t)!} \int\limits_V d\mathbf{r}^{Nn+n_t} \exp\left[-\beta\Phi(\mathbf{r}^{Nn+n_t})\right]. \end{aligned} \quad \text{(A.46)}$$

From this description of the composite system, equations for the chemical potential and MC acceptance probabilities can be derived.

A.3 Mean Density Gauge Cell

Equilibrium of the two systems implies the minimization of the total Helmholtz free energy of the two cells in the same fashion as the Gibbs ensemble method,[69], in this case the free energy of test chain and the gauge cell,

$$F_t(\{N,n;1,n_t\},V,T) + F_g(n_g,V,T) \Rightarrow \min. \quad \text{(A.47)}$$

Minimization of Eq. (A.47) leads to the equilibrium condition, given by

$$\mu_{incr} = \left(\frac{\partial F_t}{\partial n_t}\right)_{\{N,n\},V,T} = \left(\frac{\partial F_g}{\partial n_g}\right)_{V_g,T}. \quad \text{(A.48)}$$

Thus, for large enough n_t and n_g, we can calculate the incremental chemical potential of the test chain by using the gauge cell as reference. Once equilibrium is reached, we assume there is an average number of particles in the test chain ($\overline{n_t}$) and a corresponding average number of monomers in the gauge cell ($\overline{n_g}$). Since the gauge fluid is defined as an ideal gas, the expression for the incremental chemical potential is

$$\mu_{incr}(\overline{n_t}) = -k_\mathrm{B}T \ln\left(\frac{V_g}{\Lambda^3(\overline{n_g}+1)}\right). \quad \text{(A.49)}$$

A.4 Harmonic Bond Reference State

If bond contributions are not sampled during the simulation, a correction to the reference state defined in Eq. (A.26) must be made. In this section, we assume the reference state is an ideal gas of chains with only internal interactions, specifially harmonic bonds. We also assume the monomers are independent of each other. The molecular partition function for such a system is

$$q(n,V,T) = \frac{V}{\Lambda^3}\left[\frac{1}{\Lambda^3}\int\limits_{V_b}\exp\left(-\frac{U_{bond}(r)}{k_\mathrm{B}T}\right)dr\right]^{n-1}. \tag{A.50}$$

The first coefficient is contributed by insertion of a single ideal gas particle in a system with volume V. The partition function for each of the remaining bonds is the product of the integral of their Boltzmann factor over the volume the bond can occupy, V_b. The partition function of the total gas of indistinguishable particles is then

$$Q(\{N,n\},V,T) = \frac{q^N}{N!} = \frac{V^N}{\Lambda^{3N}N!}\left[\frac{1}{\Lambda^3}\int\limits_{V_b}\exp\left(-\frac{U_b(r)}{k_\mathrm{B}T}\right)dr\right]^{N(n-1)}. \tag{A.51}$$

To calculate the chemical potential of this reference system, we proceed exactly as above, by using Eq. (A.4) to define chemical potential and substituting Eq. (A.51) into Eq. A.2. The ratio of the two partition functions is

$$-\frac{\mu}{k_\mathrm{B}T} = \ln\left[\frac{Q(\{N+1,n\},V,T)}{Q(\{N,n\},V,T)}\right] \tag{A.52}$$

$$= \ln\left[\frac{V}{\Lambda^3(N+1)}\left\{\frac{1}{\Lambda^3}\int\limits_{V_b}\exp\left(-\frac{U_b(r)}{k_\mathrm{B}T}\right)dr\right\}^{n-1}\right] \tag{A.53}$$

$$= \ln\left[\frac{V}{\Lambda^{3n}(N+1)}\right] + (n-1)\ln\left[\int\limits_{V_b}\exp\left(-\frac{U_b(r)}{k_\mathrm{B}T}\right)dr\right]. \tag{A.54}$$

We now arrive the reference state used above, the ideal gas of non-interacting chains (μ_{igc}), plus $n-1$ contributions of the bond potential. The bond is defined as a simple harmonic potential,

$$U_b = \frac{\kappa}{2}\left(r-\ell\right)^2, \tag{A.55}$$

where κ is the spring constant, r is the distance between the i and $i{+}1$ monomers, and ℓ is the equilibrium bond length. Next, we assume that the volume V_b contains all contributions from the bonded potential; that is, U_b is rigorously zero for all of the system outside of volume V_b. By supposing this, we can replace the finite bounds with infinite ones,

$$\begin{aligned}\int\limits_{V_b} \exp\left[-\frac{\kappa}{2k_{\mathrm{B}}T}\left(r-\ell\right)^2\right] dr &= \int\limits_{r_{ins}^{min}}^{r_{ins}^{max}} \exp\left[-\frac{\kappa}{2k_{\mathrm{B}}T}\left(r-\ell\right)^2\right] dr && \text{(A.56)}\\ &= 4\pi\int\limits_{-\infty}^{\infty} r^2 \exp\left[-\frac{\kappa}{2k_{\mathrm{B}}T}\left(r-\ell\right)^2\right] dr && \text{(A.57)}\\ &= 4\pi\left[\frac{\ell^2\sqrt{2\pi}}{\sqrt{\frac{\kappa}{k_{\mathrm{B}}T}}} + \frac{\sqrt{2\pi}k_{\mathrm{B}}T}{\kappa\sqrt{\frac{\kappa}{k_{\mathrm{B}}T}}}\right]. && \text{(A.58)}\end{aligned}$$

Although a closed-form solution is possible with finite bounds, the expression is very long and complex. This yields identical results to the method above, provided that V_{ins} when sampling bonds is large enough to contain all contributions (this is especially relevant at high temperatures when the average bond length is appreciably longer than the equilibrium bond length). The final expressions for the chemical potential are

$$-\frac{\mu}{k_{\mathrm{B}}T} = \ln\left[\frac{V}{\Lambda^{3n}(N+1)}\right] + (n-1)\ln\left[\frac{4\sqrt{2}\pi^{3/2}(\kappa\ell^2+k_{\mathrm{B}}T)}{\kappa\sqrt{\frac{\kappa}{k_{\mathrm{B}}T}}}\right] \tag{A.59}$$

$$\mu = \mu_{igc} + (n-1)\mu_{incr}^{bond}, \tag{A.60}$$

where the incremental chemical potential of the reference chain is

$$\mu_{incr}^{bond} = -k_{\mathrm{B}}T\ln\left[\frac{4\sqrt{2}\pi^{3/2}(\kappa\ell^2+k_{\mathrm{B}}T)}{\kappa\sqrt{\frac{\kappa}{k_{\mathrm{B}}T}}}\right]. \tag{A.61}$$

A.5 Chain with rigid bonds

Particle Insertion Approach

Consider a system containing N_i linear chains each composed of i beads. All beads are taken to be the same species. The number of chains of any length are contained in set $\{N_\alpha\} = \{N_1, N_2, \cdots, N_i, N_{i+1}, \cdots\}$, where N_1 is the number of *non-bonded* monomers, N_2 is the number of dimers, and so on. The total number of molecules (chains) in the system is

$$M = \sum_{i=1}^{\infty} N_i. \tag{A.62}$$

Similarly, the total number of monomers (beads) in the system is

$$m = \sum_{i=1}^{\infty} iN_i. \tag{A.63}$$

The beads interact though non-bonded potential functions and are rigidly bonded to the subsequent monomer of a chain with a constant bond length ℓ. This implies that each $i+1$ monomer can only exist on the surface of a sphere with radius ℓ, centered at $\vec{r}_i$. The first monomer of any chain ($i = 1$) can exist anywhere in the system volume V. The canonical partition function for such a system is

$$Q(\{N_\alpha\}, V, T) = \frac{1}{\prod_{i=1}^{\infty}({\Lambda_i}^{3N_i} N_i!)} \times \int_{(D_j^i)^m} \exp\left[-\frac{U(\mathbf{r}^m)}{k_\mathrm{B}T}\right] d\mathbf{r}^m \tag{A.64}$$

where $\Lambda_i = \sqrt{h^2/2\pi \sum_{j=1}^{i}(m_j) k_\mathrm{B}T}$ is the thermal de Broglie wavelength of an i-mer and m_j is the mass of a constituent bead in the j-th position, $\mathbf{r}^m$ is the set of all m position vectors $\{\vec{r}_1, \cdots, \vec{r}_m\}$, (D_i^j) is the domain of integration for a bead at position j of an i-mer, and U is the potential energy of a given set of monomer positions, defined through non-bonded, monomer interactions (e.g. the Lennard-Jones potential), external potential (an adsorption or electrostatic potential), and internal stiffness potential (such as bond angle or torsion potentials). Equation (A.64) fully describes the system of $\{N_\alpha\}$ chains in a constant volume V at constant temperature T. The domain of integration for a given bead k (at position j

on an i-mer) is determined by the connectivity of the chain,

$$\left(D_i^j\right)_k = \begin{cases} V, & \text{for } j = 1 \\ S_k, & \text{otherwise} \end{cases} \tag{A.65}$$

where S_k is the surface of a sphere of radius ℓ centered at $\vec{r}_{k-1}$. For example, suppose we have a system composed of 2 monomers and 2 dimers ($N_1 = 2, N_2 = 2$), for a total of $m = 6$ beads. The configurational integration for this system would be

$$\int\limits_V \int\limits_V \int\limits_V \int\limits_{S_4} \int\limits_V \int\limits_{S_6} \exp\left[-\frac{U(\mathbf{r}^6)}{k_\mathrm{B}T}\right] d\vec{r}_1\, d\vec{r}_2\, d\vec{r}_3\, d\vec{r}_4\, d\vec{r}_5\, d\vec{r}_6.$$

By defining the integral domains as such, there is no loss of information compared to the general approach of integration over the entire volume for all positions.

The Helmholtz free energy of the $\{N_\alpha\}$ system is related to the partition function by

$$F(\{N_\alpha\}, V, T) = -k_\mathrm{B}T \ln Q(\{N_\alpha\}, V, T). \tag{A.66}$$

From the fundamental equation of thermodynamics, the chemical potential of a ν-mer is the change in free energy per molecule at constant volume and temperature,

$$\mu_\nu(\{N_\alpha\}) = \left.\frac{\partial F}{\partial N_\nu}\right|_{\{N_{\alpha\neq\nu}\},V,T}, \tag{A.67}$$

which can be represented by a finite difference in the thermodynamic limit of $N_\nu \to \infty$,

$$\mu_\nu(\{N_\alpha\}) = F(\{N_1, \cdots, N_\nu + 1, \cdots\}, V, T) - F(\{N_\alpha\}, V, T). \tag{A.68}$$

Equation (A.68) represents the chemical potential of an ν-mer in a fluid of $\{N_\alpha\}$ chains. Now suppose a single chain of $\nu{+}1$ monomer is inserted in the system described by $Q(\{N_\alpha\}, V, T)$. The chemical potential of this addition would be

$$\mu_{\nu+1}(\{N_\alpha\}) = F(\{N_1, \cdots, N_{\nu+1} + 1, \cdots\}, V, T) - F(\{N_\alpha\}, V, T). \tag{A.69}$$

Similarly, Eq. (A.69) represents the chemical potential of an $(\nu+1)$-mer in a fluid of $\{N_\alpha\}$ chains. We can now define the difference of chemical potentials between a chain of ν beads and a chain of $\nu+1$ beads as the **incremental chemical potential**, illustrated by substracting Eq. (A.68) from (A.69),

$$\begin{aligned}
\mu_{inc}(\nu) &\equiv \mu_{\nu+1}(\{N_\alpha\}) - \mu_\nu(\{N_\alpha\}) && \text{(A.70)}\\
&= F(\{N_1,\cdots,N_{\nu+1}+1,\cdots\},V,T) - F(\{N_1,\cdots,N_\nu+1,\cdots\},V,T) && \text{(A.71)}\\
&= -k_\mathrm{B}T\ln\left[\frac{Q(\{N_1,\cdots,N_{\nu+1}+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}\right]. && \text{(A.72)}
\end{aligned}$$

To obtain an expression for μ_ν useful for molecular simulation, we begin by substituting Eq. (A.66) into (A.68), and then expanding:

$$\mu_\nu(\{N_\alpha\}) = -k_\mathrm{B}T\ln\left[\frac{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu,\cdots\},V,T)}\right] \tag{A.73}$$

$$\begin{aligned}
= -k_\mathrm{B}T\ln\Bigg[&\frac{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_{\nu-1}+1,\cdots\},V,T)}\times\frac{Q(\{N_1,\cdots,N_{\nu-1}+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_{\nu-2}+1,\cdots\},V,T)}\times\cdots\\
&\times\frac{Q(\{N_1+1,\cdots,N_\nu,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu,\cdots\},V,T)}\Bigg]. \qquad \text{(A.74)}
\end{aligned}$$

By substituting the above definition of μ_{inc}, we see it is possible to realize the chemical potential of chain by summing its incremental values. Rewriting Eq. (A.74) in terms of incremental chemical potential as defined by Eq. (A.72),

$$\mu_\nu(\{N_\alpha\}) = \sum_{i=1}^{\nu-1}\mu_{inc}(i) + \mu_0(\{N_\alpha\}), \tag{A.75}$$

where μ_0 is the chemical potential of a unbonded monomer inserted in the system of $\{N_\alpha\}$ chains, emerging from the last term in Eq. (A.74). Treating this term first,

$$\mu_0(\{N_\alpha\}) = -k_\mathrm{B}T\ln\left[\frac{Q(\{N_1+1,\cdots,N_\nu,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu\cdots\},V,T)}\right] \tag{A.76a}$$

$$= -k_{\mathrm{B}}T \ln \left[\frac{\prod_{i=1}^{\infty}({\Lambda_i}^{3N_i} N_i!)}{{\Lambda_1}^3 (N_1+1)\prod_{i=1}^{\infty}({\Lambda_i}^{3N_i}N_i!)} \times \frac{\underset{(D_i^j)_1}{\int} \cdots \underset{(D_i^j)_m}{\int} \underset{(D_i^j)_{m+1}}{\int} \exp\left[-\frac{-U(\mathbf{r}^m)}{k_{\mathrm{B}}T}\right] \exp\left[-\frac{U(\vec{r}_{m+1})}{k_{\mathrm{B}}T}\right] d\mathbf{r}^m\, d\vec{r}_{m+1}}{\underset{(D_i^j)_1}{\int} \cdots \underset{(D_i^j)_m}{\int} \exp\left[-\frac{-U(\mathbf{r}^m)}{k_{\mathrm{B}}T}\right] d\mathbf{r}^m} \right]. \tag{A.76b}$$

Equation (A.76b) contains a thermal average of an integral. The domain for this integral is $(D_1^1)_{m+1} = V$, by Eq. (A.65), since it is known from the partition function ratio that the $m+1$ bead is a non-bonded monomer. Thus, Eq. (A.76b) becomes

$$\mu_0(\{N_\alpha\}) = -k_{\mathrm{B}}T \ln\left[\frac{V}{{\Lambda_i}^3(N_1+1)}\right] - k_{\mathrm{B}}T \ln\left[\frac{1}{V}\int_V d\vec{r}_0 \left\langle \exp\left[-\frac{U(\vec{r}_0)}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_\alpha\}}\right] \tag{A.77a}$$

$$= -k_{\mathrm{B}}T \ln\left[\frac{V}{{\Lambda_i}^3(N_1+1)}\right] - k_{\mathrm{B}}T \ln \left\langle \exp\left[-\frac{U(\vec{r}_0)}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_\alpha\},V} \tag{A.77b}$$

$$= \mu_0^{id}(\{N_\alpha\}) + \mu_0^{ex}(\{N_\alpha\}), \tag{A.77c}$$

where $\vec{r}_0$ replaces $\vec{r}_{m+1}$ and represents a bead randomly inserted into the volume V containing $\{N_\alpha\}$ chains, and the brackets $\langle\cdots\rangle$ represent a canonical average. Equation (A.79) an analog for the orginial Widom insertion method of mixtures.

An expression for μ_{inc} can be obtained in a similar fashion. Beginning with the definition presented in Eq. (A.72),

$$\mu_{inc}(\nu) = -k_{\mathrm{B}}T \ln\left[\frac{Q(\{N_1,\cdots,N_{\nu+1}+1,\cdots\},V,T)}{Q(\{N_1,\cdots,N_\nu+1,\cdots\},V,T)}\right] \tag{A.78a}$$

$$= -k_{\mathrm{B}}T \ln \left[\frac{{\Lambda_\nu}^3(N_\nu+1)\prod_{i=1}^{\infty}({\Lambda_i}^{3N_i} N_i!)}{{\Lambda_{\nu+1}}^3 (N_{\nu+1}+1)\prod_{i=1}^{\infty}({\Lambda_i}^{3N_i}N_i!)} \times \frac{\underset{(D_i^j)_1}{\int} \cdots \underset{(D_i^j)_{m+\nu}}{\int} \underset{(D_i^j)_{m+\nu+1}}{\int} \exp\left[-\frac{-U(\mathbf{r}^{m+\nu})}{k_{\mathrm{B}}T}\right] \exp\left[-\frac{U(\vec{r}_{m+\nu+1})}{k_{\mathrm{B}}T}\right] d\mathbf{r}^{m+\nu}\, d\vec{r}_{m+\nu+1}}{\underset{(D_i^j)_1}{\int} \cdots \underset{(D_i^j)_{m+\nu}}{\int} \exp\left[-\frac{-U(\mathbf{r}^{m+\nu})}{k_{\mathrm{B}}T}\right] d\mathbf{r}^{m+\nu}} \right]. \tag{A.78b}$$

Just as with the non-bonded monomer case of μ_0, there is a thermal average in Eq. (A.78b). However, the domain is not the system volume V. Instead , the domain $(D_i^j)_{m+\nu+1} =$

$S_{m+\nu+1}$, and is the surface of a sphere centered at the terminal monomer in the ν-th position of the additional ν-mer chain in the system $\{N_\alpha\}$. Thus, Eq. (A.78b) can be reduced to

$$\mu_{inc}(\nu) = -k_{\mathrm{B}}T \ln\left[\frac{{\Lambda_\nu}^3(N_\nu+1)}{{\Lambda_{\nu+1}}^3(N_{\nu+1}+1)}\right] - k_{\mathrm{B}}T \ln\left[\int\limits_{S_{ins}} d\vec{r}_{ins}\left\langle \exp\left[-\frac{U(\vec{r}_{ins})}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_1,\cdots,N_\nu+1,\cdots\}}\right] \tag{A.79a}$$

$$= -k_{\mathrm{B}}T \ln\left[\frac{{\Lambda_\nu}^3(N_\nu+1)}{{\Lambda_{\nu+1}}^3(N_{\nu+1}+1)}\right] - k_{\mathrm{B}}T \ln\left\langle \exp\left[-\frac{U(\vec{r}_{ins})}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_1,\cdots,N_\nu+1,\cdots\},S_{ins}} \tag{A.79b}$$

$$= \mu_{inc}^{id}(\nu) + \mu_{inc}^{ex}(\nu), \tag{A.79c}$$

where $\vec{r}_{ins}$ is the position of an inserted bead on the surface of a surface of a sphere S_{ins} with radius ℓ centered at $\vec{r}_{m+\nu}$, the termainl bead of the inserted ν-mer. The canonical average in Eq. (A.79b) is the average Boltzmann factor of this insertion, over a system of $\{N_1, \cdots, N_\nu + 1, \cdots\}$ chains and on the surface of the sphere S_{ins}. Putting everything together, a single expression for μ_ν is obtained,

$$\mu_\nu(\{N_\alpha\}) = -k_{\mathrm{B}}T \ln\left[\frac{V}{{\Lambda_\nu}^3(N_\nu+1)}\right] - k_{\mathrm{B}}T \ln\left\langle \exp\left[-\frac{U(\vec{r}_0)}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_\alpha\},V}$$
$$- \sum_{i=1}^{\nu-1} k_{\mathrm{B}}T \ln\left\langle \exp\left[-\frac{U(\vec{r}_{ins})}{k_{\mathrm{B}}T}\right]\right\rangle_{\{N_1,\cdots,N_i+1,\cdots\},S_{ins}} \tag{A.80}$$

B CHAINBUILD code

`CHAINBUILD` is a FORTRAN90 program in which the incremental gauge cell method is implemented. It was used for all work presented in this thesis, with the exception of the first cavitation work (Section 3.1). It is capable of simulating single chains with harmonic or rigid bonds, with the Lennard-Jones (LJ) potential and an externally supplied adsorption potential. Additionally, it can simulate LJ fluids in bulk or spherical pores. Canonical, grand canonical, and mesocanonical ensembles are implemented. The sections below refer to individual files. The program is constantly evolving to meet current needs; as such, the code below is a snapshot at the time of writing.

B.1 main.f

```
ccccccccccccccccccccccccccccccccccccccccccccc
ccc            MAIN - chainbuild             ccc
ccc     Calculate chemical potential in      ccc
ccc             chain molecules.             ccc
ccccccccccccccccccccccccccccccccccccccccccccc
      program chainbuild

      use inputs  !Contains system parameter variables and IO subroutines
      use outputs !Contains subroutines to write at levels
      use sample
      use fort_rand !Contains random number function and initializer
      use energy
      use moves !Contains energy variables, and subroutines to
               !  calculate energies and preform MC move_pct
      use constants

      implicit none

      integer INI_MOVES
      parameter (INI_MOVES=200) ! Number of moves per bead to initialize
         system

      real*8 :: start_time,finish_time,runtime
      real*8 :: xx,yy,zz
      real*8 :: xn,yn,zn,dd,r,rr,theta
      real*8 rnd_move
      real*8 dBw,pf1,pf2 !de Broglie wavelength
      real*8 olduff, oldukbl, oldusf, efail !for testing conservation of E
      real*8 Vg_avg !average of Vg used in equilbration phase
      integer :: i,j,k,w,nset1,vg_avg_n,sets !counters
      logical overlap
      ! Data for storing a sequence of moves
      integer,allocatable :: seq_mvtype(:), seq_molid(:)
      real*8,allocatable :: seq_oldpos(:,:)
      integer seq_step, seq_set, seq_steps
      logical suc, exitnormal

      !!!!!!!!!!!!!!!!!!!!!!!!
      !!     Initialize     !!
      !!!!!!!!!!!!!!!!!!!!!!!!

      ! Read in initial configuration and run settings
      call readinput()
      ! Intialize the random number generator
      call rand_init(rseed)
      call readcoor() !Read in coordinates

      ! Equations, constants, and variable initialization
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      n_dis=0; n_rep=0; mu=0; Eins=0; mu_avg=0
      mu_set=0; mu_nid=0; Uff_avg=0; rho=0; Vg_avg=0
      mass=mass/Na/1000. !kg/molecule
      beta=1/T/eps
      step_warning = .false.
      ! insertion volume
      select case (ins_geom)
      case (1)
         Vins=4.0*pi*(rins_max**3.-rins_min**3.)/3.0
      case (3)
         vins=8.0*(rins_max**3.)!-rins_min**3.)
      case default

      end select

      sig=sqrt(T/(2.0*kb0_len)); aa=(r0+3.0*sig)**2 ! values relating to
          generation of harmonic bonds
      dBW=h/sqrt(2.0*pi*mass*kb*T*eps) !m
      dBW=dBw*1e10/sigma !per sigma (sigma in variable stored in Angstroms)
      insprob=0.5
      allocate( seq_mvtype(maxsavesteps), seq_molid(maxsavesteps) )
      allocate( seq_oldpos(3,maxsavesteps) )

      if (bonds) then !! .and. ensemble.ne.2
         if (.not.ins_bias) Vins=1.0 ! "Fixed Distribution"
         a=Vins/Vg
         if (harm_len) then
            mu_id=-T*log(4.0*pi*sqrt(2.0*pi)*(2.0*kb0_len*r0**2+T)/
     &      (2.0*kb0_len*sqrt(2.0*kb0_len/T))) !per epsilon
         endif
         if (ins_bias) mu_id=-T*log(Vins)
      else
         select case (ensemble)
         case (2)
            a=Volume/Vg
         case (1)
            a=Volume*exp(mu_bulk/T)/dBw**3
            Ng=1
         end select
         mu_id=T*log(dBw**3)
      endif

      rcut2=rcut*rcut
      if (hardbond) then
        step_dis = 1.5
      else
        step_dis=0.1
      endif
      cs_stp=0.5
      sets=(nset+neqset)
      r=Li(1)/2.

      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      call initialize_system()
      call writecoor(coorfile)

      ! Print move matrix
      print *, "Selected move percentages:"
      do k=1,nmoves
        write(*,510) movename(k),move_pct(k), move_cumpct(k)
      enddo
510   format(A20,F10.3,F10.3)
      !!
```

```
 if (run_ini) then
     !! MINIMIZATION --
     call calc_energy(.false.)
     Print *, 'Minimizing initial configuration...'
     print *, ' Old System Energy = ', Uext+Uint
     do j=1,INI_MOVES
         do i=1,n
            call displacement(suc)
         enddo
     enddo
     print *,"Completed ",n*INI_MOVES,"minimiziation moves!"
 endif
 call calc_energy(.true.)
 print *,' Initial System Energy = ',Uext+Uint

 !call calc_energy(.true.) !Calculate initial energy
   write(*,'(A,E13.6,a,F8.3,a,E13.6,a,F8.3,A)')
&    ' Initial Configuration: Uff= ',
&   Uff,'(',Uff/float(n),' per bead)'
 write(*,'(A29,E13.6,A,F8.3,A)') 'Usf= ',Usf,'(',
&   Usf/float(n),' per bead)'
 write(*,'(A29,E13.6,A,F8.3,A)') 'Ukbl=',Ukbl,'(',
&   Ukbl/float(n),' per bead)'
 write(*,'(A29,E13.6,A,F8.3,A)') 'Ukba=',Ukba,'(',
&   Ukba/float(n),' per bead)'

 do i=1,N
    x_org(i)=x(i)
    y_org(i)=y(i)
    z_org(i)=z(i)
 enddo

 !End initialization
 if (calc_traj) call writetraj(0)
 if (bonds) print *,"RADIUS OF GRYATION = ",radius_gryation()

 write(*,*) 'Starting simulation . . .'

 call cpu_time(start_time)

 !!!!!!!!!!!!!!!!!!!!!!!!!
 !! MAIN LOOP OVER SETS !!
 !!!!!!!!!!!!!!!!!!!!!!!!!
 vg_avg_n=0; seq_step=0;

 do set=1,sets
 !Initialize variables
    !! MOVE COUNTERS !!
    acc_dis=0; acc_rep=0; acc_cs=0; n_reg_acc=0
    n_dis=0; n_rep=0; n_cs=0; n_reg=0
    ng_rem=0; ng_rem_suc=0
    ng_ins=0; ng_ins_suc=0
    exitnormal=.false.

    !! CHEM. POT. !!
    mu=0.; Eins=0.
    mu_set=0.; mu_nid=0.
    rhogauge=0.; rho_set=0
    U_set=0;
```

```
         UFF_Set=0; USF_Set=0
         Ukbl_set=0; Ukba_set=0
         n_widom=0; N_set=0


      do step=1,nstep ! Loop over steps.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Randomly select one of the Monte-Carlo move_pct
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


500   continue

         !print *, "UFF (from MAIN)", Uff
         rnd_move=rnd()
         if (rnd_move.lt.move_cumpct(1)) then
         !Displace particle on chain
            call displacement(suc)
            k=1
         elseif (rnd_move.lt.move_cumpct(2)) then
         !Shift particles down the chain by 1
            call reptation()
            k=2
         elseif (rnd_move.lt.move_cumpct(3)) then
            if (n.gt.cs_len) then
               call crankshaft(cs_len)
               k=5
            else
               goto 500
            endif
         elseif (rnd_move.lt.move_cumpct(4)) then
            if (ensemble.ge.1) then
               call gauge_exchange(suc)
               k=3
            else
               !call bulk_exchange()
               !k=4
            endif
         elseif (rnd_move.lt.move_cumpct(5)) then
            if (n.gt.3) then
               call regrow_move(ktrials,N-rndint(N/2+1))
            else
               goto 500
            endif
            k=6
         else
            write(*,*) "Error selecting MC move_pct!"
            write(*,*) rnd_move
         endif

         ! See if we're saving a trajectory of steps to contruct cavitation
             event
         if (suc .and. maxsavesteps > 0) then
            seq_step = seq_step + 1
            if (seq_step > maxsavesteps) then
               seq_step = 1
            end if
            seq_mvtype(seq_step) = k ! Store type of move
```

```
       seq_molid(seq_step) = MOLID ! Store attempted move's molecule ID
       seq_oldpos(:,seq_step) = (/ XOLD, YOLD, ZOLD /) ! Store
          attempted move's previous position
       seq_set=set
       seq_steps=step
       if ((rhorec_lt .and. N/Volume < rhostop) .or.
& (rhorec_gt .and. N/Volume > rhostop)) then
           maxsavesteps=-1 ! minus one is sign to stop, but print
              results later
           call writecoor(trim(jobname)//".cavcoor")
       end if
   end if

 !! Test for conservation of energy, if requested
 if (energy_conserve) call isenergyconserved(cons_type,k)

 ! SAMPLING -----------------------------------------------------
 ! Tests and histograms if simulation is far enough along
 if (set.gt.nEqSet) then
    ! Test insertion for NVT
    if (calc_widom) call insert()
    ! update N_gauge histogram if MCE
    if (ensemble.ge.1) call samplegauge(.false.)
 endif

 ! Update Energies
 !U_SET=U_SET+(UFF+UKBl+UKBa+USF)
 UFF_SET=UFF_SET+UFF
 USF_SET=USF_SET+USF
 if (bonds) Ukbl_SET=Ukbl_SET+Ukbl
 if (bonds) UKBa_SET=UKBa_SET+UKBa
 N_set=N_set+N
 !if (ensemble.eq.2) rhogauge = rhogauge + dble(NG)/(nstep*VG)

 if (rhostop_lt .and. N/Volume < rhostop) exit
 if (rhostop_gt .and. N/Volume > rhostop) exit

 enddo ! over steps ----------------------------------------

   N_set=N_set/float(nstep)
   rhogauge = (Ntotal - N_set) / VG

 ! Automatically adjust gauge-sze
 ! the equilibriation stage has three parts- adjust the gauge, no
     averaging;
 !  adjust the gauge, with averaging; and take the average gauge and
     set so simulation
 !  can start.  This will hopefully even out some fluctuations
     observed .
 if (set.lt.neqset .and. autogauge) then
    if (set.gt.(neqset/10).and.set.lt.neqset*9/10) then
       Vg_avg=Vg_avg+Vg
       vg_avg_n=vg_avg_n+1
       if (rhogauge.gt.0) then
          VG=float(NGTARGET)/(rhogauge)
       else
          VG=VG*100
       endif
    elseif (set.ge.neqset*3/4) then
```

```
      Vg=Vg_avg/vg_avg_n
   elseif (set.lt.neqset/4) then
      if (rhogauge.gt.0) then
          VG=float(NGTARGET)/(rhogauge)
      else
         VG=VG*100
      endif
   endif
   if (bonds) then
      a=Vins/Vg
   else
      a=Volume/Vg
   endif
endif

 ! Adjust step sizes for displacement
XX=(acc_dis*1.0)/(n_dis*1.0) !! FRACTION OF SUCCESSFUL move_pct
if (set.lt.neqset) then
   iF (XX.LT.0.3) then
      step_dis=step_dis/1.1
      write(*,*) "NEW DISP STEP = ",step_dis
   elseIF ((XX.gT.0.5).AND.(step_dis.LE.(Li(1)/2/1.1))) then
      step_dis=step_dis*1.1
      write(*,*) "NEW DISP STEP = ",step_dis
   endif
endif
! Adjust step sizes for crankshaft MC move
XX=(acc_cs*1.0)/(n_cs*1.0) !! FRACTION OF SUCCESSFUL move_pct
if (set.lt.neqset) then
   iF (XX.LT.0.5) then
      cs_stp=cs_stp/1.1
      write(*,*) "NEW CS STEP = ",cs_stp
   elseIF (XX.gT.0.6) then
      cs_stp=cs_stp*1.1
      write(*,*) "NEW CS STEP = ",cs_stp
   endif
endif

! Reduce Average set values
Uff_SET=UFF_SET/float(nstep)
Usf_SET=Usf_SET/float(nstep)
if (bonds) Ukbl_SET=Ukbl_SET/float(nstep)
if (bonds) Ukba_SET=Ukba_SET/float(nstep)
U_SET=Uff_set+Usf_set+Ukbl_set+Ukba_set
rho_set=N_set/Volume

if (set.eq.neqset) rho_g=0.0 ! Initialize the gauge density for MDGC

if (set.gt.nEqSet) then
   ! Calculate chemical potential
   if (ensemble.eq.0) then
      !!! widom insertion
      mu=-T*log(mu_nid/n_widom)
      rho_g = rho_g + mu_nid/n_widom ! rho_g is container for
          <exp(-Uins/kT)> here
   elseif (ensemble.eq.2) then
      ! from gauge density
      rho_g=rho_g+rhogauge ! update the running average
      mu=T*log(rhogauge)
```

```
         endif
         mu_set=mu

         ! SAMPLING ----------------------------------------------
         ! Tests and histograms if simulation is far enough along
         if (calc_bondl) call samplebonds(.false.)
         !Includes bond len distribution, RMS, Tortousity, Random Coil &
             radius of gyration
         if (calc_bonda) call sampleangles(.false.) !Calculate bond angle
             distribution
         if (calc_gr) call sample_gr(.false.) !Calculate g(r)
         if (calc_dens) call sample_density(.false.) !Calculate density
             profile

         !--------------------------------------------------------

         !Update simulation averages
         Uff_avg=Uff_avg+Uff_set
         Usf_avg=Usf_avg+Usf_set
         if (bonds) Ukba_avg=Ukba_avg+Ukba_set
         if (bonds) Ukbl_avg=Ukbl_avg+Ukbl_set
         if (ensemble.ge.0) rho=rho + rho_set
         if (ensemble.ne.1) mu_avg=mu_avg+mu_set
      endif


      !DISPLAY AFTER SET COMPLETE
      call set_output(iolev(1),iolev(2)) ! write output to screen and disk
      call writecoor(coorfile) ! write coordinates to save file
      if (calc_traj .and. mod(set,trajfreq).eq.0) call writetraj(0)
      if (bonds) print *,"RADIUS OF GRYATION = ",radius_gryation()

      if (rhostop_lt .and. N/Volume < rhostop) exit
      if (rhostop_gt .and. N/Volume > rhostop) exit
      exitnormal=.true.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      enddo ! loop over sets
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!      END OF MAIN METROPOLIS ALGORITHIM
      if (step_dis .lt. 1e-4) step_warning = .true.

      if (exitnormal) then
        set=set-1
        step=step-1
      end if

      !! Averages !!
      mu_avg=mu_avg/nset
      Uff_avg=Uff_avg/nset
      Usf_avg=Usf_avg/nset
      if (bonds) Ukbl_avg=Ukbl_avg/nset
      if (bonds) Ukba_avg=Ukba_avg/nset
      rho=rho/nset
      rho_g=rho_g/nset
      !print *, mu_avg, -T*log(rho_g), mu_id
      if (ensemble==0) mu_avg = -T*log(Vins*rho_g) ! the PROPER averages
          for mu...
      if (ensemble>=2) mu_avg = T*log(rho_g)
```

```
 !reduce histograms, averages and write to disk
 call samplegauge(.true.)
 if (calc_bondl) call samplebonds(.true.)
 if (calc_bonda) call sampleangles(.true.)
 if (calc_gr) call sample_gr(.true.)
 if (calc_dens) call sample_density(.true.)

 ! Simulation done, output duties
 call writesummary(resfile)
 call writeaverageenergies()
 call writefinalle()
 ! Write out previous moves, if required
 if (maxsavesteps /= 0) then
    ! print the coordinates, if not already done so
    if (maxsavesteps /= -1) then
       call writecoor(trim(jobname)//".cavcoor")
    else
       set = seq_set
       step = seq_steps
    end if
    open(file=trim(jobname)//’.cavtraj’,unit=909)
    write(909,’(A,I10,A,I10)’) "# Finished on set ",set,
& " at step ",step
    write(909,’(A)’) "# MOVE_TYPE   MOL_ID    OLD_POS"
    do i = seq_step, 1, -1
       write(909,*) seq_mvtype(i), seq_molid(i), seq_oldpos(:,i)
    end do
    do i = maxsavesteps, seq_step+1, -1
       write(909,*) seq_mvtype(i), seq_molid(i), seq_oldpos(:,i)
    end do
    close(909)
 end if

 call cpu_time(finish_time)
 runtime=finish_time-start_time
 write(*,*)  ’  CPU Time: ’, time_str(runtime)
 open(file=logfile,position=’APPEND’,unit=27)
 write(27,*) ’  CPU Time: ’, time_str(runtime)


 if (.not. exitnormal) then
   write(*,*) "PROGRAM STOPPED BEFORE FINISHING ALL REQUESTED"//
& " SETS!"
   write(27,*) "PROGRAM STOPPED BEFORE FINISHING ALL REQUESTED"//
& " SETS!"
 end if

 deallocate( move_pct, move_cumpct,movename )
 deallocate( seq_oldpos, seq_molid, seq_mvtype )
 if (allocated(USF_band)) deallocate(USF_band)
 close(27)
 end program chainbuild



 !! Recalculates U and compares new value to running value.
 !!  Stops program is value is less than specified amount.
 subroutine isenergyconserved(set_or_step,whichmove)
 use energy
 use inputs
 use outputs

 implicit none
 real*8 olduff, oldukbl, oldusf, efail !for testing conservation of E
 integer set_or_step,whichmove

 select case (set_or_step)

 case (1)
    !Every set
    if (step.ne.nstep) return

 case (0)
    !Every step

 case default
    write(*,*)
&  "Invalid selection for ONSET or ONSTEP conservation!"
    stop

 end select

 olduff=uff; oldusf=usf; oldukbl=ukbl
 call calc_energy(.false.)
 efail=abs(olduff-uff)+abs(oldukbl-ukbl)+abs(oldusf-usf)

 if (efail.gt.cons_tol) then
    select case (whichmove)
    case (1)
       write(*,*)
&      'Energy conservation fail on displacement step!'
    case(2)
       write(*,*)
&      'Energy conservation fail on reptation step!'
    case(3)
       write(*,*)
&      'Energy conservation fail on gauge exchange step!'
       write(*,*) 'Accepted Gauge move_pct (I/R):',
&      ng_ins_suc,ng_rem_suc
    case(5)
       write(*,*)
&      'Energy conservation fail on crankshaft step!'
    case(6)
       write(*,*)
&      'Energy conservation fail on CB regrow step!'

 case default
       write(*,*) "MC move not programmed for conservation..."
       stop
    end select

    write(*,*) ' Uff(OLD), Uff(TRUE)=',olduff,uff
    write(*,*) ' Usf(OLD), Usf(TRUE)=',oldusf,usf
    write(*,*) ' Ukbl(OLD), Ukbl(TRUE)=',oldukbl,ukbl
    write(*,*) ' Current: N=',N
    call writecoor(coorfile) ! write coordinates to save file
    stop
```

```
      endif

      return

      end subroutine isenergyconserved
```

B.2 inputs.f

```
      module inputs
      use constants
      implicit none

!     ### Compilier parameters ##
      integer maxpart,maxlines
      parameter(MAXPART=10005, MAXLINES=50)
!! MAXPART is the longest chain that can be generated
!!   Program will probably segfault if N goes over this value.
!! MAXLINES is the most allowed

!     ### INPUTS ###
! Job Details
      integer :: nset,nstep,neqset,rseed
      real*8 :: cons_tol
      logical energy_conserve, ready
! System Parameters
      integer ensemble
      real*8 :: T,mu_bulk,beta,gr_delr
      real*8 :: rho,rho_set,temp, N_set
      logical calc_widom,calc_dens,calc_hist,calc_bonda,calc_bondl
      logical recordene, recordpos, recordcom, calc_gr,calc_traj
      logical movesassigned, run_ini, autogauge,tethered(MAXPART)
      integer a_bins,bl_bins,dens_bins,cons_type,nmoves,ktrials
      integer gr_maxbin,ba_bins,n_max_regrow
      real,allocatable :: move_pct(:),move_cumpct(:)
      character*14,allocatable :: movename(:)
      character*3,allocatable :: mv_short(:)
      character*4 :: trajtype
! Chain Parameters
      integer :: Ni,Nf,Ncalc,Ntotal,Ng,ngtarget,length
      real*8 :: rins_max,rins_min,vins,Vg,rhogauge,rho_g,a
      logical gauge_auto
!!!!! Potentials !!!!!
      logical :: harm_len,harm_3bond,harm_ang,angle,cb_regrow
      logical :: softcore,lennjones,wall_pot,SetPBC(3)
      logical :: hardbond,hardcore,bonds,ins_bias
      integer ncut, ins_geom, freq_widom, cs_len
      real*8 :: the_max,the_min,kb0_len,kb0_ang,kb1_3bd,kb2_3bd
      real*8 :: eps, sigma, r0,theta0,th1,th2,rcut,rcut2,mass
      real*8 :: hibnd_HB,lobnd_HB, lj_shift
      logical tailcorr, hardwall, sphere
      logical both_ends         ! should we sample both ends of the chain?
! Solid Paramters - U=U(r)
      integer geom,layers,bands !geometry of confinement,# of layer
      real*8 Li(3),Ri(3)        !confinement dimensions (r) (r,z) or (x,y,z)
      real*8 Volume
      real*8, allocatable :: gr_hist(:),USF_Band(:,:) !tabbulated USF
         potential
! Solid Parameters - U=U(r,theta)
```

```
      integer dens_type,rlayers,tlayers,rad_bins,ax_bins ! layers in radius
         and theta
      real*8 xz_dy,omega        ! thinkness of XZ plane for planar dens,
         radians per tlayers
!----------------GG
      real*8 Di(6)              ! geometrical parameters:
! diameters D1, D2, opening size, distance between centers, l1, l2
      real*8 rod_a,rod_b,rod_c,alpha,beta ! values relating to
         parallelogram PBC --
!   tan(alpha)=b/c, c=Li(1) (x length of box), a+b=Li(2) (y length of box),
   Li(3) = z length of box
!----------------GG
!     ### OUTPUTS ###
      integer :: iolev(2),trajfreq
      integer :: maxsavesteps
      character*80 :: logfile,coorfile,resfile,trajfile
      character*80 :: solfile,inicoorfile,comfile
      character*80 :: enefile,mufile,densfile,histfile,bondlfile
      character*80 :: bondafile,jobname,posfile !,jobrfile

!     ### System Variables ###
      real*8 :: x(MAXPART),y(MAXPART),z(MAXPART)
      real*8 :: x_org(MAXPART),y_org(MAXPART),z_org(MAXPART)
      integer :: n              !current length of chain


!     ----------- LATE ADDITIONS!! --------------__!
!     ## functionalized polymers ##
      integer func_loc
      real*8 func_eps, eps_start,eps_end,eps_rep,func_s1
      logical func_chain(3), space
!     ## trajectory tracing and cavitation halting ##
      logical :: rhostop_lt, rhostop_gt, rhorec_lt, rhorec_gt
      real*8 :: rhostop

      contains

!     Read in parameters for simulation
cccccccccccccccccccccccccccccccccccccc
      subroutine readinput()
      character*50 filein
      character*50 tmp(7)
      character*80 str,calcstr
      character*1 first
      integer i,j,k,ie,linecount,lpos,ncalcs
      real summoves

      tmp(1:7)=''               ! Clear temporary strings

!if (iargc().ge.1) then
!   call getarg(1,filein)
!   open(file=filein,unit=10,status='old')
!else
!   open(file='chainbuild.stdin',unit=10,status='old')
!   filein='chainbuild.stdin'
!endif

      linecount=0
      rins_min=0.0              !Default value
      inicoorfile='NONE'        !by default
      resfile='NONE'
      rseed=0
      trajfreq=0; freq_widom=0
      lj_shift=0.d0
      ins_bias=.false.
      movesassigned=.false.
      ready=.false.
      tethered=.false.
      both_ends=.false.
      harm_len=.false.
      hardwall=.false.
      rad_bins=0; ax_bins=0
      maxsavesteps=0
      rhostop_lt=.false.; rhostop_gt=.false.
      rhorec_lt = .false.; rhorec_gt = .false.
      rhostop = 0.d0
      eps_start=1.d0; eps_end=1.d0; eps_rep=1.d0
      func_chain=.false.; space = .false.


      write(*,*) 'CHAINBUILD - MC simulation of flexible LJ chains'
      write(*,*) '================================================'

!!!!!!!!!Read input file, ignoring comments !!!!!!!!!!!!!
      do while (.not. ready)
         first='#'
         do while (first.eq.'#' .or. first.eq.'!')
            write(*,'(A)',advance='NO') "> "
            read(*,'(A80)') str
            first=str(1:1)
         enddo

!str=takestr(10,ie,linecount)
         read(str,*) tmp(1)
         tmp(1)=trim(tmp(1))
         tmp(1)=upcase(tmp(1))
c     print *,i,str

         CHECK_OPERATOR: SELECT CASE (tmp(1))

      case ('HELP')
         write(*,*) 'No help yet!'

      case ('RUN')
         write(*,*) 'Running simulation...'
         ready=.true.

      case ('NAME')
         read(str,*) tmp(1),jobname
         write(*,*) 'jobname= ',jobname

      case ('RSEED')
         read(str,*)tmp(1),rseed
         write(*,*) 'random seed= ',rseed

      case ('TRACK')            ! track level
         read(str,*)tmp(1),iolev(1),iolev(2)
```

```
    write(*,*) 'I/O tracking at levels ',iolev(1),iolev(2)

 case ('JOB')
    read(str,*)tmp(1),neqset,nset,nstep
    write(*,*) 'eq.sets,avg.sets,steps/set= ',neqset,nset,nstep

 case ('ENS')              ! ensemble
    read(str,*)tmp(1),tmp(2)
    tmp(2)=trim(tmp(2))
    tmp(2)=upcase(tmp(2))

    CHECK_ENSEMBLE: SELECT CASE (tmp(2))

 case ('NVT')
    ensemble = 0           ! NVT ensemble
    read(str,*)tmp(1),tmp(2),length,T
    print *,'Canonical ensemble selected.'
    print *,'  Monomers =',length
    write(*,*) '  Temperature (K) =',T
    calc_widom=.true.

 case ('GCE')
    ensemble = 1           ! grand canonical
    read(str,*)
*        tmp(1),tmp(2),mu_bulk,T
    print *,'Grand canonical ensemble selected.'
    print *,'  Bulk chemical potential (in epsilon_FF) =',
&        mu_bulk
    write(*,*) '  Temperature (K) =',T

 case ('MCE')
    ensemble = 2           ! gauge cell method
    read(str,*)
*        tmp(1),tmp(2),Ntotal,T,Vg
    vg=vg*vg*vg
    if (vg.eq.0) gauge_auto=.true.
    print *,'Mesocanonical ensemble selected.'
    print *,'  N_cage+N_gauge =',Ntotal
    write(*,*) '  Temperature (K) =',T
    write(*,*)'  Gauge cell volume =',Vg
    calc_hist=.true.

 case ('GAU')
    ensemble = 2           ! gauge cell method, with automatic gauge
       cell volume
    read(str,*)
*        tmp(1),tmp(2),Ntotal,T,Vg,Ngtarget
    Vg=Vg**3
    Ntotal=Ntotal+Ngtarget ! this makes input for batch jobs much
       easier.  still have to update other comments and
       documentation...
    write(*,*)'Mesocanonical ensemble selected.'
    write(*,*)'  N_cage+N_gauge =',Ntotal
    write(*,*)'  Temperature (K) =',T
    write(*,*)'  Initial gauge volume =',Vg
    calc_hist=.true.
    autogauge=.true.

 case default
```

```
         print *,'Bad Ensemble Choice!'
         print *,'Select NVT, GAU (Auto gauge cell), MCE (Ma'//
     &        'nual gauge), or GCE (Grand canonical) ensemble.'
         stop

      eND SELECT CHECK_ENSEMBLE

      case('TETHER')
! Tethering will only work on the 1st bead of the chain if using gauge or
    CBMC
! To tether multiple beads, you must use NVT and displacement moves
    (FE-CBMC will be capable of tethering both ends)
         read(str,*)tmp(1),i
         both_ends=.false.
         tethered(i)=.true.
         write(*,*) "Will fix position of Mol-ID ",i

      case ('RECORD')
         read(str,*)tmp(1),tmp(2)
         tmp(2)=upcase(tmp(2))

         CHECK_REC: select case (tmp(2))
      case ('ENERGY')
         recordene=.true.
         write(*,*) 'Will record energy values.'
      case ('POSITION')
         recordpos=.true.
         if (.not.calc_bondl) then
            calc_bondl=.true.
            bl_bins=100         !default value
         endif
         write(*,*) 'Will record position-related values.'
      case ('COM')
         recordcom=.true.
         if (.not.calc_bondl) then
            calc_bondl=.true.
            bl_bins=100         !default value
         endif
         write(*,*) 'Will record center-of-mass coordinates.'
      case ('TRAJECTORY')
         read(str,*) tmp(1),tmp(2),trajfreq,trajtype
         calc_traj=.true.
!trajtype = tidy(trajtype)
         write(*,*)'Will record trajectory every',trajfreq,
     &        'sets as ',trajtype
      case ('CAVITATION')
         read(str,*) tmp(1),tmp(2),maxsavesteps
         write(*,*)'Will save last ',maxsavesteps,' moves.'
! Can then work backward through that number of moves

      case default
         write(*,*) 'WARNING: ',trim(tmp(j)),
     &        ' is not a valid selection for RECORD!'
         stop
      end select CHECK_REC

      case  ('SAMPLE')
         read(str,*)tmp(1),tmp(2)
         tmp(2)=upcase(tmp(2))
         CHECK_CALC: select case (tmp(2))
      case ('DENSITY')
         read(str,*) tmp(1),tmp(2),dens_type,rad_bins,xz_dy
         calc_dens=.true.
         dens_bins=rad_bins
         if (dens_type.lt.4) ax_bins=int(xz_dy)
         write(*,*) "Will sample density profile with bins=",
     &        dens_bins, ax_bins

      case ('BONDLEN')
         read(str,*) tmp(1),tmp(2),bl_bins
         calc_bondl=.true.
         write(*,*) "Will sample bond lengths with bins=",
     &        bl_bins

      case ('BONDANGLE')
         read(str,*) tmp(1),tmp(2),ba_bins
         calc_bonda=.true.
         write(*,*) "Will sample bond angles with bins=",
     &        ba_bins

      case ('GR')
         read(str,*) tmp(1),tmp(2),gr_delr
         calc_gr=.true.
         write(*,*) "Will sample radial distribution with "//
     &        "delta_r=",gr_delr

      case default
         write(*,*) 'ERROR: ',trim(tmp(2)),
     &        ' is not a valid selection for SAMPLE!'

      end select CHECK_CALC

      case ('MODEL')
         read(str,*),tmp(1),tmp(2)
         call readmol(tmp(2))
         print *, "Read in ",trim(tmp(2)),"... OK!"

      case ('HARDWALL')         ! A replusive wall at z=0 (in any geometry)
! If selected, SPACE becomes BOX, so choose size appropriately.
         wall_pot=.true.
         hardwall=.true.

      case ('SOLID')
         read(str,*),tmp(1),tmp(2)
         tmp(2)=trim(tmp(2))
         tmp(2)=upcase(tmp(2))
         if (tmp(2)(1:4).eq.'BULK') then
            read(str,*)
     *           tmp(1),tmp(2),Li(1)
            SetPBC=.true.       !turn on periodic boundary conditions
            Li(:)=Li(1)
            Volume=product(Li)
            geom=3

         elseif (tmp(2)(1:5).eq.'SPACE') then
            read(str,*)
     *           tmp(1),tmp(2),Li(1)
```

```
            SetPBC=.false.      !turn off periodic boundary conditions
            Li(:)=Li(1)
            layers=3
!! This is only necessary if user selects HARDWALL.  Ignored otherwise
            allocate ( USF_Band(4,4) )
            USF_Band=0          ! zero interaction energy in box
!!!
            Volume=product(Li)
            geom=3
            space = .true.

         elseif (tmp(2)(1:3).eq.'BOX') then
            read(str,*)
     *           tmp(1),tmp(2),Li(1)
            SetPBC=.false.      !turn off periodic boundary conditions
            layers=2
            allocate ( USF_Band(4,4) )
            USF_Band=0          ! zero interaction energy in box
            wall_pot=.true.     !turn on box (eg, reject move_pct outside
                the box)
            Li(:)=Li(1)
            Volume=product(Li)
            geom=3

         elseif (tmp(2)(1:6).eq.'SPHERE') then
            read(str,*)
     *           tmp(1),tmp(2),Li(1)
            SetPBC=.false.      !turn off periodic boundary conditions
            Li(1)=Li(1)-1.0     !Convert input to Dext (by subtracting 1
                sigma.  Algorithm computes hard wall for centers of beads,
                not the "surface")
! Note for hard wall, Dint = Dext
            layers=2
            allocate ( USF_Band(1,4) )
            USF_Band=0          ! zero interaction energy in box,
            wall_pot=.true.     !turn on box (eg, reject move_pct outside
                the box)
            Li(:)=Li(1)
            Volume=pi*Li(1)**3/6
            geom=1
            write(*,'(A,F8.2)') 'Spherical confinement, D_ext/sig=',
     &           Li(1)+1.0
            write(*,'(A33,F8.2)') 'D_int/sig=',Li(1)

         else                   ! Anything else will be interpeted as a
             file name
            read(str,*) tmp(1),solfile
            call readsol(solfile) !Read in solid geometry and sold-fluid
                potential
            print *, "Read in ",trim(tmp(2)),"... OK!"

         endif
         do i=1,3
            if ( (rcut.gt.Li(i)/2).and.(SetPBC(i)) ) then
               write (*,*)"STOPPING: PBC is greater than 1/2 box width!"
               stop
            endif
         enddo
```

```
c     case ('SPHERE')
c     ! Enables use of spherical geometry insertion shortcut
c     ! Does NOT alter potential in any way
c     sphere=.true.
c     write(*,*) "Will use spherical shortcuts."

      case ('RHOSTOP')
         read(str,*),tmp(1),tmp(2),rhostop
         if (tmp(2)=="<") then
            rhostop_lt=.true.
            write(*,*) 'Will stop when rho is less than ',rhostop
         else if (tmp(2)==">") then
            rhostop_gt=.true.
            write(*,*) 'Will stop when rho is greater than ',rhostop
         else
            write(*,*) 'On RHOSTOP command, unknown symbol ',tmp(2)
            stop
         end if

      case ('RHOSTOPTRAJ')
         read(str,*),tmp(1),tmp(2),rhostop
         if (tmp(2)=="<") then
            rhorec_lt=.true.
            write(*,*)
     &           'Will stop recording cavtraj when rho is less than
        ',rhostop
         else if (tmp(2)==">") then
            rhorec_gt=.true.
            write(*,*)
     &           'Will stop recording cavtrajwhen rho is greater than
        ',rhostop
         else
            write(*,*) 'On RHOSTOP command, unknown symbol ',tmp(2)
            stop
         end if

! QnD extension to functionalized polymers
      case ('FUNC')
         read(str,*),tmp(1),func_loc,func_eps,func_s1
         if (func_loc.eq.1) then ! epsilon value 1st bead in chain
            eps_start = func_eps
            func_chain(1)=.true.
! e_F1 is actually the ratio e_F1/e_FF
! e_S1 is actually the ratio e_S1/e_SF
            write(*,*) 'Initial bead epsilon = e_F1*', func_eps
            write(*,*) 'Initial bead epsilon = e_S1*', func_s1
c     else if (func_loc.eq.2) then ! epsilon value, terminal bead
c     eps_end = func_eps
c     func_chain(2)=.true.
c     write(*,*) 'Terminal bead epsilon = e_FF*', func_eps
c     else ! epsilon value
c     eps_rep = func_eps
c     func_chain(3)=.true.
c     write(*,*) 'Repeat unit bead epsilon = e_FF*', func_eps
         else
            print *,"FUNCTIONALS CAN ONLY BE PLACED AT N=1 FOR NOW!"
            STOP
         end if
         both_ends=.false.
      case ('INITIAL')
         read(str,'(A8,A80)'),tmp(1),inicoorfile
         inicoorfile=trim(inicoorfile)
         write(*,*) 'initial coordinates= ',inicoorfile

      case ('SUMMARY')
         read(str,*),tmp(1),resfile
         resfile=trim(resfile)
         write(*,*) 'summary file= ',resfile

      case ('CONSERVE')
         read(str,*),tmp(1),tmp(2),cons_tol
         tmp(2)=upcase(tmp(2))
         tmp(2)=trim(tmp(2))
         energy_conserve=.true.
         if (tmp(2).eq.'ONSTEP') THEN
            cons_type=0
            write(*,'(A)')
     &           "Will test conservation of energy every STEP."
         elseif (tmp(2).eq.'ONSET') then
            cons_type=1
            write(*,'(A)')
     &           "Will test conservation of energy every SET."
         else
            write(*,*) "Syntax Error!"
            write(*,*) "CONserve [type] [absolute tolerance]"
            write(*,*)
     &           " [type] can be ONSET or ONSTEP; indicates when to
        recalculate U"
            write(*,*) " [absolute tolerance] is the difference in ",
     &           " U_old - U_true required to stop calculations."
         endif

      case ('STOP')
         stop
      case ('EXIT')
         stop
      case ('QUIT')
         stop

      case default
         write(*,*) ' Invalid operator ',tmp(1)


      END SELECT CHECK_OPERATOR

      end do                    ! i !


c     goto 100
c99   write(*,*)
c     &    'unable to read from the initial input file ',linecount
c     stop
c
c     100     close (10)

!     add ability to resume stopped simulation
!     add error catch if log run+debug output is requested
```

```
!This will be the default move_pct
!Maybe put this in a configuration file later?
      if ( .not.movesassigned) then
         nmoves=5
         allocate( move_pct(nmoves), move_cumpct(nmoves) )
         move_pct(:)=1
         if (ensemble.eq.0) then
! NVT, So no exchange
            move_pct(4)=0
         endif
         if (.not.bonds) then
!No bonds, so no chain molecule move_pct
            move_pct(2)=0       !Reptation
            move_pct(3)=0       !Crankshaft
            if (rins_max.ne.0) then
               write(*,*) "Warning: Insertion bias selected with ",
     &              "unbonded particles...turning off insertion bias!"
               ins_bias=.false.
            endif
         endif

!! Normallize the move percentages
         summoves=sum(move_pct)
         do j=1,nmoves
            move_pct(j)=move_pct(j)/summoves
         enddo
!! Now store them as cumulative percentages for random selection in MC
    algorithm
         move_cumpct(1)=move_pct(1)
         do j=2,nmoves
            move_cumpct(j)=move_cumpct(j-1)+move_pct(j)
         enddo
      endif


! Move names (better place to put this?)
      allocate (movename(nmoves),mv_short(nmoves))
      movename(1)="Displacement"
      mv_short(1)="DIS"
      movename(2)="Reptation"
      mv_short(2)="REP"
      movename(3)="Crankshaft"
      mv_short(3)="CS"
      movename(4)="Exchange"
      mv_short(4)="EXC"
      movename(5)="CB Regrow"
      mv_short(5)="CBR"


! Some errors I can think of-
      if ( (ncut.lt.1).and.(bonds) ) then
         write(*,*)'WARNING: nCut = 0 and bonded molecules!  Continuing...'
      endif
      if ( (ncut.ne.0).and.(.not.bonds) ) then
         write(*,*) 'Nonbonded particles!  Change nCut to 0!'
         stop
      endif
      if (ensemble.ge.1 .and. move_pct(4).le.0) then
```

```
         write(*,*) 'You must exchange particles in this ensemble!'
         stop
      endif
      if (ensemble.eq.0 .and. move_pct(4).gt.0) then
         write(*,*) 'You cannot exchange particles in this ensemble!'
         stop
      endif
      if (.not.bonds .and. (move_pct(2)+move_pct(3)).gt.0) then
         write(*,*) 'Invalid move for system without bonds!'
         stop
      endif
      if (ins_bias .and. hardbond) then
         write(*,*) 'Cannot use an insertion bias with hard bonds!'
         write(*,*) 'Will ignore this.'
         ins_bias=.false.
      endif
      if (calc_dens .and. geom.eq.2 .and.
     &     dens_type.eq.1) then
         write(*,*) "If using U=U(r,theta) external potential, please"//
     &        " sample density with DENS_TYPE=2 or 4 (RZ, or XZ;Y=0)"
         stop
      endif
      if (ensemble.eq.1 .and. bonds) then
         write(*,*) "GCMC calculates are not supported with chain mole"//
     &        "cules!"
         stop
      endif

! Output summary of potentials
      write(*,*) 'Will use the following potentials/models:'
      if (lennjones) write(*,*) ' * Lennard-Jones'
      if (harm_ang) write(*,*) ' * Harmonic Bond Angle (Bending)'
      if (harm_len) write(*,*) ' * Harmonic Bond Length (Stretching)'
      if (hardbond) write(*,*) ' * Hard Bonds'
      if (softcore) write(*,*) ' * Soft Core'
      if (hardcore) write(*,*) ' * Hard Core'
      if (harm_3bond) write(*,*) ' * 1, 3 Harmonic Bond Angles'
      if (wall_pot) write(*,*) ' * External potential from ',solfile

!! PREPARE FILE NAMES
      enefile=jobname
      mufile=jobname
      densfile=jobname
      histfile=jobname
      bondlfile=jobname
      bondafile=jobname
      trajfile=jobname
      logfile=jobname
      coorfile=jobname
      posfile=jobname
      comfile=jobname
!jobrfile=jobname

      i=len_trim(jobname)

      enefile(i+1:i+5)='.ene'   !appends
      mufile(i+1:i+4)='.mu'     !appends
      densfile(i+1:i+6)='.dens' !appends
      histfile(i+1:i+6)='.hist'
      bondlfile(i+1:i+5)='.bdl'
      bondafile(i+1:i+5)='.bda'
      trajfile(i+1:i+1)='.'     !appends
      trajfile(i+2:i+6)=trajtype !appends
      logfile(i+1:i+5)='.log'   !appends
      coorfile(i+1:i+6)='.coor'
      posfile(i+1:i+5)='.pos'   !appends
      comfile(i+1:i+5)='.com'   !appends
!jobrfile(i+1:i+5)='.jrs'

! "Delete" files instead of appending
      open(file=enefile,unit=21)
      close(21,status='DELETE')
      open(file=mufile,unit=22)
      close(22,status='DELETE')
      open(file=densfile,unit=23)
      close(23,status='DELETE')
      open(file=trajfile,unit=24)
      close(24,status='DELETE')
      open(file=trim(jobname)//'.gr',unit=24)
      close(24,status='DELETE')
      open(file=trim(jobname)//'.rg',unit=24)
      close(24,status='DELETE')
c     if (recordpos) then
c     open(file=posfile,unit=25)
c     write(25,*) ''
c     close(25)
c     endif
      open(file=comfile,unit=26)
      close(26,status='DELETE')
!open(file=jobrfile,unit=27)
!close(27,status='DELETE')

      open(file=mufile,unit=26,status='NEW')
      close(26)

!! If auto gauge is on, set 1st Vg to Volume
!if (autogauge) Vg=Volume
!! Set the maximum bin for g(r)
      if (calc_gr) then
         gr_maxbin=int(Li(1)/gr_delr)
         allocate(gr_hist(gr_maxbin))
         gr_hist=0
      endif

      if (sphere) Volume=pi*Li(1)**3/6
      Ri=Li/2
      T=T/eps

      end subroutine readinput


!     Read in coordinates
cccccccccccccccccccccccccccccccccccccccc
      subroutine readcoor()
      integer :: i
      character*80 tmpstr
      logical lexists
      if (upcase(inicoorfile).eq.’NONE’) then
! no initial coordinates specified
         n=0
!if (.not. hardbond) run_ini=.true. !run the initiallization sequence
         return
      else
         if (ensemble.eq.0) N=length
         if (ensemble.eq.1) N=MAXPART
         if (ensemble.eq.2) N=Ntotal
      endif

      inquire(file=inicoorfile, exist=lexists)
      if (.not. lexists) then
         write(*,*) "Requested initial coordinate FILE NOT FOUND: ",
     &        inicoorfile
         stop
      end if

      open(file=inicoorfile,unit=101,status=’OLD’)
! add- if no file, generate new coordinates
      tmpstr=readline(101)

      do i=1,N
         if (tmpstr.eq."EOF") exit
         read(tmpstr,*) x(i),y(i),z(i)
         tmpstr=readline(101)
      enddo
      N=i-1

      close(101)
      run_ini=.false.
      write(*,*) "Read in ",n," beads from ", trim(inicoorfile),
     %     "... OK!"
      end subroutine readcoor


!     Read in solid (SOL) file
cccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine readsol(filename)
      integer i,j,k
      character*50 filename
      character*80 tmpstr
      real*8 temp
      logical lexists
!---------------------------GG
      real*8 rho_temp, z_temp   ! to read U_sf from file
!---------------------------GG

      inquire(file=filename, exist=lexists)
      if (.not. lexists) then
         write(*,*) "SOLID FILE NOT FOUND: ", filename
         stop
      end if

      wall_pot=.true.
      open(file=filename,unit=102,status=’OLD’)
      tmpstr=readline(102)
      read(tmpstr,*) geom
      tmpstr=readline(102)
      read(tmpstr,*) Li(1),Li(2),Li(3)

      tmpstr=readline(102)
      select case (geom)
      case (1)                  ! Spherical, U=U(r)
         read(tmpstr,*) layers,bands
         allocate ( USF_Band(1,layers) )
         do i=1,layers          ! Read in tabulated potential
            tmpstr=readline(102)
            read(tmpstr,*) temp,USF_Band(1,i)
         enddo
! Misc settings
         write(*,*) 'Spherical confinement, U=U(r), D_ext/sig=',Li(1)
         layers=layers-1        ! First layer is boundary, U(r=0)
         Volume=pi*Li(1)**3/6
         SetPBC=.false.

      case (2)                  ! Spherical, U=U(r,theta)
         read(tmpstr,*) layers,rlayers,tlayers
         allocate ( USF_Band(rlayers,tlayers) )
         do i=1,layers          ! Read in tabulated potential
            tmpstr=readline(102)
            read(tmpstr,*) j,k,temp !j==radius, k==theta
            USF_Band(j,k)=temp
         enddo
! Misc settings
         write(*,*) 'Spherical confinement, U=U(r,theta), D_ext/sig='
     &        ,Li(1)
         rlayers=rlayers-1      ! First layer is boundary, U(r=0)
         tlayers=tlayers-1      ! First layer is boundary, U(theta=0)
         Volume=pi*Li(1)**3/6
         SetPBC=.false.
         omega=pi/tlayers

      case (3)
         read(tmpstr,*) layers,bands
         allocate ( USF_Band(3,layers) )
         do i=1,layers
            tmpstr=readline(102)
            read(tmpstr,*) temp,USF_Band(1,i),USF_Band(2,i),
     &           USF_Band(3,i)
         enddo
         write(*,*) 'Box confinement, Lx,Ly,Lz=',Li
         layers=layers-1        ! First layer is boundary, U(x,y,z=-L/2)
         Volume=product(Li)
         setPBC=.false.

c     case (4)
c     read(tmpstr,*) layers,bands
c     allocate ( USF_Band(3,layers) )
c     do i=1,layers
c     tmpstr=readline(102)
c     read(tmpstr,*) temp,USF_Band(1,i),USF_Band(2,i),
c     &                                 USF_Band(3,i)
c     enddo
c     write(*,*) 'Channel confinement, Lx,Ly,Lz=',Li
c     layers=layers-1 ! First layer is boundary, U(x,y,z=-L/2)
c     Volume=product(Li)
c     SetPBC(1)=.true.
```

```
c     setPBC(2:3)=.false.

      case (4)                  ! Cylindrical, U=U(r)
         read(tmpstr,*) layers,bands
         allocate ( USF_Band(1,layers) )
         do i=1,layers          ! Read in tabulated potential
            tmpstr=readline(102)
            read(tmpstr,*) temp,USF_Band(1,i)
         enddo
! Misc settings
         write(*,*) 'Cylindrical confinement, U=U(r), ' //
     &        'D_ext/sig=',Li(1),', L/sig=',Li(3)
         layers=layers-1        ! First layer is boundary, U(r=0)
         Volume=pi*Li(1)**2*Li(3)
         SetPBC(1:2) = .false.
         SetPBC(3)   = .true.

!-------------------------------------GG
      case (5)                  ! Cylindrical, U=U(z,rho)
         read(tmpstr,*) layers,rlayers,tlayers !rlayers = z, tlayers = rho
         allocate ( USF_Band(rlayers,tlayers) )

         tmpstr=readline(102)
         read(tmpstr,*) Di(1),Di(2),Di(3),Di(4),Di(5),Di(6) ! Read the additional geometry info
! diameters D1, D2, opening size, distance between centers, l1, l2

! Misc settings
         write(*,*) 'Cylindrical confinement, U=U(z,rho)'
         write(*,*) 'x, y, z (in sigma)=', Li(1), Li(2), Li(3)
         write(*,*) 'Additional: D1, D2, Op ', Di(1), Di(2), Di(3)
         write(*,*) 'Additional: Dist, l1, l2 ', Di(4), Di(5), Di(6)

         do i=1,layers          ! Read in tabulated potential
            tmpstr=readline(102)
            read(tmpstr,*) j,k,temp !j==z, k==rho
!write (*,*) "j, k: ", j, k
            USF_Band(j+1,k+1)=temp
         enddo


!     do i=1,layers ! Read in tabulated potential
!     tmpstr=readline(102)
!     read(tmpstr,*) z_temp,rho_temp,temp !j==z, k==rho
!     j = (z_temp + Di(1))/2.d0/rlayers + 1!((Di(1)+Di(2))/2.d0+Di(4))
!     k = rho_temp/tlayers + 1
!     ! Patch
!     if ((j .le. rlayers) .and. (k .le. tlayers)) then
!     write(*,*) "Getting U at (j,k) ", j, k
!     USF_Band(j,k)=temp
!     else
!     write(*,*) "Allocation error at (j,k) ", j, k
!     end if
!     enddo

         rlayers=rlayers-1      ! First layer is boundary, U(r=0)
         tlayers=tlayers-1      ! First layer is boundary, U(theta=0)
!     Volume=pi*Li(1)*(Li(2)**2)/4.0 !WRONG
         if ((Di(1) .eq. Di(2)) .and. (Di(4) .lt. 1.0e-3)) then
```

```
            Volume = pi/6.0*Di(1)**3
            write(*,*) "Limiting case for Volume"
         else
            write(*,*) "General case for Volume"
            Volume = pi/6*(Di(1)**3 + Di(2)**3)
            Volume = Volume -
     &           pi*((Di(1)+Di(2))/2.0-Di(4))**2*
     &           (Di(4)**2 + Di(4)*Di(2) - 3./4.*Di(2)**2 +
     &            Di(4)*Di(1) + 3.0/2.0*Di(1)*Di(2) - 3.0/4.0*Di(1)**2)/
     &            (12*Di(4))
         endif
!     eta=eta-pi*(r1+r2-dd)**2*(dd**2+2*dd*r2-3*r2**2 + 2*dd*r1 + 6*r1*r2 -
   3*r1**2)/(12*dd)
         SetPBC=.false.
!     omega=pi/tlayers
!-------------------------------------GG

!-------------------------------------GG
!     NEW
!     A system of 7 rods
      case (7)
         read(tmpstr,*) layers,rlayers,tlayers ! layers = x*y, rlayers = x,
             tlayers = y
         allocate ( USF_Band(rlayers,tlayers) )

         tmpstr=readline(102)
         read(tmpstr,*) Di(1),Di(2),Di(3) ! Read the additional geometry
             info
         read(tmpstr,*) rod_a,rod_b,rod_c ! Read the additional geometry
             info
! diameters D, L, cylD

! Misc settings
         write(*,*) 'Box confinement, U=U(x,y), cylD/sig=', Li(1) !
             Diameter of the full cylinder
         write(*,*) 'Box confinement, U=U(x,y), cylL/sig=', Li(3) ! Length
             of the cylinder = 2*cutoff = 10sigma
         write(*,*) 'Additional: D, L, cylD ', Di(1), Di(2), Di(3)

         do i=1,layers          ! Read in tabulated potential
            tmpstr=readline(102)
            read(tmpstr,*) j,k,temp !j==x, k==y
            USF_Band(j+1,k+1)=temp
         enddo

         rlayers=rlayers-1      ! First layer is boundary, U(r=0)
         tlayers=tlayers-1      ! First layer is boundary, U(theta=0)
         Volume=Li(1)*Li(2)*Li(3)
! Periodic boundary conditions on Z only
         SetPBC(1:2)=.false.
         SetPBC(3)=.true.

         alpha = atan(rod_b/rod_c) ! parallelogram angle
         beta = pi/2.0 - alpha

!-------------------------------------GG

      case default
         write(*,*) 'Potential not yet supported!'
         stop

      end select
      close(102)

      end subroutine readsol

ccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine readmol(filename)
      character*50 filename
      character*80 txtin
      character*50 tmp(3)
      real summoves
      integer i,j
      logical lexists

      inquire(file=filename, exist=lexists)
      if (.not. lexists) then
         write(*,*) "MODEL FILE NOT FOUND: ", filename
         stop
      end if


      open(file=filename, unit=103, status=’OLD’)
      do i=1,maxlines
         txtin=readline(103)
         read(txtin,*) tmp(1)
         tmp(1)=trim(tmp(1))
         select case (tmp(1))

         case ("POTENTIAL")
            read(txtin,*) tmp(1), tmp(2)
            tmp(2)=trim(tmp(2))
            select case (tmp(2))
            case ("LJ")
               lennjones=.true.
               do j=1,9
                  txtin=readline(103)
                  read(txtin,*) tmp(1), tmp(2)
                  tmp(1)=trim(tmp(1))
                  select case (tmp(1))
                  case("SIGMA")
                     read(tmp(2),*) sigma
                  case("EPSILON")
                     read(tmp(2),*) eps
                  case("RCUT")
                     read(tmp(2),*) rcut
                  case("NCUT")
                     read(tmp(2),*) ncut
                  case("MASS")
                     read(tmp(2),*) mass
                  case("TAILCORR")
                     read(tmp(2),*) tailcorr
                  case("SHIFT")
                     read(tmp(2),*) tmp(3)
                     if (tidy(tmp(3)).eq."ON".or.
     &                    tidy(tmp(3)).eq."TRUE") then
                        lj_shift=rcut**(-6)-rcut**(-12)
                        print *, "USING LJ SHIFT ",lj_shift
```

```
                endif
             case("END")
                goto 290
             CASE DEFAULT
                WRITE(*,*) tmp(1)," is not a valid LJ parameter!"
                stop
             end select
          enddo

       case ("IDEAL")
          do j=1,3
             txtin=readline(103)
             read(txtin,*) tmp(1), tmp(2)
             tmp(1)=trim(tmp(1))
             select case (tmp(1))
             case("EPSILON")
                read(tmp(2),*) eps
             case("MASS")
                read(tmp(2),*) mass
             case("END")
                goto 290
             CASE DEFAULT
                WRITE(*,*) tmp(1)," is not a valid IDEAL parameter!"
                stop
             end select
          enddo

       case ("HARMBOND")
          lobnd_HB=0
          hibnd_HB=9999
          harm_len=.true.
          bonds=.true.
          do j=1,5
             txtin=readline(103)
             read(txtin,*) tmp(1), tmp(2)
             tmp(1)=trim(tmp(1))
             select case (tmp(1))
             case("K")
                read(tmp(2),*) kb0_len
                kb0_len=kb0_len*0.5
             case("L")
                read(tmp(2),*) r0
             case("LOWBOUND")
                read(tmp(2),*) lobnd_HB
             case("HIGHBOUND")
                read(tmp(2),*) hibnd_HB
             case("END")
                goto 290
             CASE DEFAULT
                WRITE(*,*) tmp(1),
&                    " is not a valid HARMBOND parameter!"
                stop
             end select
          enddo

       case ("HARDBOND")
          hardbond=.true.
          bonds=.true.
          run_ini=.true.
          do j=1,5
             txtin=readline(103)
             read(txtin,*) tmp(1), tmp(2)
             tmp(1)=trim(tmp(1))
             select case (tmp(1))
             case("R")
                read(tmp(2),*) r0
                print *, r0
             case("END")
                goto 290
             CASE DEFAULT
                WRITE(*,*) trim(tmp(1)),
&                    " is not a valid HARDBOND parameter!"
                stop
             end select
          enddo

       case default
          write(*,*) tmp(2)," is not a valid potential!"
          stop
       end select

    case ("INSERTION")
       ins_bias=.true.
       rins_min=0
       DO j=1,5
          txtin=readline(103)
          read(txtin,*) tmp(1), tmp(2)
          tmp(1)=trim(tmp(1))
          select case (tmp(1))
          case("VOLUME")
             select case (tmp(2))
             case ("SPHERE")
                ins_geom=1
             case ("CUBE")
                ins_geom=3
             end select
          case("RMAX")
             read(tmp(2),*) rins_max
          case("RMIN")
             read(tmp(2),*) rins_min
          case("END")
             goto 290
          CASE DEFAULT
             WRITE(*,*) tmp(1),
&                 " is not a valid INSERTION parameter!"
             stop
          end select
       enddo
       if (rins_max.eq.999) then
          print *, "Please select a valid Rins_max!"
          stop
       endif
       if (ins_geom.eq.3 .and. rins_min.gt.0) then
          print *, "Rins_min not supported in cube!"
          stop
       endif

    case ("MOVES")
```

```
            movesassigned=.true.
            nmoves=5
            allocate( move_pct(nmoves), move_cumpct(nmoves) )
            move_pct(:)=0

            DO j=1,nmoves+2
               txtin=readline(103)
               read(txtin,*,err=299,end=299) tmp(1), tmp(2)
               tmp(1)=trim(tmp(1))
               select case (tmp(1))
               case("DISPLACEMENT")
                  read(tmp(2),*,err=299,end=299) move_pct(1)
               case("REPTATION")
                  read(tmp(2),*,err=299,end=299) move_pct(2)
               case("REGROW")
                  read(txtin,*,err=299,end=299) tmp(3),move_pct(5),
     &                 ktrials
               case("CRANKSHAFT")
                  read(txtin,*,err=299,end=299) tmp(3),move_pct(3), cs_len
                  if (cs_len.lt.2) then
                     write(*,*) "Crankshaft move needs a minimum of 2 ",
     &                    "bond-lengths to perform!"
                     stop
                  endif
               case("EXCHANGE")
                  read(tmp(2),*,err=299,end=299) move_pct(4)
               case("INSERTION")
                  read(tmp(2),*,err=299,end=299) freq_widom
               case("END")
                  goto 289
               CASE DEFAULT
                  WRITE(*,*) tmp(1),
     &                 " is not a valid move_pct parameter!"
                  stop
               end select
            enddo
         case("EOF")
            goto 291
         case default
            write(*,*) tmp(1)," is not a valid operator in ",filename
            stop
         end select

 289     continue
!! Normallize the move percentages
         summoves=sum(move_pct)
         do j=1,nmoves
            move_pct(j)=move_pct(j)/summoves
         enddo
!! Now store them as cumulative percentages for random selection in MC
   algorithm
         move_cumpct(1)=move_pct(1)
         do j=2,nmoves
            move_cumpct(j)=move_cumpct(j-1)+move_pct(j)
         enddo


 290     continue
      enddo
      goto 291
 299  continue
      write(*,*) "Read error on ", tmp(1)
      stop

 291  continue
      close(103)
      end subroutine readmol



!     ! Reads 1 line of a file at a time, ignoring comments.
!     ! Also, capitalizes all inputs
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      character*80 function readline(io)
      character*1 comment
      integer io

      comment="#"
 209  read(io,'(A80)',err=210,end=220) readline
      if (readline(1:1).eq."#") goto 209
      readline=upcase(readline)
      return

 210  write(*,*) "File read error!"
      return

 220  readline="EOF"
      return
      end function readline


!     ! TakeStr - Reads a line from a file as 1 string
ccccccccccccccccccccccccccccccccccccccccc
      CHARACTER*80 FUNCTION TAKESTR(KAN,IE,LINECOUNT)

      INTEGER LINECOUNT
      CHARACTER*80 AUX, FN*50
      INTEGER*4 KAN, IE
      SAVE AUX
      IF(IE.EQ.99)GO TO 10
 1    LINECOUNT=LINECOUNT+1
      READ(KAN,'(A80)',ERR=10,END=20)AUX
      IF(AUX(1:1).EQ.'#')GO TO 1
      TAKESTR=AUX
      IE=0
      RETURN
 10   WRITE(*,*)'!!! ERROR IN INPUT FILE ',LINECOUNT
      IF(KAN.EQ.5)THEN
         WRITE(*,*)' STANDARD INPUT '
      ELSE
         INQUIRE(UNIT=KAN,NAME=FN)
         WRITE(*,*)' FILE ',FN
      END IF
      WRITE(*,*)' IN LINE ',LINECOUNT+1
      WRITE(*,*)AUX
      STOP
 20   IF(IE.GE.0)THEN
         WRITE(*,*)'!!! END OF FILE REACHED', LINECOUNT+1
         IF(KAN.EQ.5)THEN
            WRITE(*,*)' STANDARD INPUT '
         ELSE
            INQUIRE(UNIT=KAN,NAME=FN)
            WRITE(*,*)' FILE ',FN
         END IF
c     STOP
      END IF
      TAKESTR='  '
      IE=-1
      RETURN
      END FUNCTION


      end module inputs
```

B.3 energy.f

```
!     Calculates potential energy for CHAINBUILD program
!     Organization: Subroutines calculate energy of a specific "situation"
    (e.g. U of bead i)
!     Functions calculate a type of potential (e.g. LJ, harmonic bond, etc.)

      module energy
      use inputs
      use fort_rand
      use constants
      implicit none

!     ### ENERGIES ###
!     fluid-fluid, solid-fluid, and bond (kb): averages,per set,change on
    step
      real*8 Uff,Usf,Uff_avg,Usf_avg,Ukbl_avg,Ukba_avg
      real*8 Uff_set,Usf_set,Ukba_set,Ukbl_set
      real*8 Uext,Uint,U_SET
      real*8 dUff,dUsf,Ukbl,ukba,dUkbl,dukba,dp

      contains


!     Calculate total system enegery
cccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine calc_energy(calc_esttime)
      double precision rr
      integer i,j,k
      double precision begin_e,end_e,timeleft
      double precision punit,timepunit,time_est,movesper
      logical calc_esttime

      Uff=0.0D0
      Usf=0.0d0
      Ukbl=0.0D0
!Ukba=0.0d0

      if (calc_esttime) then
         call cpu_time(begin_e)
      endif
      do I=1,N-1

         if (harm_len) Ukbl = Ukbl + U_harmbond(i,i+1) ! Harmonic Bonds

         if (lennjones) then
            DO J=I+1,N
               if (abs(j-i).gt.ncut) then ! Intra-chain cutoff
                  Uff=Uff+U_LJ(i,j) ! Lennard-Jones
               endif
            enddo
         endif

      enddo

      if (wall_pot) then        !External Potential
         do i=1,n
            Usf=Usf+U_SF(i)
         enddo
      endif

!cccccccccccccc

      if (calc_esttime) then
! Small routine to estimate time of calculation
         call cpu_time(end_e)
         if (ensemble.eq.0) then
            punit=0; movesper=0
            if (harm_len) punit=N-1
            If (lennjones) punit=punit+comb(n,2)
            if (wall_pot) punit=punit+N

            timeleft=(end_e-begin_e)
            timepunit=timeleft/punit

            if (lennjones) movesper=movesper+n
            if (harm_len) movesper=movesper+2
            if (wall_pot) movesper=movesper+1
            movesper=movesper*2 !old and new images

            time_est=((neqset+nset)*nstep*(move_pct(1)*movesper+
     &           move_pct(2)*2*movesper+move_pct(3)*cs_len*movesper)+nstep*
     &           nset*(movesper/2))*timepunit

            write(*,*) "Estimated time to complete: ",time_str(time_est)
            call sleep(1)
         else
            write(*,*) "Estimated time to complete: unknown."
         endif
      endif

      Uext=Uff+Usf
      Uint=Ukbl                 !+Ukba

      return
      end subroutine calc_energy
cccccccccccccccccccccccccccccccccccccccccccccccc

!
cccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      real*8 function U_i(n1)
      integer n1
      U_i=Unb_i(n1)+Ubonds_i(n1,n1)
      end function

!     Calculation non-bonded potential energy of n=i
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function Unb_i(n1)
      integer n1,i,j

      dUff=0.0D0; dUsf=0.0D0

! Lennard-Jones
      if (lennjones) then
         do I=1,N
            if (abs(n1-i).gt.ncut) then ! Intra-chain cutoff
               dUff=dUff+U_LJ(i,n1) ! Lennard-Jones
            endif
         enddo
      endif

!External Potential
      if (wall_pot) then
         dUsf=U_SF(N1)
      endif

      Unb_i = dUff + dUsf
      end function Unb_i
ccccccccccccccccccccccccccccccccccccccccccccccc


!     Calculate the bond energies for all bonds relevant to beads i to j
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function Ubonds_i(i_in,j_in)
      integer i_in,j_in,k,i,j
      dUkbl = 0.0
      if (.not.harm_len) return

      i = i_in
      j = j_in
      if (i_in.eq.1) i=2
      if (i_in.eq.N) j=N-1

!print *,'pre loop J=',j
      do k=i-1,j                !loop over bonds
!   print *,'in loop',k,i-1,j
         dukbl=dukbl+u_harmbond(k,k+1)
      enddo
      Ubonds_i = dukbl


      end function Ubonds_i

!     Calculation potential energy of particle at (x,y,z),
!     that is bonded to n=N, ignoring the interaction of x,y,z and ii
!     Specify ii=0 to count all LJ interactions (besides ones ignored by
   ncut).
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function U_xyz(xx,yy,zz,ii)
      integer n1,i,j,ii
      real*8 xx,yy,zz

      n1=n+1
      dUff=0.0D0
      dUsf=0.0d0
      dUkbl=0.0D0
!dUkba=0.0d0

!Assign a temporary bead
      x(n1) = xx
      y(n1) = yy
      z(n1) = zz

! Harmonic Bonds
      if (harm_len .and. n.ne.ii) dUkbl = U_harmbond(n,n1)

! Lennard-Jones
      if (lennjones) then
         do I=1,N
            if ((n1-i).gt.ncut .and. i.ne.ii) then ! Intra-chain cutoff
               if (ii.eq.1 .and. i.eq.2 .and. func_chain(1)) then ! case of
                   repatation with func. group at N=1
                  dUff=dUff+eps_start*U_LJ(i,n1) ! Lennard-Jones
               else
                  dUff=dUff+U_LJ(i,n1) ! Lennard-Jones
               end if
            endif
         enddo
      endif

!External Potential
      if (wall_pot) then
         dUsf=U_SF(n1)
      endif

      U_xyz = dUff + dUsf + dUkbl
      return
      end function U_xyz
ccccccccccccccccccccccccccccccccccccccccccccccc


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!     ! Calculate the harmonic energy between 2 beads
!     !    U = 0.5*k*(r_ij - l)^2
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function U_harmbond(n1,n2)
      integer n1, n2
      real*8 dd
      if (n1.eq.0 .or. n2.eq.0) then
         U_harmbond = 0
         return
      endif
      dd=r_ij(n1,n2)

      if (dd.gt.hibnd_hb) then
         U_harmbond = 1e38      !Outside the bound, U=inf
      elseif (dd.lt.lobnd_hb) then
         U_harmbond = 1e38      !Outside the bound, U=inf
      else
         dd=dd-r0
         U_harmbond = kb0_len*dd*dd
      endif

      return
      end function U_harmbond
ccccccccccccccccccccccccccccccccccccccccccccccc


!     ! Calculate the Lennard-Jones potential between 2 beads
!     ! If rr is greater than 0, function calculates U using that distance
!     !    U = 4*e*( (s/r)^12 - (s/r)^6)
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function U_LJ(n1,n2)
      integer n1,n2
      real*8 xr,yr,zr,rij2,irij6
      U_LJ = 0

      xr=x(n1)-x(n2)
      yr=y(n1)-y(n2)
      zr=z(n1)-z(n2)

      if (SetPBC(1)) xr=xr - Li(1)*anint(xr/Li(1))
      if (SetPBC(2)) yr=yr - Li(2)*anint(yr/Li(2))
      if (SetPBC(3)) zr=zr - Li(3)*anint(zr/Li(3))

      rij2=xr*xr + yr*yr + zr*zr
      if (rij2.lt.rcut2) then
         irij6=1.0/(rij2*rij2*rij2)
         U_LJ=4.0*(irij6*(irij6-1.0)+lj_shift)
      endif

! modification for functionalized polymers
      if (func_chain(1)) then   ! First bead is functionalized
         if (n1.eq.1 .or. n2.eq.1) then
            U_LJ = eps_start*U_LJ
         endif
      endif

      return
      end function U_LJ
ccccccccccccccccccccccccccccccccccccccccccccccc

! careful not to call this function from others that add 1 to particle array
      real*8 function USF_xyz(xn,yn,zn,isFirst)
      real*8 xn,yn,zn,xo,yo,zo
      integer N1
      logical,optional :: isFirst
      logical :: isFirst2
      if (.not. present(isFirst)) then
         isFirst2 = .false.
      else
         isFirst2 = isFirst
      endif
      if (isFirst2) then
         N1 = 1
      else
         N1 = N+1
      end if
! just in case...
      xo = x(N1)
      yo = y(N1)
      zo = z(N1)
      x(N1) = xn
      y(N1) = yn
      z(N1) = zn
      USF_xyz = U_SF(N1)
      x(N1) = xo
      y(N1) = yo
      z(N1) = zo
      end function



!       ! Calculate solid-fluid interaction
cccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function U_SF(ii)
      real*8 rw,rr,cband,xn,yn,zn,y1,y2,rn(3)
      real*8 r1,r2,t1,t2,t,p1,p2,cbr,cbth,rth,theta,tt
      integer x1,x2,ri1,ri2,ti1,ti2
      integer i,j,ii
!----------------------------------------GG
      real*8 z0, rho            ! cylindrical coordinates
      real*8 z_step, rho_step, z_lower, z_higher, t_factor
      real*8 rho_lower, rho_higher, u_factor, Ugrid
      integer z_i_lower, z_i_higher,rho_i_lower, rho_i_higher

!----------------------------------------GG
      U_SF = 0.d0

      xn = x(ii)
      yn = y(ii)
      zn = z(ii)

! hardwall - a solid replusive wall at z=0
      if (hardwall) then
         if (zn.le.0) U_SF=1e37
         return
      endif
      if (space) return         ! easy way out...

!! IMPORTANT NOTE!!
!  This program uses tabulated data for external potential.  Care must be
    taken
!  when specifying the bounds of the system.  The file with the potential
    should
!  have N+1 layers, with the first line the potential of the LEFT wall, or
    the
!  value of U(r=0) if spherical.  Each line after that is the value of the
!  potential at the UPPER bound of that particular bin.

      select case (geom)
      case (1)                  ! Spherical, U=U(r)
         rw=dsqrt(xn*xn+yn*yn+zn*zn)
         rr=rw/Ri(1)            !relative distance
         cband=layers*rr+1      !closest band (plus 1 since array starts at
             index 1)
         if (cband.gt.layers+1 .or. cband.lt.1) then
            U_SF=1e37           !Particle is outside system, U=+Inf
            return
         endif
         x1=int(cband)
         x2=x1+1
         y1=USF_Band(1,x1)
         y2=USF_Band(1,x2)
         U_SF=interpolate(cband,dble(x2),dble(x1),y2,y1)

      case (2)                  ! Spherical, U=U(r,theta)
         rw=dsqrt(xn*xn+yn*yn+zn*zn)
         if (rw.eq.0) then
            theta=0
         else
            theta=acos(zn/rw)
         endif
         rr=rw/Ri(1)            !relative distance
         cbr=rlayers*rr+1       !closest band in r
         if (cbr.gt.rlayers .or. cbr.lt.1) then
            U_SF=1e37           !Particle is outside system, U=+Inf
            return
         endif
         rth=theta/pi           !relative theta
         cbth=tlayers*rth+1     !closest band in theta
         r1=int(cbr); r2=cbr+1; t1=int(cbth); t2=t1+1 ! Reals
         ri1=int(r1); ri2=int(r2); ti1=int(t1); ti2=int(t2) ! Integers
!! Spherical Linear Interpolation (SLERP)
         p1=interpolate(cbr,r2,r1,USF_Band(ri2,ti1),
     &        USF_Band(ri1,ti1))
         p2=interpolate(cbr,r2,r1,USF_Band(ri2,ti2),
     &        USF_Band(ri1,ti2))
         tt=cbth-t1
         U_SF=sin((1-tt)*omega)*p1/sin(omega) +
     &        sin(tt*omega)*p2/sin(omega)

      case (3)                  ! Cartesian
         U_SF=0
         rn=(/xn,yn,zn/)
         do j=1,3
            rw=rn(j)
            rr=(rw+Ri(j))/Li(j) !relative distance (USF grid is not
                symmetric...)
            cband=layers*rr+1   !closest band (plus 1 since array starts at
                index 1)
            if (cband.gt.layers .or. cband.lt.1) then
               U_SF=1e37        !Particle is outside system, U=+Inf
               return
            endif
            x1=int(cband)
            x2=x1+1
            y1=USF_Band(j,x1)
            y2=USF_Band(j,x2)
            U_SF=U_SF+interpolate(cband,dble(x2),dble(x1),y2,y1)
         end do

      case (4)                  ! Cylindrical, U=U(r)
```

```
         rw=dsqrt(xn*xn+yn*yn)
         rr=rw/Ri(1)             !relative distance
         cband=layers*rr+1       !closest band (plus 1 since array starts at
             index 1)
         if (cband.gt.layers+1 .or. cband.lt.1) then
            U_SF=1e37            !Particle is outside system, U=+Inf
            return
         endif
         x1 = int(cband)
         x2 = x1+1
         y1 = USF_Band(1,x1)
         y2 = USF_Band(1,x2)
         U_SF = interpolate(cband,dble(x2),dble(x1),y2,y1)

!----------------------------------------------------GG
      case (5)                   ! Cylindrical, U=U(z,rho)
! We need only z and rho, z = z, rho = sqrt
         rho=dsqrt(xn*xn+yn*yn)
! rr=rho/Ri(1) ! I don't need relative distance! Check?!

         z_step = ((Di(1)+Di(2))/2.d0 + Di(4))/dble(rlayers)
! = z_length/dble(zlayers)
         rho_step = Di(1)/2.d0/dble(tlayers)
! = rho_length/dble(rholayers)


         z_i_lower = int((zn+Di(1)/2.d0)/z_step) + 1 !
         z_i_higher = z_i_lower + 1 !
         z_lower = (dble(z_i_lower) - 1)*z_step - Di(1)/2.d0 !
         z_higher = (dble(z_i_higher) -1)*z_step - Di(1)/2.d0 !

         t_factor = (zn - z_lower)/(z_higher - z_lower) !

         rho_i_lower = int(rho/rho_step) + 1 !
         rho_i_higher = rho_i_lower + 1 !
         rho_lower = (dble(rho_i_lower) - 1)*rho_step !
         rho_higher = (dble(rho_i_higher) - 1)*rho_step !

         u_factor = (rho - rho_lower)/(rho_higher - rho_lower) !

!write(*,*) "rho, r_l, tlayers", rho, rho_i_lower, tlayers

         if (z_i_lower .gt. rlayers .or. rho_i_lower.gt.tlayers) then
             !rlayers = z, tlayers = rho
! write(*,*) "!: ",z_i_lower,rlayers,rho_i_lower,tlayers
            U_SF=1e37            !Particle is outside system, U=+Inf
            return
         endif


         Ugrid = 0.d0
         Ugrid = Ugrid + (1.d0 - t_factor)*(1.d0 - u_factor)
     &        *USF_Band(z_i_lower,rho_i_lower)
         Ugrid = Ugrid + t_factor*(1.d0 - u_factor)
     &        *USF_Band(z_i_higher,rho_i_lower)
         Ugrid = Ugrid + t_factor*u_factor
     &        *USF_Band(z_i_higher,rho_i_higher)
         Ugrid = Ugrid + (1.d0 - t_factor)*u_factor
     &        *USF_Band(z_i_lower,rho_i_higher)
```

```
! Here
         U_SF = Ugrid           !z_l+z_i,rho_l+rho_i

!     write(*,*) "x=", xn, "y=", yn, "z=", zn, "rho=", rho
!     write(*,*) "rho_i_lower=", rho_i_lower
!     write(*,*) "z_i_lower=", z_i_lower
!     write(*,*) "rho_lower=", rho_lower, "z_lower=", z_lower
!     write(*,*) "U_sf=", U_SF
!     STOP


!---------------------------------------------------GG

      case default
         print *,"This type of interaction is not yet supported!"
         stop

      end select

      if (func_chain(1)) then   ! First bead is functionalized
         if (ii.eq.1) then
            U_SF = func_S1*U_SF
         endif
      endif


      end function U_SF
cccccccccccccccccccccccccccccccccccccccccccccccc


!     ! Returns the (closest periodic) distance between two beads
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function r_ij(n1,n2)
      integer n1,n2
      real*8 xr,yr,zr

      xr=x(n1)-x(n2)
      yr=y(n1)-y(n2)
      zr=z(n1)-z(n2)

      if (SetPBC(1)) xr=xr - Li(1)*anint(xr/Li(1))
      if (SetPBC(2)) yr=yr - Li(2)*anint(yr/Li(2))
      if (SetPBC(3)) zr=zr - Li(3)*anint(zr/Li(3))

      r_ij=dsqrt(xr*xr+yr*yr+zr*zr)
      return
      end function r_ij
ccccccccccccccccccccccccccccccccccccccccccccccc


!     ! Returns the (closest periodic) distance between two coordinates
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function r_xyz(x1,y1,z1,x2,y2,z2)
      real*8 x1,y1,z1,x2,y2,z2
      real*8 xr,yr,zr

      xr=x1-x2
      yr=y1-y2
      zr=z1-z2
      if (SetPBC(1)) xr=xr - Li(1)*anint(xr/Li(1))
      if (SetPBC(2)) yr=yr - Li(2)*anint(yr/Li(2))
      if (SetPBC(3)) zr=zr - Li(3)*anint(zr/Li(3))

      r_xyz=dsqrt(xr*xr + yr*yr + zr*zr)
      return
      end function r_xyz
ccccccccccccccccccccccccccccccccccccccccccccccc



!     ! Returns the (closest periodic) distance squared between two
    coordinates
ccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function r2_xyz(x1,y1,z1,x2,y2,z2)
      real*8 x1,y1,z1,x2,y2,z2
      real*8 xr,yr,zr

      xr=x1-x2
      yr=y1-y2
      zr=z1-z2

      if (SetPBC(1)) xr=xr + Li(1)*nint(xr/Li(1))
      if (SetPBC(2)) yr=yr + Li(2)*nint(yr/Li(2))
      if (SetPBC(3)) zr=zr + Li(3)*nint(zr/Li(3))

      r2_xyz=(xr*xr + yr*yr + zr*zr)

      return
      end function r2_xyz
ccccccccccccccccccccccccccccccccccccccccccccccc


!     ! Calculate bond angle between 3 beads
cccccccccccccccccccccccccccccccccccccccccccccccc
      real*8 function bond_angle(n1,n2,n3) result(theta)
      integer n1,n2,n3
      real*8 d12x,d12y,d12z,d23x,d23y,d23z,d12,d23
      real*8,dimension(3) :: d12xyz,d23xyz

!Calculate straight-line distances
      d12x=x(n1)-x(n2)
      d12y=y(n1)-y(n2)
      d12z=z(n1)-z(n2)
      d23x=x(n2)-x(n3)
      d23y=y(n2)-y(n3)
      d23z=z(n2)-z(n3)
      d12=sqrt(d12x*d12x+d12y*d12y+d12z*d12z)
      d23=sqrt(d23x*d23x+d23y*d23y+d23z*d23z)

      d12xyz=(/d12x,d12y,d12z/) !vector
      d23xyz=(/d23x,d23y,d23z/) !
      d12xyz=d12xyz/d12         !normalize vector
      d23xyz=d23xyz/d23         !

      theta=acos(dot_product(d12xyz,d23xyz))
      end function bond_angle
cccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      end module
```

B.4 moves.f

```
      module moves
      use inputs
      use energy
      use fort_rand
      use constants
      implicit none

!     # Counters #
!     number of moves accepted; displacement,repetition
      integer :: acc_dis,acc_rep,acc_cs,n_dis,n_rep,set,step,n_cs
      integer :: ng_ins,ng_rem,ng_ins_suc,ng_rem_suc,n_reg,n_reg_acc
      real*8 :: step_dis, cs_stp, sig, aa
!     chemical potential related
      real*8 :: mu,mu_set,mu_nid,mu_avg,mu_id,mu_ex
      real*8 :: Wext_o(MAXPART)
      real*8 Wold,Wavg
      integer n_widom
      integer MOLID              ! the last particle acted on
      real*8 XOLD,YOLD,ZOLD      ! the last particle acted on's original
         position
!     Misc.
      real*8 samp,Eins,Ravg,insprob


      contains



      subroutine initialize_system()
      integer i,j,w
      real*8 dd,xx,yy,zz,xn,yn,zn,r
      real*8 du, maxu
      logical overlap, suc

!! --- Initialization ---
!! NVT ENSEMBLE
      r=Li(1)/2.
      if (ensemble.eq.0) then
         if (n.lt.length) then
!If there are no bonds, place all particles in pore
            do while (n.lt.length)
               maxU = 100; dU = maxU + 1
               do while (dU > maxU)
                  maxU = maxU*1.5
                  call insert_into_volume(xx,yy,zz)
                  x(n+1)=xx; y(n+1)=yy; z(n+1)=zz
                  dU = Unb_i(n+1)
               end do
               n=n+1
            enddo

         elseif (length.lt.n) then !Too many beads given in coordinates
```

```
            print *,'Removing ',(n-length),
     &           ' beads from end of initial chain...'
         endif


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! GRAND CAN. ENSEMBLE
      elseif (ensemble.eq.1) then
         do i=1,MAXPART
            call gauge_exchange(suc)
         enddo

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! GAUGE
      elseif (ensemble.eq.2) then
!! Build gauge cell and place particles inside
         if (n.eq.0) then
            n=ntotal-ngtarget
            n_reg_acc=0; i=0
            if (move_pct(5).gt.0) then
               write(*,*) "Attempting to grow a chain in the pore..."
               do while (n_reg_acc.eq.0)
                  i=i+1
                  call regrow_move(ktrials,0)
                  if (i.gt.25) then
                     write(*,*) "Could not grow chain in pore, starti"//
     &                    "ng with Ng=N..."
                     Ng=ntotal
                     n=0
                     exit
                  endif
               enddo
               ng=ntotal-n
            else
               ng=ntotal
               n=0
            endif
         else
            ng=ntotal-n
         endif
         print *,'N,Ng=',n,ng
         print *,'V_gauge=',Vg
!Fail
      else
         write(*,*) 'Bad ensemble selection.'
         stop

      endif

      end subroutine initialize_system



!     Randomly swap end of chain to test insertion
cccccccccccccccccccccccccccccccccccccccccccccc
      subroutine swapends()
      real*8 :: xx,yy,zz
      integer :: i,j,k
      if (rnd().lt.0.5) then
```

```
         k=N/2
         do i=1,k
            J=N-I+1
            XX=X(I); YY=Y(I); ZZ=Z(I)
            X(I)=X(J); Y(I)=Y(J); Z(I)=Z(J)
            X(J)=XX; Y(J)=YY; Z(J)=ZZ
         enddo
      endif
      end subroutine
ccccccccccccccccccccccccccccccccccccccccccccccc



!     Exchange particles from gauge to pore
cccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine gauge_exchange(success)
      real*8 xnew,ynew,znew,unew
      real*8 xx,yy,zz,rr,dd,nv(3)
      real*8 Erem,theta,r6
      integer i,j,k
      logical success
      success=.false.
      dp=0

      if (both_ends) call swapends() !Randomly swap end of chains

      if (rnd().lt.insprob) then
!Insert....................................................
!..........................................................
         MOLID = N+1
         XOLD=1e36; YOLD=1e36; ZOLD=1e36 ! AKA, the gauge cell
         if (ng.eq.0) return    !Nothing in gauge to exchange
         ng_ins=ng_ins+1

         if (tethered(n+1)) then
            x(n+1)=x_org(n+1)
            y(n+1)=y_org(n+1)
            z(n+1)=z_org(n+1)
         else
            call insert_into_volume(x(n+1),y(n+1),z(n+1))
         endif

         if (ins_bias) then
            Eins=Unb_i(n+1)+U_harmbond(n+1,n)
         else
            Eins=Unb_i(n+1)
         endif

         if (bonds.and.Eins.lt.88.7) then
            if (N.gt.0) then
               dp=exp(-Eins/T)*Ng*a ! a=1/Vg or Vins/Vg
            else
               dp=exp(-Eins/T)*Ng*a*Volume ! a=1/Vg
            endif
         elseif (Eins.lt.88.7) then
            dp=exp(-Eins/T)*a*Ng/(n+1) !a=volume/Vg
!ELSE dp still 0
         endif
         if (iolev(1).ge.4) print *,'i',Eins,dp,a
         if (iolev(1).ge.5) print *,duff,dusf,dukbl

         if (rnd().lt.dp) then
c     if (abs(dukbl).ge.1e10) then
c     stop
c     endif
!Insertion Successful
            ng_ins_suc=ng_ins_suc+1
            success = .true.
            if (ensemble.eq.2) Ng=Ng-1
            N=N+1
!x(n)=xnew
!y(n)=ynew
!z(n)=znew
            Uff=Uff+DUFF        !! UPDATING ENERGIES
!Ukba=Ukba+dUkba
            Usf=Usf+dUsf
            if (harm_len) Ukbl=Ukbl+U_harmbond(n-1,n)
            Uint=Ukbl+Ukba
         else
            success = .false.
         endif

         return

      else
!!!!!!!!!REMOVAL!!!!!!!!!!!!!!!!!!!!!!!!
         if (n.eq.0) return     ! Nothing in pore to remove
         ng_rem=ng_rem+1

         if (bonds) then
! Remove last bead from chain
            k=n
!Calculate energy of removal
            if (ins_bias) then
               Erem=unb_i(k)+u_harmbond(k,k-1)
            else
               Erem = Unb_i(k)
            endif
         else                   !NONbonded: select a random bead.
            k=int(rnd()*n)+1
            Erem = Unb_i(k)
         endif
         MOLID=k
         XOLD=x(k); YOLD=y(k); ZOLD=z(k)
         if (tethered(k)) return

! Gauge: incremental chain
         if (bonds .and. N.gt.1) then
            dp=dexp(Erem/T)/(Ng+1)/a ! a=Vins/Vg
!Gauge: Simple fluid/Removal of last particle
! True because N = 1 = number of polymer chains
         elseif (ensemble.eq.2 .or. (bonds .and. N.eq.1)) then
            dp=dexp(Erem/T)*N/(Ng+1)/a !a=Volume/Vg
!GCMC
         elseif (ensemble.eq.1) then
            dp=dexp(Erem/T)*N/a !a=V/l^3*exp(mu/kT)
         endif
```

```
!if (iolev(1).ge.4) print *,'r',Erem,dp,dd
!if (iolev(1).ge.5) print *,duff,dusf,dukbl

         if (rnd().lt.dp) then
!Removal Successful
            ng_rem_suc=ng_rem_suc+1
            success = .true.
            if (.not.bonds) then
               do i=k,n         ! Fill in gap left by removal...
                  x(i)=x(i+1)
                  y(i)=y(i+1)
                  z(i)=z(i+1)
               enddo
            endif
            if (ensemble.eq.2) Ng=Ng+1
            N=N-1
            Uff=Uff-DUFF        !! UPDATING ENERGIES
            Ukba=Ukba-dUkba
            Usf=Usf-dUsf
            if (harm_len) Ukbl=Ukbl-U_harmbond(n,n+1)
            Uint=Ukbl+Ukba
         else
            success = .false.
         endif

      endif
      end subroutine gauge_exchange


!     Displace particles in chain (Main MC move)
cccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine displacement(success)
      integer i,j
      real*8 xnew,ynew,znew,rr,nv(3)
      real*8 Unew,Uold,duffo,dusfo,dukblo,du
      real*8 ud,vd,wd,rd,theta,a,b,c,sinth,costh,L ! rotation vars
      logical success
      success = .false.

      if (n.eq.0) return        !No particles to move.
      n_dis=n_dis+1

 1010 continue
      MOLID=rndint(N)           !! bead to displace
      XOLD=x(MOLID); YOLD=y(MOLID); ZOLD=z(MOLID)
      if (tethered(MOLID)) return !!

! hardbond case, perform a kink-jump move
      if (hardbond) then
! one end of the chain selected, generate random vector
         if (MOLID == 1) then
            call ran_nxyz( nv )
            XNEW = X(2) + nv(1)*r0
            YNEW = Y(2) + nv(2)*r0
            ZNEW = Z(2) + nv(3)*r0
         elseif (MOLID == N) then
            call ran_nxyz( nv )
            XNEW = X(N-1) + nv(1)*r0
            YNEW = Y(N-1) + nv(2)*r0
            ZNEW = Z(N-1) + nv(3)*r0
! Pivot an inner-bead
         else
! direction vector for rotation-axis
! from http://inside.mines.edu/~gmurray/ArbitraryAxisRotation/
            ud = x(MOLID+1) - x(MOLID-1)
            vd = y(MOLID+1) - y(MOLID-1)
            wd = z(MOLID+1) - z(MOLID-1)
            L = ud*ud+vd*vd+wd*wd
            rd = sqrt(L)
            ud = ud / rd        ! unit vector <u,v,w> between i-1 and i+1
            vd = vd / rd
            wd = wd / rd
            a = X(MOLID-1)
            b = Y(MOLID-1)
            c = Z(MOLID-1)
            theta = rndz()*step_dis
            costh = cos(theta)
            sinth = sin(theta)

            XNEW = ((a*(vd*vd+wd*wd)-ud*(b*vd+c*wd-ud*XOLD-vd*YOLD-wd*
     &           ZOLD))*(1-costh)+XOLD*costh+(-c*vd+b*wd-
     &           wd*YOLD+vd*ZOLD)*sinth)
            YNEW = ((b*(ud*ud+wd*wd)-vd*(a*ud+c*wd-ud*XOLD-vd*YOLD-wd*
     &           ZOLD))*(1-costh)+YOLD*costh+(c*ud-a*wd+
     &           wd*XOLD-ud*ZOLD)*sinth)
            ZNEW = ((c*(ud*ud+vd*vd)-wd*(a*ud+b*vd-ud*XOLD-vd*YOLD-wd*
     &           ZOLD))*(1-costh)+ZOLD*costh+(-b*ud+a*vd-
     &           vd*XOLD+ud*YOLD)*sinth)
         endif
! soft-bond case, perform a monomer displacement
      else
         XNEW=X(MOLID)+rndz()*step_dis !! NEW COORDINATES
         YNEW=Y(MOLID)+rndz()*step_dis !!
         ZNEW=Z(MOLID)+rndz()*step_dis !!
      end if

!! Old position energy
      Uold=Unb_i(MOLID)+Ubonds_i(MOLID,MOLID)
      duffo=duff; dusfo=dusf; dukblo=dukbl
      x(MOLID)=xnew; y(MOLID)=ynew; z(MOLID)=znew
!! New position energy
      Unew=Unb_i(MOLID)+Ubonds_i(MOLID,MOLID)
      duff=duff-duffo; dusf=dusf-dusfo; dukbl=dukbl-dukblo
      du = duff+dusf+dukbl
      DP=EXP(-du/T)
!print *, du, dp
!print *, XOLD, YOLD, ZOLD
!print *, XNEW, YNEW, ZNEW

      if (rnd().lt.dp) then
         acc_dis=acc_dis+1      !! ACCEPTED MOVES COUNTER ++
         success = .true.

         if (SetPBC(1)) x(MOLID)=x(MOLID)-Li(1)*anint(x(MOLID)/Li(1))
         if (SetPBC(2)) y(MOLID)=y(MOLID)-Li(2)*anint(y(MOLID)/Li(2))
         if (SetPBC(3)) z(MOLID)=z(MOLID)-Li(3)*anint(z(MOLID)/Li(3))

         Uff=Uff+DUFF           !! UPDATING ENERGIES
         Ukbl=Ukbl+dUkbl
         Ukba=Ukba+dUkba
         Usf=Usf+dUsf
         Uint=Ukbl+Ukba
      else
         success = .false.
         x(MOLID)=xold
         y(MOLID)=yold
         z(MOLID)=zold
      endif

      end subroutine displacement


c     Attempt a CBMC regrow
ccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine regrow_move(ks,k)
      integer ks,k,i,j,tlen
      real*8 Wold,Wnew,dp
      real*8,dimension(n-k+1) :: xn,yn,zn
      logical new_conf

      n_reg=n_reg+1
      tlen=n-k+1                ! length of regrown chain
      if (both_ends) call swapends()

! First calculation old Rosenbulth factor, then regrow a new chain.
      new_conf=.false.
      call regrow_chain(ks,k,Wold,new_conf,xn,yn,zn)
      new_conf=.true.
      call regrow_chain(ks,k,Wnew,new_conf,xn,yn,zn)

!if (isnan(wnew).or.isnan(wold)) return ! and the world is safe for another
   cycle...
! Rigorously, should NaN appear in this calculation?  Probably not..
      dp=Wnew/Wold
!if (isnan(dp)) return

!print '(F8.3,2F12.6)',dp,wnew,wold
      if (rnd().lt.dp) then     ! Accept move
         n_reg_acc=n_reg_acc+1

! First, calculation of old energies
         dukbl=0
!! Calculate the LJ energy of the unperturbed section of chain.
         if (lennjones) then
            do i=1,k-1
               do j=i+1,k
                  if (abs(j-i).gt.ncut) then
                     duff=duff+u_LJ(i,j)
                  endif
               enddo
            enddo
         endif

! Energy of old bonds
         if (harm_len) then
            do i=k+1,n
               dUKBL=dUkbl-u_harmbond(i-1,i)
            enddo
         endif
! Solid-fluid energy of old configuration of chain
         if (wall_pot) then
            do i=k+1,n
               dusf=dusf-u_sf(i)
            enddo
         endif

! Change coordinates

         do i=2,tlen
            j=i+k-1             ! on chain position
            x(j)=xn(i)
            y(j)=yn(i)
            z(j)=zn(i)
         enddo

! Now find bond energy of new configuration
         if (harm_len) then
            do i=k+1,n
               dUKBL=dUkbl+u_harmbond(i-1,i)
            enddo
         endif

         goto 919
         if (abs(duff).gt.abs(10*uff)) then
            print *, "Warning!! High UFF move accepted!"
            print *, "U_FF (OLD) = ",uff
            print *, "U_FF (NEW) = ",duff
            print *, "set,step= ", set,step
            print *, "wold,wnew= ",wold,wnew
            print *, "dp = ",dp
            print *, "n,ng= ",n,ng
            open (file="UFF_COOR.DUMP",unit=299,position='APPEND')
            write(299,*) set,step
            do i=1,N
               write(299,*) x(i),y(i),z(i)
            enddo
            close(299)
         endif
         if (abs(Usf+dUsf).gt.abs(10*usf)) then
            print *, "Warning!! High USF move accepted!"
            print *, "U_SF (OLD) = ",usf
            print *, "U_SF (NEW) = ",dusf+usf
            print *, "set,step= ", set,step
            print *, "wold,wnew= ",wold,wnew
            print *, "dp = ",dp
            print *, "n,ng= ",n,ng
            open (file="USF_COOR.DUMP",unit=299,position='APPEND')
            write(299,*) set,step
            do i=1,N
               write(299,*) x(i),y(i),z(i)
            enddo
            close(299)
         endif
 919     continue

! UPDATING ENERGIES
```

```
         Wavg=Wnew+Wavg
         Uff=DUFF
         Ukbl=Ukbl+dukbl
         Usf=Usf+dUsf
      endif

      end subroutine regrow_move


!     calculate the Rosenbluth factor of a chain
cccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine regrow_chain(ks,k,W,regrow,xrg,yrg,zrg)
!ks = number of trial segments to calculate per bead
!k = location to begin calculation
      integer i,j,k,nn,ks,tlen,im1,onc
      real*8,dimension(n-k+1) :: xrg,yrg,zrg,wext ! coordinates of regrown
          chain
      real*8,dimension(ks) :: xt,yt,zt,wtext,dulj,duext ! coordinates of
          trial segments
      real*8 xnew,ynew,znew,drr,rr,ir6,nv(3),duffo
      real*8 W,dp,cumw,ws,prefactor
      logical ready,regrow

      tlen=n-k+1                ! Length of regrown chain + 1
!allocate( xrg(tlen),yrg(tlen),zrg(tlen),wext(tlen) ) !trial chain
!allocate( wext(tlen) ) !trial chain
      wext=0.0; duff=0.d0; dusf=0.d0; W=1

      if (k.eq.0 .and. (tethered(1).or.func_chain(1))) k=1 !This means the
          first bead will never be moved if tethering is enabled, or if
          functional chains are involved
      if (k.eq.0) then
         if (ins_bias) then
            ins_bias=.false.    ! To insert into system, not bias, volume
            call insert_into_volume(xrg(1),yrg(1),zrg(1))
            ins_bias=.true.
         else
            call insert_into_volume(xrg(1),yrg(1),zrg(1))
         endif                  ! ugly, i know...
         if (wall_pot) dusf=USF_xyz(xrg(1),yrg(1),zrg(1),.true.)
             !solid-fluid only, nothing else in system
         wext(1)=ks*exp(-dusf/T)
      else
         xrg(1)=x(k); yrg(1)=y(k); zrg(1)=z(k)
      endif
      do i=2,tlen               ! Loop over TEST chain
         im1=i-1
         onc=i+k-1              ! position on chain

         do j=1,ks              ! Loop over trial segments per bead
            wtext(j)=0.d0
            dulj(j)=0.d0; duext(j)=0.d0

            if (.not.regrow .and. j.eq.1) then
               xt(j)=x(onc)
               yt(j)=y(onc)
               zt(j)=z(onc)
            else
               if (harm_len) then
```

```
                  rr=gen_bond_len()
               elseif (hardbond) then
                  rr=r0
               endif

               call ran_nxyz(nv)
               xt(j)=xrg(im1)+nv(1)*rr
               yt(j)=yrg(im1)+nv(2)*rr
               zt(j)=zrg(im1)+nv(3)*rr
               rr=sqrt((xt(j)-xrg(im1))**2+(yt(j)-yrg(im1))**2+
     &              (zt(j)-zrg(im1))**2)
!if (rr.lt.0.99999 .or. rr.gt.1.0001) then
!  write(998,*) i,j,rr,r0
!endif  ! I don't know what this is here for...
            endif

! Now calculate external boltzman weight
            if (wall_pot) duext(j)=USF_xyz(xt(j),yt(j),zt(j)) !solid-fluid
            if (lennjones) then
               do nn=1,k-1      !LJ with origial chain (k-1 since k is
                   included in loop below [n(k)=n_rg(1)])
                  if (abs(onc-nn).gt.ncut) then !apply ncut to trial chain
                     rr=r2_xyz(xt(j),yt(j),zt(j),x(nn),y(nn),z(nn))
!print '(A,2I2,f12.4)','Orr',onc,nn,sqrt(rr)
!print *,xt(j),yt(j),zt(j),x(nn),y(nn),z(nn)
                     if (rr.lt.rcut2) then
                        ir6=1.d0/(rr*rr*rr)
                        if (nn.eq.1 .and. func_chain(1)) then
                           prefactor = 4.0*eps_start
                        else
                           prefactor = 4.0
                        end if
                        dulj(j)=dulj(j)+prefactor*(ir6*(ir6 - 1.0) +
     &                        lj_shift)
                     endif
                  endif
               enddo
               do nn=1,i        !LJ in trial chains
                  if (abs(i-nn).gt.ncut) then
                     rr=r2_xyz(xt(j),yt(j),zt(j),xrg(nn),yrg(nn),zrg(nn))
!print '(A,2I2,f12.4)','RGrr',i,nn,sqrt(rr)
                     if (rr.lt.rcut2) then
                        ir6=1.d0/(rr*rr*rr)
                        dulj(j)=dulj(j)+4.0*(ir6*(ir6 - 1.0) + lj_shift)
                     endif
                  endif
               enddo
            endif
            wtext(j)=exp(-(dulj(j)+duext(j))/T) !external Rosenbluth weight
                for trial j of ks
            wext(i)=wext(i)+wtext(j)
         enddo                  !over trial segments ks
!print *,regrow,onc,i,wext(i)/ks,dulj(1)
! Now, select a trial segment (from Frenkel&Smit)
         if (regrow) then
            ws=rnd()*wext(i)
            cumw=wtext(1)
            nn=1
            do while (cumw.lt.ws)
```

```
               nn=nn+1
               cumw=cumw+wtext(nn)
            enddo
! EARLY REJECTION CRITERION (DO NOT USE THIS FEATURE)
!if (regrow.and.rnd().gt.wtext(nn)) return
! store trial move into test chain
            xrg(i)=xt(nn)
            yrg(i)=yt(nn)
            zrg(i)=zt(nn)
            duff=duff+dulj(nn)  ! Update fluid-fluid energy, in case trial
                chain is accepted
            dusf=dusf+duext(nn) ! Update solid-fluid energy, in case trial
                chain is accepted
         else
            xrg(i)=x(onc)
            yrg(i)=y(onc)
            zrg(i)=z(onc)
         endif
         W=W*(wext(i)/ks)       ! Calculate normalized Rosenbluth factor of
             the trial chain
      enddo                     !over trial chain

      end subroutine regrow_chain


!     Move particles down chain
!     Only works for a 1 chain system (which is this program, so that's ok)
ccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine reptation()
      integer i,j,k
      real*8 xn,yn,zn,nv(3)
      real*8 Uold, Unew, dp
      real*8 duff_tmp, dusf_tmp, dukbl_tmp, dukba_tmp

      if (n.lt.1) return
      n_rep=n_rep+1

      if (n.eq.1) then          ! only one particle, randomly displace in
          volume
         Uold = U_SF(1)
         call insert_into_volume(xn,yn,zn)
         Unew=USF_xyz(xn,yn,zn) ! only SF interaction remains

      else
         if (both_ends) call swapends()

!! Calculate energy of N=1
         if (harm_len) then
            dukbl_tmp=u_harmbond(1,2)
         else
            dukbl_tmp=0.0
         endif
         Uold=Unb_i(1)+dukbl_tmp
         duff_tmp=duff          ! store dU's for updating U's if accepted
         dusf_tmp=dusf
         dukbl_tmp=dukbl

         call insert_into_volume(xn,yn,zn)
         Unew=U_xyz(xn,yn,zn,1)
      end if

      dp = exp((Uold-Unew)/T)

      if (rnd().lt.dp) then
         acc_rep=acc_rep+1
!! Accepted, update coordinates
         do i=1,N-1
            x(i)=x(i+1)
            y(i)=y(i+1)
            z(i)=z(i+1)
         enddo
         x(n)=xn; y(n)=yn; z(n)=zn
! Update energies
         if (N .eq. 1) then
            Usf = Unew
         else
            Uff=Uff-duff_tmp+duff
            Usf=Usf-dusf_tmp+dusf
            if (harm_len) Ukbl=Ukbl-dukbl_tmp+dukbl
         end if
      endif

      end subroutine reptation
cccccccccccccccccccccccccccccccccccccccccccccccc



!     Perform a crankshaft move, with nb bonds being rotated.
cccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine crankshaft(nb)
      real*8,allocatable :: xn(:),yn(:),zn(:), xo(:),yo(:),zo(:)
      real*8 px, py, pz, theta, uold, unew, dp
      real*8 duff_cs,dusf_cs,dukbl_cs,dukba_cs
      real*8 duff_ol,dusf_ol,dukbl_ol,dukba_ol
      real a,b,c,u2,v2,w2,sqrt_u2v2w2,sum_u2v2w2,
     &     au,av,aw,bu,bv,bw,cu,cv,cw,ux,uy,uz,vx,vy,vz,wx,wy,wz
      integer i,j,k,nb,np

      np=nb-1                   ! number of particles we are going to move
      allocate ( xn(np),yn(np),zn(np),xo(np),yo(np),zo(np) )
      duff_cs=0; dusf_cs=0; dukbl_cs=0; dukba_cs=0
      duff_ol=0; dusf_ol=0; dukbl_ol=0; dukba_ol=0
      unew=0; uold=0; dp=0
      n_cs=n_cs+1

      k=rndint(n-nb)            ! 1st bead in crankshaft

!Orientation vector
      px = x(k+nb) - x(k)       !u
      py = y(k+nb) - y(k)       !v
      pz = z(k+nb) - z(k)       !w

!! Rotation matrices components
      a=x(k); b=y(k); c=z(k)
      u2=px*px; v2=py*py; w2=pz*pz
      sum_u2v2w2=u2+v2+w2
      sqrt_u2v2w2=sqrt(sum_u2v2w2)
      au=a*px; av=a*py; aw=a*pz
      bu=b*px; bv=b*py; bw=b*pz
      cu=c*px; cv=c*py; cw=c*pz

      theta = cs_stp*2.0*pi*(rnd()-0.5) ! random angle of rotation, from
         -pi to pi, times cs_stp

! Store old configuration's energies
      do i = 1,nb-1             !Loop over beads being rotated
         j=k+i                  !Index in positional array
         if (harm_len) dukbl_ol=dukbl_ol+u_harmbond(j-1,j)
         uold=uold+unb_i(j)
         duff_ol=duff_ol+duff
         dusf_ol=dusf_ol+dusf
      enddo
      if (harm_len) dukbl_ol=dukbl_ol+u_harmbond(j,j+1) !count last bond,
         not counted in loop
      uold=uold+dukbl_ol

! Store old configuration's coordinates
      do i = 1,nb-1
         j=k+i
         xo(i)=x(j)
         yo(i)=y(j)
         zo(i)=z(j)
      enddo

! Calculate new configuration's coordinates
      do i = 1,nb-1             ! loop over points to be rotated.
         j=k+i                  ! index of point being tried.

! More rotation matrices components
         ux=px*x(j); uy=px*y(j); uz=px*z(j)
         vx=py*x(j); vy=py*y(j); vz=py*y(j)
         wx=pz*x(j); wy=pz*y(j); wz=pz*z(j)

! Rotated coordinates
         xn(i) = (a*(v2+w2)+px*(-bv-cw+ux+vy+wz)+(-a*(v2+w2)+px*(bv+cw-
     &        vy-wz)+(v2+w2)*x(j))*cos(theta)+sqrt_u2v2w2*(-cv+bw-wy+vz)*
     &        sin(theta)) / (sum_u2v2w2)
         yn(i) = (b*(u2+w2)+py*(-au-cw+ux+vy+wz)+(-b*(u2+w2)+py*(au+cw-
     &        ux-wz)+(u2+w2)*y(j))*cos(theta)+sqrt_u2v2w2*(cu-aw+wx-uz)*
     &        sin(theta)) / (sum_u2v2w2)
         zn(i) = (c*(u2+v2)+pz*(-au-bv+ux+vy+wz)+(-c*(u2+v2)+pz*(au+bv-
     &        ux-vy)+(u2+v2)*z(j))*cos(theta)+sqrt_u2v2w2*(-bu+av-vx+uy)*
     &        sin(theta)) / (sum_u2v2w2)

         x(j)=xn(i)             ! store new coordinates into position array
         y(j)=yn(i)             !
         z(j)=zn(i)             !
      enddo

!Calculate new configuration's energy
      do i = 1,nb-1
         j=k+i
         if (harm_len) dukbl_cs = dukbl_cs+u_harmbond(j-1,j)
         unew=unew+unb_i(j)
         duff_cs=duff_cs+duff
         dusf_cs=dusf_cs+dusf
      enddo
      if (harm_len) then
         dukbl_cs=dukbl_cs+u_harmbond(j,j+1) !count last bond, not counted
             in loop
         unew=unew+dukbl_cs
      endif

! Acceptance probability
      dp=exp((uold-unew)/T)
!print *,'dp',dp,unew,uold
!print *,'dukbl_cs,dukbl_ol',dukbl_cs,dukbl_ol
      if (rnd().lt.dp) then
! Success, update counter and energies
         acc_cs=acc_cs+1
         uff=uff-duff_ol+duff_cs
         usf=usf-dusf_ol+dusf_cs
         if (harm_len) ukbl=ukbl-dukbl_ol+dukbl_cs
         if (harm_ang) ukba=ukba-dukba_ol+dukba_cs
      else
! Fail, restore original coordinates
         do i=1,nb-1
            j=k+i
            x(j)=xo(i)
            y(j)=yo(i)
            z(j)=zo(i)
         enddo
      endif

      deallocate( xn,yn,zn, xo,yo,zo )
      endsubroutine crankshaft
ccccccccccccccccccccccccccccccccccccccccccccccc



!     Trial insertion to measure chemical potential
ccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine insert()
      real*8 :: xnew,ynew,znew

      if (both_ends) call swapends()
      n_widom=n_widom+1

      call insert_into_volume(x(n+1),y(n+1),z(n+1))
!write(999,*) x(n+1),y(n+1),z(n+1)
!! Calculate insertion energy
      if (ins_bias) then        ! If insertion bias is on, measure total
          energy
         Eins = Unb_i(n+1)+U_harmbond(n+1,n)
      else                      ! If insertion bias is off, measure
          non-bonded energy
         Eins = Unb_i(n+1)
      endif
      mu_nid=exp(-Eins/T)+mu_nid

      end subroutine insert
ccccccccccccccccccccccccccccccccccccccccccccccc


      function gen_bond_len() result (rr)
      real*8 rr
      if (harm_len) then
         ready=.false.
         do while (.not.ready)
            rr=rndgauss(sig,r0)
            if (rnd().le.rr*rr/aa .and.
     &           (rr.lt.hibnd_hb.and.rr.gt.lobnd_hb)) then
               ready=.true.
            endif
         enddo
      elseif (hardbond) then
         rr=r0
      endif
      end function



      subroutine insert_into_volume(xnew,ynew,znew)
      real*8 xnew,ynew,znew,rr,nv(3)
!------------------GG
      real*8 znew2, rr2
      znew2 = 0.d0
      rr2=1e30                   ! distance in second sphere coord
!------------------GG
      rr=1e30

!! CASE: Solid Bonds.  Generate random coordinate on sphere face
      if (hardbond) then
         if (n.gt.0) then
            call ran_nxyz(nv)
            xnew = x(n)+r0*nv(1)
            ynew = y(n)+r0*nv(2)
            znew = z(n)+r0*nv(3)
            return

         elseif (n.eq.0) then   ! No chain, insert into system
            if (geom.le.2) then ! sphere
               do while (rr.gt.Li(1)/2.)
                  xnew=(2*rnd()-1.)*Li(1)/2.
                  ynew=(2*rnd()-1.)*Li(1)/2.
                  znew=(2*rnd()-1.)*Li(1)/2.
                  rr=sqrt(xnew*xnew+ynew*ynew+znew*znew)
               enddo
            elseif (geom.eq.3) then ! bulk
               xnew=(2*rnd()-1.)*Li(1)/2.
               ynew=(2*rnd()-1.)*Li(2)/2.
               znew=(2*rnd()-1.)*Li(3)/2.
            elseif (geom.eq.4) then ! cyl
               do while (rr.gt.Li(1)/2.)
                  xnew=(2*rnd()-1.)*Li(1)/2.
                  ynew=(2*rnd()-1.)*Li(2)/2.
                  znew=(2*rnd()-1.)*Li(3)/2.
                  rr = sqrt(xnew*xnew+ynew*ynew)
               enddo
            endif
            return
         endif
      endif

!! Insertion particle into random volume center at n=N
!!  Using the specified insertion volume in the model file.
!!  Not the prettiest implimentation..
      if (ins_bias.and.n.gt.0) then
 401     continue
         xnew = (2.0*rnd()-1.0)*rins_max+x(n)
         ynew = (2.0*rnd()-1.0)*rins_max+y(n)
         znew = (2.0*rnd()-1.0)*rins_max+z(n)
         rr = sqrt((xnew-x(n))**2+(ynew-y(n))**2+(znew-z(n))**2)
         if (ins_geom.eq.1 .and. rr.gt.rins_max) goto 401 !new particle is
            outside pore.
         if (ins_geom.eq.1 .and. rr.lt.rins_min) goto 401 !new particle not
            in annulus.

      elseif (bonds .and. n.gt.0) then ! No bias
         if (harm_len) then
            rr=gen_bond_len()
         elseif (hardbond) then
            rr=r0
         endif

         call ran_nxyz(nv)
         xnew=x(n)+rr*nv(1)
         ynew=y(n)+rr*nv(2)
         znew=z(n)+rr*nv(3)


      else                          ! No chain, insert into system
         if (geom.le.2) then        ! sphere
            do while (rr.gt.Li(1)/2.)
               xnew=(2*rnd()-1.)*Li(1)/2.
               ynew=(2*rnd()-1.)*Li(1)/2.
               znew=(2*rnd()-1.)*Li(1)/2.
               rr=sqrt(xnew*xnew+ynew*ynew+znew*znew)
            enddo

         elseif (geom.eq.3) then ! bulk
            xnew=(2*rnd()-1.)*Li(1)/2.
            ynew=(2*rnd()-1.)*Li(2)/2.
            znew=(2*rnd()-1.)*Li(3)/2.

         elseif (geom.eq.4) then ! cyl
            do while (rr.gt.Li(1)/2.)
               xnew=(2*rnd()-1.)*Li(1)/2.
               ynew=(2*rnd()-1.)*Li(2)/2.
               znew=(2*rnd()-1.)*Li(3)/2.
               rr = sqrt(xnew*xnew+ynew*ynew)
            enddo
!--------------------------------------GG

         elseif (geom .eq. 5) then ! cylinder, U=U(r,z)
            do while ( (rr.gt.Di(1)/2.).and.(rr2.gt.Di(2)/2.))
               xnew=(2*rnd()-1.)*Di(1)/2.
               ynew=(2*rnd()-1.)*Di(1)/2.
               znew=rnd()*Li(3) - Di(1)/2.
               znew2 = znew - Di(4)
               rr=sqrt(xnew*xnew+ynew*ynew+znew*znew)
               rr2=sqrt(xnew*xnew+ynew*ynew+znew2*znew2)
            enddo
```

```
!------------------------------------GG

         elseif (geom .eq. 7) then ! periodic parallelogram
! quick n dirty, not efficient at all...
            in_box = .false.
            do while ( .not. in_box )
               xnew = (2*rnd()-1.) * Li(1)/2.
               ynew = (2*rnd()-1.) * Li(2)/2.
               znew = (2*rnd()-1.) * Li(3)/2.

               xscaled = xnew + Li(1)/2.0
               yscaled = ynew + Li(2)/2.0

               if ( yscaled > rod_a ) then
                  height = Li(2) - xscaled*tan(alpha)
                  if (yscaled < height) in_box = .true.
               elseif ( yscaled < rod_b ) then
                  height = rod_b - xscaled/tan(beta)
                  if (yscaled > height) in_box = .true.
               else
                  in_box = .true.
               end if
            enddo

         endif


      endif

      endif
      end subroutine insert_into_volume

      end module moves
```

B.5 sample.f

```
!!!   SAMPLE MODULE !!!!
!     Includes subroutines for calculation/ensemble sampling.
      module sample
      use inputs
      use moves
      use constants
      use energy
      use fort_rand

      implicit none
      DOUBLE PRECISION,dimension(MAXPART) :: nh_ng
      DOUBLE PRECISION,dimension(MAXPART-1) :: bl
      DOUBLE PRECISION,dimension(:),allocatable :: nh_bl,h_bl
      DOUBLE PRECISION,allocatable :: nh_den(:,:),h_den(:),ends_den(:)
      DOUBLE PRECISION,dimension(:),allocatable :: nh_ba,h_ba
      real*8 :: RMS,tort,rg2,rg
      real*8 :: rg2_avg,rg_avg,tort_avg,RMS_avg
      real*8 :: bl_set,bl_avg, ba_set,ba_avg
      real*8 :: chlen, blmin, blmax
      real*8 :: a_stiff, contour_len
      real*8 RMS_1N,RMS_1N_avg
!integer counter1
```

```
      contains

!! Sampling related to bonds and molecular positions
!!  Includes RMS, Radius of Gyration, Bond Lengths, etc.
      subroutine samplebonds(output)
      logical output
      integer bins,i,j
      real histmax, histmin, histlen, dx
      real*8 x12,y12,z12,rk
      real*8 xm,ym,zm,xr,yr,zr

!Clear variables
      bl_set=0; RMS=0; rg2=0
      chlen=0; tort=0
      xm=0; ym=0; zm=0;
      if (set.eq.neqset+1) then
!clear averages
         rg2_avg=0; bl_avg=0; tort_avg=0
         RMS_1N_avg=0; RMS_avg=0
         allocate( nh_bl(bl_bins),h_bl(bl_bins) )
         nh_bl(:)=0; bl(:)=0
         open(file=posfile,unit=77)
         write(77,'(A8,A15,A15,A15,A15,A15,A15,A15)')
     &        '#(1)set,','(2)bond len,','(3)chain len,','(4)tort,',
     &        '(5)<R_mean>,','(6)sqrt<S2>,','(7)sqrt<R2>,', '(8)<S2>/<R2>'
         close(77)
      endif

      if (n.lt.2) return        ! Not enough bonds to sample!

      if (.not.output) then
         do i=1,N               !Loop over all particles
            if (i.lt.N) then
               x12=x(i)-x(i+1)
               y12=y(i)-y(i+1)
               z12=z(i)-z(i+1)
               bl(i)=sqrt(x12*x12+y12*y12+z12*z12)
               bl_set=bl_set+bl(i) !Calculate mean bond
               chlen=bl(i)+chlen !Total chain length
            endif

! Calculate mean position
            xm= xm+x(i)
            ym= ym+y(i)
            zm= zm+z(i)
         enddo

         if (N.gt.0) then
            xm= xm/N
            ym= ym/N
            zm= zm/N
            rms=xm*xm+ym*ym+zm*zm
         else
            rms=0
         endif
         RMS_avg= RMS+RMS_avg

         if (recordcom) then    !append the center com to file
```

```
            open(file=comfile,unit=27,position='append')
            write(27,*) xm,ym,zm
            close(27)
         endif

! Initiallize the bond length histogram
         if (set.eq.neqset+1) then
            histmax=maxval(bl)
            histmin=minval(bl)
            histlen=histmax-histmin
            dx=2.0*histlen/float(bl_bins)
            histmin=histmin-0.25*histlen
!NOTE: the initial bins here are the range of bond lengths of the first
    averaging set,
! times 2.  So the distribution will have a length of L, with 0.5L of 0
    weight on each side.
! If the system fluxuates past this, some will not be counted.  The program
    will warn a user
! when this happens, but it will not stop and distribution will not reflect
    those missed points.
            do i=1,bl_bins
               h_bl(i)=dx*i+histmin !assign bin 'names'
            enddo
         endif

! Sample bond lengths
         bl_set= bl_set/dble(N-1)
         bl_avg= bl_set+bl_avg
         do i=1,N-1             !Loop over bonds
            do j=1,bl_bins      !Loop over bins
               if (bl(i).le.h_bl(j)) then
                  nh_bl(j)=nh_bl(j)+1
                  goto 151      !bin placement successful, go to next bond
               endif
            enddo
 151        continue
         enddo

! Calculation of Tortuosity
         if (N.gt.0) then
            xr=x(1)-x(N)
            yr=y(1)-y(N)
            zr=z(1)-z(N)
            RMS_1N= xr*xr+yr*yr+zr*zr
            tort= chlen/sqrt(RMS_1N)
         else
            tort=0
            RMS_1N=0
         endif
         tort_avg= tort+tort_avg
         RMS_1N_avg=RMS_1N+RMS_1N_avg

! Calculate radius of gyration
         do i=1,N
            xr=x(i)-xm
            yr=y(i)-ym
            zr=z(i)-zm
            rg2= rg2+(xr*xr+yr*yr+zr*zr)
         enddo
         if (N.gt.0) then
            rg2=rg2/dble(N)
         else
            rg2=0
         endif
         rg2_avg= rg2+rg2_avg
         rg_avg=rg_avg+sqrt(rg2)

         if (recordpos) then
            open(file=posfile,unit=77,position='append')
            open(file=trim(jobname)//'.rg',unit=78,position='append')
            write(77,7701) set,bl_set,chlen,tort,sqrt(RMS),sqrt(rg2),
     &                 sqrt(RMS_1N),rg2/RMS_1N
!For a random coil, ^  these  ^ should be equal
            write(78,*) set,N,sqrt(rg2),rg2

 7701       format(I8,F15.3,F15.3,F15.3,F15.3,F15.3,F15.3,F15.4)
            close(77)
            close(78)
         endif

!Reduce and write variables
      else

! Reduce averages
         bl_avg=bl_avg/nset
         RMS_avg=sqrt(RMS_avg/nset)
         rg2_avg=rg2_avg/nset
         rg_avg=rg_avg/nset
         tort_avg=tort_avg/nset
         RMS_1N_avg=sqrt(RMS_1N_avg/nset)

! Reduce bond length histogram and write to disk
         open(file=bondlfile, unit=25)
         do i=1,bl_bins
            nh_bl(i)=nh_bl(i)/dble(nset)
            write(25, *) h_bl(i),nh_bl(i)
         enddo
         close(25)

      endif
      end subroutine samplebonds

!!!!!!!!!!!
!!!!!!!!!!!
      subroutine sample_density(output)
      integer i,j,k,g
      real*8 ii,ii1
      real*8 rr,rw,vs,rdist,zdist,theta,rz,zr,rx,xr
      real*8 r1,r2,h1,h2,h,rr1,rr2
      logical output

      if (.not.output) then

         if (set.eq.neqset+1) then
            if (dens_type.eq.1) then ! R
               allocate( nh_den(1,dens_bins),h_den(dens_bins),
     &                    ends_den(dens_bins) )
               nh_den=0; h_den=0; ends_den=0
            elseif (dens_type.eq.2) then ! RZ
               allocate( nh_den(rad_bins,ax_bins) )
               nh_den=0
            elseif (dens_type.eq.3) then !XYZ
               allocate( nh_den(3,dens_bins),h_den(dens_bins),
     &              ends_den(dens_bins) )
               nh_den=0; h_den=0; ends_den=0
            elseif (dens_type.eq.4) then ! XZ,y=0
               allocate( nh_den(dens_bins,dens_bins) )
               nh_den=0
            endif
         ENDIF

         if (n.le.0) return
         if (dens_type.eq.1) then ! rho(r)
            do j=1,N
               rr=sqrt(x(j)*x(j)+y(j)*y(j)+z(j)*z(j))
! Now "bin" the distances into a histogram
               i = int(rr*dens_bins)+1
               if (i.le.dens_bins) then ! don't count particles out of box
                  nh_den(1,i)=nh_den(1,i)+1
               endif
            enddo

         else if (dens_type.eq.2) then ! rho(r,theta)
            do j=1,N
! First calculate distances respective of system geometry
               rr=sqrt(x(j)*x(j)+y(j)*y(j)+z(j)*z(j))
               theta=acos(z(j)/rr)
               rz=rr*sin(theta)/Ri(1)
               rr=rr/Ri(1)
! Now "bin" the distances into a histogram
               i = int(rz*rad_bins)+1
               zr=(z(j)+Ri(1))/Li(1)
               k = int(zr*ax_bins)+1 ! z+R so bins are from 0-->D rather
!print *, rz,zr
               if (i.le.rad_bins.and.k.le.ax_bins) then ! than -R --> R
                  nh_den(i,k)=nh_den(i,k)+1
               endif
            enddo

         else if (dens_type.eq.3) then ! rho(x,y,z)  (NOT TESTED)
            do j=1,N
               do g=1,3
                  select case (g)
                  case(1)
                     rr=(x(j)+Ri(g))/Li(g)
                  case(2)
                     rr=(y(j)+Ri(g))/Li(g)
                  case(3)
                     rr=(z(j)+Ri(g))/Li(g)
                  end select
! Now "bin" the distances into a histogram
                  i = int(rr*dens_bins)+1
                  if (i.le.dens_bins) then ! don't count particles out of
                      box
                     nh_den(g,i)=nh_den(g,i)+1
                  endif
               enddo
            enddo

         else if (dens_type.eq.4) then ! rho(x,y=0,z)
            do j=1,N
               if (abs(y(j)).lt.xz_dy) then ! see if particle is in XZ plane
                  rx=(x(j)+Ri(1))/Li(1) ! relative distance in X
                  rz=(z(j)+Ri(1))/Li(1) ! relative distance in Z
                  i=int(rx*dens_bins)+1
                  k=int(rz*dens_bins)+1
                  if (i.le.dens_bins.and.k.le.dens_bins) then
                     nh_den(i,k)=nh_den(i,k)+1
                  endif
               endif
            enddo

         endif

      else                      !! --- Output --- !!

         open(file=densfile,unit=230)
!! Reduce the hystograms and write to disk
         if (dens_type.eq.1) then
            rw=Ri(1)
            do i=1,dens_bins
               ii = float(i)/dens_bins*rw
               ii1 = float(i-1)/dens_bins*rw
               vs = 4*pi*(ii**3 - ii1**3)/3 ! Volume of slice
               nh_den(1,i) = nh_den(1,i)/nset/vs
               ends_den(i) = ends_den(i)/nset
            enddo
            do i=1,dens_bins
               h_den(i)=float(i)/dens_bins !assign the profile's index,
                   relative distance
               write(230,*) h_den(i),nh_den(1,i),ends_den(i)
            enddo

         elseif (dens_type.eq.2) then
            rw=Ri(1)
            do i=1,rad_bins
               rr2=dble(i)/rad_bins*rw
               rr1=dble(i-1)/rad_bins*rw
               vs=pi*(rr2**2-rr1**2)*(Li(1)/ax_bins) ! pi*r^2*dz
               do j=1,ax_bins
                  nh_den(i,j) = nh_den(i,j)/nset/vs
               enddo
            enddo
            do i=1,ax_bins
               zdist=dble(i)/ax_bins*Li(1)
               do j=1,rad_bins
                  rdist=dble(j)/rad_bins*Ri(1)
                  write(230,*) rdist,zdist,nh_den(j,i)
               enddo
               write(230,*)
            enddo

         elseif (dens_type.eq.3) then
            do g=1,3
               rw=Ri(g)
               do i=1,dens_bins
```

```
                ii = float(i)/dens_bins*rw
                ii1 = float(i-1)/dens_bins*rw
                if (sphere) then
                   h1=rw-ii    !! NOT TESTED !!
                   h2=rw-ii1
                   h=h1-h2
                   r1=sqrt(rw**2-h1**2)
                   r2=sqrt(rw**2-h1**2)
                   vs=pi/6*(3*r1**2+3*r2**2+h**2)*h ! Volume of a slice
                       of a sphere in the x,y, or z plane
                else
                   vs = (ii-ii1)*product(Li)/Li(g) ! Volume of a slice of
                       the x,y or z plane
                end if
                nh_den(g,i) = nh_den(g,i)/nset/vs
                ends_den(i) = ends_den(i)/nset
             enddo
          enddo
          do i=1,dens_bins
             h_den(i)=float(i)/dens_bins !assign the profile's index,
                 relative distance
             write(230,*) h_den(i),nh_den(1,i),nh_den(2,i),
     &              nh_den(3,i),ends_den(i)
          enddo

       else if (dens_type.eq.4) then
          vs=pi*Ri(1)**2*(2.0*xz_dy)
          do i=1,dens_bins
             rx=dble(i)/dens_bins*li(1)
             do j=1,dens_bins
                rz=dble(j)/dens_bins*li(1)
                nh_den(i,j) = nh_den(i,j)/nset/vs
                write(230,*) rx,rz,nh_den(i,j)
             enddo
             write(230,*)
          enddo

       endif
       close(230)

    endif

    end subroutine sample_density


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    function radius_gryation() result (radgry)
    real*8 radgry,xm,ym,zm,xr,yr,zr
    integer i

    if (N.lt.1) return
    xm=0; ym=0; zm=0

    do i=1,N                 !Loop over all particles
! Calculate mean position
       xm= xm+x(i)
       ym= ym+y(i)
       zm= zm+z(i)
```

```
      enddo

      xm= xm/N
      ym= ym/N
      zm= zm/N

! Calculate radius of gryation
      radgry=0
      do i=1,N
         xr=x(i)-xm
         yr=y(i)-ym
         zr=z(i)-zm
         radgry= radgry+(xr*xr+yr*yr+zr*zr)
      enddo
      radgry=radgry/dble(N)
      radgry=sqrt(radgry)

      end function radius_gryation




!! SAMPLE AND REDUCE GAUGE CELL HISTOGRAM
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

      subroutine samplegauge(output)
      logical output
      integer i
      real*8 P_ng,P_ng1         !gauge histogram probablities
      real*8 mu_dist            !chem. potential from gauge histogram

      if (.not.output) then     !no ouput, so sample instead.
!counter1=counter1+1
! Change this so it calculation can be resumed if simulation stops
         if (ensemble.eq.1) nh_ng(n+1)=nh_ng(n+1)+1.0 !Note the indice is
             shifted since Fortran arrays must start at 1
         if (ensemble.eq.2) nh_ng(ng+1)=nh_ng(ng+1)+1.0 !Note the indice is
             shifted since Fortran arrays must start at 1


      else                       !Reduce N_G histogram and write to disk

!print *, counter1
         open(file=histfile, unit=26)
         if (ensemble.ge.1) then
            if (ensemble.eq.2) write(26,’(A,ES12.4)’)’# Gauge Volume = ’,vg
            do i=0,MAXPART-1

               if (i.gt.0) then
                  P_ng1=P_ng    !P(i-1)
               else
                  P_ng1=0
               endif
               P_ng=nh_ng(i+1)/dble(nstep*nset) !P(i)
               if (ensemble.eq.2) then
                  if (P_ng.eq.0 .or. P_ng1.eq.0) then
                     mu_dist=0
                  else
                     if (bonds) then
```

```
                        mu_dist = T*log(dble(i)/Vg*P_ng/P_ng1)
                     else
                        mu_dist = T*log(dble(i)/Vg*P_ng/P_ng1)+mu_id
                     endif
                  endif
               else
                  mu_dist=0
               endif
               if (P_ng.gt.1e-8) write(26,*) i,P_ng,mu_dist,Ntotal-i
               if (Ntotal-1.le.0) goto 775

            enddo
         endif
 775     close(26)

      endif
      end subroutine samplegauge
!ccccccccccccccccccccccccccccccccccccccccc

      subroutine sampleangles(output)
      logical output
      integer bins,i,j
      real histmax, histmin, histlen, dx
      REAL ba(MAXPART-2)


!Clear variables
      ba_set=0;
      if (set.eq.neqset+1) then
!clear averages
         ba_avg=0
         allocate( nh_ba(ba_bins),h_ba(ba_bins) )
         nh_ba(:)=0; ba(:)=0
      endif

      if (n.lt.3) return        ! Less than 1 bond angle in the system

      if (.not.output) then
         do i=1,N-2             !Loop over all bond angles
            ba(i)=bond_angle(i,i+1,i+2)
         enddo

! Initiallize the bond angle histogram
         if (set.eq.neqset+1) then
            histmax=maxval(ba)
            histmin=minval(ba)
            histlen=histmax-histmin
            dx=2*histlen/float(ba_bins)
            histmin=histmin-0.25*histlen
!NOTE: the initial bins here are the range of bond lengths of the first
   averaging set,
! times 2.  So the distribution will have a length of L, with 0.5L of 0
   weight on each side.
! If the system fluxuations past this, some will not be counted.  The
   program will warn a user
! when this happens, but it will not stop and distribution will not reflect
   those missed points.
            do i=1,ba_bins
               h_ba(i)=dx*i+histmin !assign bin 'names'
            enddo
         endif

! Sample bond angles
         ba_set= ba_set/dble(N-2)
         ba_avg= ba_set+ba_avg
         do i=1,N-2             !Loop over angles
            do j=1,ba_bins      !Loop over bins
               if (ba(i).le.h_ba(j)) then
                  nh_ba(j)=nh_ba(j)+1
                  goto 152      !bin placement successful, go to next bond
               endif
            enddo
 152        continue
         enddo

!Reduce and write variables
      else

! Reduce averages
         ba_avg=ba_avg/nset

! Reduce bond length histogram and write to disk
         open(file=bondafile, unit=26)
         do i=1,ba_bins
            nh_ba(i)=nh_ba(i)/dble(nset)
            write(26, *) h_ba(i),nh_ba(i)
         enddo
         close(26)

      endif
      end subroutine sampleangles



      subroutine sample_gr(output)
      real*8 constant,n_avg,r_low,r_high,n_id,const
      integer bin,i,j
      logical output

      if (.not.output) then     !no ouput, so sample instead.
         do i=1,N-1
            do j=i+1,N
               bin=int(r_ij(i,j)/gr_delr)+1
               if (bin.le.gr_maxbin) then
                  gr_hist(bin)=gr_hist(bin)+2
               endif
            enddo
         enddo

      else                      !Reduce g(r) histogram and write to disk
         constant=4.0*pi/3.0
         n_avg=rho*Volume
         open(file=trim(jobname)//'.gr',unit=79)
         do bin=1,gr_maxbin
            r_low=float(bin-1)*gr_delr
            r_high=r_low+gr_delr
            n_id=constant*(r_high**3-r_low**3)
            gr_hist(bin)=gr_hist(bin)/float(nset)/n_avg/n_id
            write(79,*) gr_delr*bin, gr_hist(bin)
         enddo
         close(79)

      endif

      end subroutine sample_gr


      end module sample
```

B.6 output.f

```
      module outputs
      use inputs
      use moves
      use energy
      use sample
      implicit none

      character*1 xmol(4)
      logical step_warning

      contains

!     Write system values to terminal and disk at various levels
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine set_output(termlev,disklev)
      integer :: termlev,disklev,pct,i
      real dis_pct, rep_pct, cs_pct,cbr_pct !percent of total moves
      real dpct, rpct, cpct, ipct, rempct,cbrpct,expct !acceptance
          percentage
      real mv_pct(nmoves)
!logical AnyPBC

      open(file=enefile,unit=203,status='NEW',ERR=333)
      write(203,'(A1,A6,6A14)')
     &     "#","SETS","N_SET","U_SET","U_FF","U_SF","U_KBL","U_KBA"
      GOTO 334
 333  continue
      OPEN(FILE=enefile,unit=203,position='APPEND')
 334  continue
      open(file=mufile,unit=204,position='APPEND')
      open(file=logfile,unit=202,position='APPEND')

      dpct=0; rpct=0; ipct=0; rempct=0; cpct=0;cbrpct=0

      pct = 100*set/(nset+neqset)
      if (n_dis.gt.0) dpct = float(acc_dis)/float(n_dis)
      dis_pct = float(n_dis)/float(nstep)
      if (n_rep.gt.0) rpct = float(acc_rep)/float(n_rep)
      rep_pct = float(n_rep)/float(nstep)
      if (n_cs.gt.0) cpct = float(acc_cs)/float(n_cs)
      cs_pct = float(n_cs)/float(nstep)
      if (n_reg.gt.0) cbrpct=float(n_reg_acc)/float(n_reg)
      cbr_pct = float(n_reg)/float(nstep)
      if ((ng_ins+ng_rem).gt.0) then
         expct = float(ng_ins_suc+ng_rem_suc)/float(ng_ins+ng_rem)
```

```
endif


mv_pct(1)=dpct
mv_pct(2)=rpct
mv_pct(3)=cpct
mv_pct(4)=expct
mv_pct(5)=cbrpct

if (termlev.ge.1) then
   write(*,501) set,nset+neqset,pct
endif
if (disklev.ge.1) then
   write(202,501) set,nset+neqset,pct
endif
if (termlev.ge.2) then
   write(*,'(A8)',advance='no') "Mov Acc:"
   do i=1,nmoves
      if (move_pct(i).gt.0) then
         write(*,'(A3)',advance='no') mv_short(i)
         if (move_cumpct(i).lt.1) then
            write(*,'(A1)',advance='no') ","
         endif
      endif
   enddo
   write(*,'(A2)',advance='no') ": "
   do i=1,nmoves
      if (move_pct(i).gt.0) then
         write(*,'(ES8.2)',advance='no') mv_pct(i)
         if (move_cumpct(i).lt.1) then
            write(*,'(A1)',advance='no') ","
         endif
      endif
   enddo
   write(*,'(A1)') "."
endif
if (disklev.ge.2) then
   write(202,'(A8)',advance='no') "Mov Acc:"
   do i=1,nmoves
      if (move_pct(i).gt.0) then
         write(202,'(A3)',advance='no') mv_short(i)
         if (move_cumpct(i).lt.1) then
            write(202,'(A1)',advance='no') ","
         endif
      endif
   enddo
   write(202,'(A2)',advance='no') ": "
   do i=1,nmoves
      if (move_pct(i).gt.0) then
         write(202,'(ES8.2)',advance='no') mv_pct(i)
         if (move_cumpct(i).lt.1) then
            write(202,'(A1)',advance='no') ","
         endif
      endif
   enddo
   write(202,'(A1)') "."
endif
if (termlev.ge.3) then
   if (ensemble.eq.1) then
```

```
           write(*,505) Uff_set,Usf_set,N_set
        else
           write(*,503) Uff_set,Ukbl_set+Ukba_set,Usf_set,mu_set
        endif
     endif
     if (disklev.ge.3) then
        if (ensemble.eq.1) then
           write(202,505) Uff_set,Usf_set,rho_set*Volume
        else
           write(202,503) Uff_set,Ukbl_set+Ukba_set,Usf_set,mu_set
        endif
     endif
     if (recordene.and.set.gt.1) then
        write(203,'(I7,E14.5,6E14.5)')
    &        set,N_set,U_SET,Uff_set,Usf_set,Ukbl_set,Ukba_set
        write(204,*) set*(step-1),mu_set
     endif
     if (disklev.ge.5) then
     endif

     if (ensemble.eq.1) then
        if (termlev.ge.3) then
           write(*,701) ng_ins,ng_ins_suc,
    &           ng_rem,ng_rem_suc
        endif
        if (disklev.ge.3) then
           write(202,701) ng_ins,ng_ins_suc,
    &           ng_rem,ng_rem_suc
        endif
     elseif (ensemble.eq.2) then
        if (termlev.ge.3) then
           write(*,504)'  N,Ng,rho_g,V_g=',n,ng,rhogauge,Vg
           write(*,701) ng_ins,ng_ins_suc,
    &           ng_rem,ng_rem_suc
        endif
        if (disklev.ge.3) then
           write(202,504)'  N,Ng,rho_g,V_g=',n,ng,rhogauge,Vg
           write(202,701) ng_ins,ng_ins_suc,
    &           ng_rem,ng_rem_suc
        endif
     endif

501  format('Set ',I8,'/',I8,' completed.   ',I3,'% done.')
503  format('  Uff=',E12.4,', Uint=',E12.4,', Usf=',E12.4,
    &     ', mu_ex=',E12.4)
505  format('  Uff=',E12.4,', Usf=',E12.4,
    &     ', N=',F10.2)
504  format(A,I5,i5,es12.3,es12.3)
701  format('   Exchange attempts,successes (i:r)=',I9,',',
    &     I6,' :',I9,',',I6)

     close(202);close(203);close(204)
     end subroutine set_output


     subroutine writecoor(filename)
     integer :: i
     character*50 filename
     open(file=filename,unit=11)
```

```
      do i=1,n
         write(11,*) x(i),y(i),z(i)
      enddo
      close(11)
      end subroutine writecoor



      subroutine writetraj(formattype)
      integer formattype,i
      real xcm(n),ycm(n),zcm(n)
      real r,xm,ym,zm,rm
      logical AnyPBC

      if (n.le.0) return
!File format, 0=XMOL
      XMOL(1)=’C’
      XMOL(2)=’N’
      XMOL(3)=’O’
      XMOL(4)=’S’

! rescale coordinates so center of mass is at origin, if there are no
    boundaries in the system.
      AnyPBC=.false.
      do i=1,3
         if (SetPBC(i)) AnyPBC=.true.
      enddo
      if (geom.ge.3 .and. .not.AnyPBC .and. .not.tethered(1)) then
         xm=0; ym=0; zm=0
         do i=1,N
            xm=xm+x(i)
            ym=ym+y(i)
            zm=zm+z(i)
         enddo
!rm=sqrt(xm*xm+ym*ym+zm*zm)
         xm=xm/n
         ym=ym/n
         zm=zm/n
         do i=1,N
            xcm(i)=x(i)-xm
            ycm(i)=y(i)-ym
            zcm(i)=z(i)-zm
         enddo
      else
         do i=1,N
            xcm(i)=x(i)
            ycm(i)=y(i)
            zcm(i)=z(i)
         enddo
      endif

      r=Li(1)/2.
      open(file=trajfile,unit=101,position=’append’)

      IF (tidy(trajtype).eq.’XMOL’) then
!if (geom.eq.3) write(29,’(I10)’) N
         write(101,’(I10)’) N
         write(101,*) ’After ’,(set)*(step-1),’ Steps’
         do i=1,N
```

```
            if (i.eq.1 .or .i.eq.N) then
               write(101,'(A,F10.3,F10.3,F10.3)') xmol(2),xcm(i),ycm(i),
     &              zcm(i)
            else
               write(101,'(A,F10.3,F10.3,F10.3)') xmol(1),xcm(i),ycm(i),
     &              zcm(i)
            endif
         enddo
         if (geom.ge.3) then    !draw in corners if cube
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),r,r,r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),r,r,-r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),r,-r,-r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),-r,-r,-r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),-r,-r,r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),-r,r,r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),-r,r,-r
!~ write(29,'(A,F10.3,F10.3,F10.3)') xmol(4),r,-r,r
         endif

      ELSEIF (tidy(trajtype).eq.'VTF ') then
! Make a VTF file
         if (set.eq.0) then
! Write the structure block
            write(101,'(2A)') "atom 0 radius 1.00 name ",xmol(2)
            do i=2,N-1
               write(101,'(A,I6,2A)') "atom ",i-1," radius 1.00 name ",
     &              xmol(1)
            enddo
            write(101,'(A,I6,2A)') "atom ",N-1," radius 1.00 name ",
     &           xmol(2)
            write(101,*)
            do i=1,N-1
               write(101,'(A,I6,A,I6)') "bond ",i-1,":",i
            enddo
            write(101,*)
         endif

         write(101,'(A)') "timestep indexed"
         write(101,'(A,3F10.1)') "pbc",Li(1),Li(2),Li(3)
         do i=1,N
            write(101,'(I6,3E15.5)') i,xcm(i),ycm(i),zcm(i)
         enddo
         write(101,*)
      end if

      close(101)

      end subroutine writetraj



      subroutine writesummary(filename)
      character*15 filename
      real*8 mu_f

      if (filename == 'NONE') return
! Summary of results
      open(file=filename, unit=25, status='NEW', err=233)
      write(25,'(A1,A9,A12,A6,A15,A12,A15,A15,A15,A15,A12)')
     &     "#","N","RHO","T","RHO_G","MU","UFF","USF","UKBL","UKBA","RG"
      GOTO 235
233   continue
      open(file=filename, unit=25, position='APPEND')
235   continue
      SELECT CASE (ENSEMBLE)
      case (0)
         mu_f=-T*log(Vins*rho_g)
      case (1)
         mu_f=mu_bulk
      case (2)
         mu_f=T*log(rho_g)
      end select
      write(25,123)
     &     rho*volume,rho,T,rho_g,mu_f,
     &     Uff_avg,Usf_avg,Ukbl_avg,Ukba_avg,Rg_avg
123   Format(F10.3,F12.6,F6.3,E15.6,F12.6,E15.6,E15.6,
     &     E15.6,E15.6,F12.6)
      close(25)
      end subroutine writesummary



      subroutine writeaverageenergies()
      open(file=logfile,position='APPEND',unit=26)

      write(*,*)
      write(*,'(A,E13.6,a,F8.3,a,E13.6,a,F8.3,A)')
     &     ' Average Energies:   Uff= ',
     &     Uff_avg,'(',Uff_avg/real(n),' per bead)'
      write(*,'(A26,E13.6,A,F8.3,A)') 'Usf= ',Usf_avg,'(',
     &     Usf_avg/real(n),' per bead)'
      write(*,'(A26,E13.6,A,F8.3,A)') 'Ukbl=',Ukbl_avg,'(',
     &     Ukbl_avg/real(n),' per bead)'
      write(*,'(A26,E13.6,A,F8.3,A)') 'Ukba=',Ukba_avg,'(',
     &     Ukba_avg/real(n),' per bead)'

      write(26,'(A,E13.6,a,F8.3,a,E13.6,a,F8.3,A)')
     &     ' Average Energies:   Uff= ',
     &     Uff_avg,'(',Uff_avg/real(n),' per bead)'
      write(26,'(A26,E13.6,A,F8.3,A)') 'Usf= ',Usf_avg,'(',
     &     Usf_avg/real(n),' per bead)'
      write(26,'(A26,E13.6,A,F8.3,A)') 'Ukbl=',Ukbl_avg,'(',
     &     Ukbl_avg/real(n),' per bead)'
      write(26,'(A26,E13.6,A,F8.3,A)') 'Ukba=',Ukba_avg,'(',
     &     Ukba_avg/real(n),' per bead)'

      close(26)

      end subroutine writeaverageenergies

      subroutine writefinalle()
      real*8 olduff, oldusf, oldukbl, mu_bond, dbw

      dBW=h/sqrt(2.0*pi*mass*kb*T*eps) !m
      dBW=dBw*1e10/sigma        !per sigma (sigma in variable stored in
         Angstroms)
      if (hardbond) then
         mu_bond = 0.d0
      else
         mu_bond = -T*log(4.0*pi*
     &        sqrt(2.0*pi)*(2.0*kb0_len*r0**2+T)/(2.0*kb0_len*sqrt(2.0*
     &        kb0_len/T)))
      end if

      open(file=logfile,position='APPEND',unit=26)

      olduff=uff; oldusf=usf; oldukbl=ukbl
      call calc_energy(.false.)
      write(*,*)
      Write(*,*) 'Uff: Running-Now:', olduff-uff
      Write(*,*) 'Usf: Running-Now:', oldusf-usf
      Write(*,*) 'Ukbl: Running-Now:', oldukbl-ukbl
      Write(*,*)
      Write(*,*), 'SIMULATION COMPLETE!'
      Write(*,*), '  <N>=',rho*volume
      if (calc_bondl) Write(*,*), '  <l>=',bl_avg
!if (calc_bondl) print *, '  RMS= ',RMS_avg
      if (calc_bondl) Write(*,*), '  <L>/<R>=',tort_avg
      if (calc_bondl) Write(*,*), '  sqrt(<S2>)=',Rg_avg
      if (calc_bondl) Write(*,*), '  sqrt(<R2>)=',RMS_1N_avg
      if (calc_bondl) Write(*,*), '  <S2>/<R2>=',
     &     Rg_avg**2/RMS_1N_avg**2
      Write(*,*), '  <rho>=',rho
      if (ensemble.eq.0) then
         write(*,*), '  <exp(-U/kT)>=',rho_g
         if (bonds) then
            Write(*,*), '  <mu_incr[bond]/e>= ', mu_bond
            if (ins_bias) then
               write(*,*), '  <mu_incr[ex]/e>=',
     &              -T*log(Vins*rho_g)-mu_bond
               write(*,*), '  <mu_incr/e>=', -T*log(Vins*rho_g)
            else
               write(*,*), '  <mu_incr[ex]/e>=', -T*log(Vins*rho_g)
               write(*,*), '  <mu_incr/e>=',-T*log(Vins*rho_g)+
     &              mu_bond
            end if
         else
            Write(*,*), '  <mu_id/e>=',mu_id
            write(*,*), '  <mu[ex]/e>=',-T*log(rho_g)
            write(*,*), '  <mu/e>=',-T*log(rho_g)+mu_id
         end if
      elseif (ensemble.eq.2) then
         write(*,*), '  <rho_g>=',rho_g
         if (bonds) then
            Write(*,*), '  <mu_incr[bond]/e>= ', mu_bond
            if (ins_bias) then
               write(*,*), '  <mu_incr[ex]/e>=',
     &              -T*log(rho_g)-mu_bond
               write(*,*), '  <mu_incr/e>=', T*log(rho_g)
            else
               write(*,*), '  <mu_incr[ex]/e>=', T*log(rho_g)
               write(*,*), '  <mu_incr/e>=',T*log(rho_g)+
     &              mu_bond
            end if
         else
            Write(*,*), '  <mu_id/e>=',mu_id
            write(*,*), '  <mu[ex]/e>=',T*log(rho_g)
            write(*,*), '  <mu/e>=',T*log(rho_g)+mu_id
         end if
      elseif (ensemble.eq.1) then
         Write(*,*), '  mu/eps=',mu_bulk
      endif
      Write(*,*), '----------------------------------------------'

! write to log
      write(26,*)
      Write(26,*) 'Uff: Running-Now:', olduff-uff
      Write(26,*) 'Usf: Running-Now:', oldusf-usf
      Write(26,*) 'Ukbl: Running-Now:', oldukbl-ukbl
      Write(26,*)
      Write(26,*), 'SIMULATION COMPLETE!'
      Write(26,*), '  <N>=',rho*volume
      if (calc_bondl) Write(26,*), '  <l>=',bl_avg
!if (calc_bondl) print *, '  RMS= ',RMS_avg
      if (calc_bondl) Write(26,*), '  <L>/<R>=',tort_avg
      if (calc_bondl) Write(26,*), '  <S>=',Rg_avg
      if (calc_bondl) Write(26,*), '  <R>=',RMS_1N_avg
      if (calc_bondl) Write(26,*), '  <S2>/<R2>=',
     &     Rg_avg**2/RMS_1N_avg**2
      Write(26,*), '  <rho>=',rho
      if (ensemble.eq.0) then
         write(26,*), '  <exp(-U/kT)>=',rho_g
         if (bonds) then
            Write(26,*), '  <mu_incr[bond]/e>= ', mu_bond
            if (ins_bias) then
               write(26,*), '  <mu_incr[ex]/e>=',
     &              -T*log(Vins*rho_g)-mu_bond
               write(26,*), '  <mu_incr/e>=', -T*log(Vins*rho_g)
            else
               write(26,*), '  <mu_incr[ex]/e>=', -T*log(Vins*rho_g)
               write(26,*), '  <mu_incr/e>=',-T*log(Vins*rho_g)+
     &              mu_bond
            end if
         else
            Write(26,*), '  <mu_id/e>=',T*log(dbw**3*(N+1)/volume)
            write(26,*), '  <mu[ex]/e>=',-T*log(rho_g)
            write(26,*), '  <mu/e>=',-T*log(rho_g)+mu_id
         end if
      elseif (ensemble.eq.2) then
         write(26,*), '  <rho_g>=',rho_g
         if (bonds) then
            Write(26,*), '  <mu_incr[bond]/e>= ', mu_bond
            if (ins_bias) then
               write(26,*), '  <mu_incr[ex]/e>=',
     &              -T*log(Vins*rho_g)-mu_bond
               write(26,*), '  <mu_incr/e>=', T*log(rho_g)
            else
               write(26,*), '  <mu_incr[ex]/e>=', T*log(rho_g)
               write(26,*), ' all vars: ',T,Vins,rho_g,mu_avg
               write(26,*), '  <mu_incr/e>=', T*log(rho_g)+
     &              mu_bond
            end if
         else
            Write(26,*), '  <mu_id/e>=',mu_id
            write(26,*), '  <mu[ex]/e>=',T*log(rho_g)
            write(26,*), '  <mu/e>=',T*log(rho_g)+mu_id
         end if
      elseif (ensemble.eq.1) then
         Write(26,*), '  mu/eps=',mu_bulk
      endif
      Write(26,*), '----------------------------------------------'

      if (step_warning) then
         write(*,*) 'WARNING: Step size was less than 1e-4. Poor '//
     &        'sampling may have occured!'
         write(26,*) 'WARNING: Step size was less than 1e-4. Poor '//
     &        'sampling may have occured!'
      endif
      close(26)

      end subroutine writefinalle

      end module outputs
```

B.7 constants.f

```
      module constants
      implicit none

      real*8 pi,kb,h,Na
      parameter( pi=3.14159265358979323846264
     &     338327950288419716939937510582097450 )
      parameter( kb=1.3806503E-23 ) !(m2*kg/s-2*K-1) or (J/K)
      parameter( h=6.62606896E-34 ) !(m2*kg/s) or (J*s)
      parameter( Na=6.0221415e23 ) !per mole

      contains

!!! Extra Math Functions
!! Linear Interpolation
      real*8 function interpolate(t,x2,x1,y2,y1)
      real*8 t,x1,x2,y1,y2
      interpolate=y1+(t-x1)*(y2-y1)/(x2-x1)
      end function

!! Factorial
      integer*4 function fact(n_)
      integer*4 n_
      integer i
      fact=1
      do i=1,n_
         fact=fact*i
      enddo
      end function fact

!! Combinatorial
      INTEGER*4 function comb(n_,r_)
      integer*4 n_, r_, a_
      integer i
      if (r_ > n_) then
         comb=0
         return
      endif
      a_ =  (n_ - r_)+1
      comb=1
      do i=a_, n_
         comb=comb*i
      enddo
      comb = comb/fact(r_)
      end function comb

!! Permutation
      integer*4 function perm(n_,r_)
      integer*4 n_, r_, a_
      integer i
      if (r_ > n_) then
         perm=0
         return
      endif
      a_ =  (n_ - r_)+1
      perm=1
      do i=a_, n_
         perm=perm*i
      enddo
      end function perm


! Estimate standard deviation using blocking method
      subroutine blocking_avg(array,samp,start,block_avg,stddev,conv)
      integer,intent(in) :: samp,start
      real blocks_ignored
      real*8 :: array(samp)
      real*8,intent(out) :: block_avg,stddev
      logical,intent(out) :: conv
      real*8 mean,mean2,var,bvar,sd,evar,esd,fl,flm1
      real*8,allocatable :: sdm(:),esdm(:),bl_max(:)
      integer i,j,k,avg_set,Mb,l,lm1

! Calculate maximum number of blocks
      avg_set=(samp-start+1)
      Mb=0
      do while (avg_set.gt.1)
         Mb=Mb+1
         avg_set=avg_set/2
      enddo
      allocate(sdm(Mb),esdm(Mb))

! Re-index the data array
      mean=0.0; avg_set=0
      do i=start,samp
         avg_set=avg_set+1
         array(avg_set)=array(i)
      enddo

! Calculate the total mean
      mean=0.0
      do i=1,avg_set
         mean=mean+array(i)
      enddo
      mean=mean/dble(avg_set)
      block_avg=mean
      mean2=mean*mean
! Calculate total variance
```

```
      bvar=0.d0; var=0.d0
      do i=1,avg_set
         bvar=bvar+(array(i)**2-mean2)
      enddo
      bvar=bvar/dble(avg_set)
      var=bvar/dble(avg_set-1)
      sd=sqrt(var)
      evar=var*sqrt(dble(2)/dble(avg_set-1))
      esd=sd*1.0/sqrt(2.0*dble(avg_set-1))
      sdm(1)=sd; esdm(1)=esd

! Begin blocking operations
      l=avg_set
      do j=2,Mb
         l=l/2; lm1=l-1
         fl=float(l); flm1=float(lm1)
         do i=1,l
            k=2*i
            array(i)=0.5*(array(k-1)+array(k))
         enddo
         mean=0.d0
         do i=1,l
            mean=mean+array(i)
         enddo
         mean=mean/fl
         mean2=mean*mean
         bvar=0.d0
         do i=1,l
            bvar=bvar+(array(i)**2-mean2)
         enddo
         bvar=bvar/fl
         var=bvar/flm1
         sd=sqrt(var)
         evar=var*sqrt(2.0/flm1)
         esd=sd*1.0/sqrt(2.0*flm1)
         sdm(j)=sd; esdm(j)=esd
      enddo

! Estimate std. dev.
      allocate( bl_max(Mb) )
      do i=1,Mb
         bl_max(i) = sdm(i)+esdm(i)
      enddo
!stddev=max(bl_max(:))

! Check for convergence
      conv=.false.

      end subroutine blocking_avg


!! Handy Functions that FORTRAN left out!
      function upcase(string) result(upper)
      character(len=*), intent(in) :: string
      character(len=len(string)) :: upper
      integer :: j
      do j = 1,len(string)
         if(string(j:j) >= "a" .and. string(j:j) <= "z") then
            upper(j:j) = achar(iachar(string(j:j)) - 32)
         else
            upper(j:j) = string(j:j)
         end if
      end do
      end function upcase

!! Tidy up long strings, e.g. make them upper case and remove
   leading/trialing spaces
      function tidy(string) result(output)
      character(len=*), intent(in) :: string
      character(len=len(string)) :: output
      output=upcase(string)
      output=trim(output)
      end function tidy


!! Input time as seconds, returns neatly formated string in
   days/hours/minutes/seconds.
      function time_str(timein) result(timeout)
      character*51 timeout
      character*8 strint8
      character*5 strint5
      character*2 strint2(2)
      real*8 timein,hours,minutes,days,secs
      integer i,j

      if (timein.lt.0.01) then
         timeout='less than 1 second.'
         return
      endif

      secs=timein
      minutes=secs/60.
      hours=minutes/60.
      days=hours/24.

      if (days.gt.1) then
         write(strint8,'(I8)') int(days)
         hours=(days-int(days))*24.
         write(strint2(1),'(I2)') int(hours)
         minutes=(hours-int(hours))*60.
         write(strint2(2),'(I2)') int(minutes)
         secs=(minutes-int(minutes))*60.
         write(strint5,'(F5.2)') secs
         timeout=strint8//' days, '//strint2(1)//' hours, '//
     &        strint2(2)//' minutes, '//strint5//' seconds.'
         return
      endif
      if (hours.gt.1) then
         write(strint2(1),'(I2)') int(hours)
         minutes=(hours-int(hours))*60.
         write(strint2(2),'(I2)') int(minutes)
         secs=(minutes-int(minutes))*60.
         write(strint5,'(F5.2)') secs
         timeout=strint2(1)//' hours, '//strint2(2)//' minutes, '//
     &        strint5//' seconds.'
         return
      endif
      if (minutes.gt.1) then
```

```
         write(strint2(2),'(I2)') int(minutes)
         secs=(minutes-int(minutes))*60.
         write(strint5,'(F5.2)') secs
         timeout=strint2(2)//' minutes, '//strint5//' seconds.'
         return
      endif
      write(timeout,'(F5.2,A)') secs,' seconds. '
      return
      end function time_str


      subroutine timestamp ( )
!**************************************************************
!
!     ! TIMESTAMP prints the current YMDHMS date as a time stamp.
!
!     Example:
!
!     May 31 2001   9:45:54.872 AM
!
!     Licensing:
!
!     This code is distributed under the GNU LGPL license.
!
!     Modified:
!
!     31 May 2001
!
!     Author:
!
!     John Burkardt
!
!     Parameters:
!
!     None
!
      implicit none

      character ( len = 8 )  ampm
      integer   ( kind = 4 ) d
      character ( len = 8 )  date
      integer   ( kind = 4 ) h
      integer   ( kind = 4 ) m
      integer   ( kind = 4 ) mm
      character ( len = 9 ), parameter, dimension(12) :: month =
     &     (/ 'January  ', 'February ', 'March    ', 'April    ',
     &     'May      ', 'June     ', 'July     ', 'August   ',
     &     'September', 'October  ', 'November ', 'December ' /)
      integer   ( kind = 4 ) n
      integer   ( kind = 4 ) s
      character ( len = 10 ) time
      integer   ( kind = 4 ) values(8)
      integer   ( kind = 4 ) y
      character ( len = 5 )  zone

      call date_and_time ( date, time, zone, values )

      y = values(1)
      m = values(2)
      d = values(3)
      h = values(5)
      n = values(6)
      s = values(7)
      mm = values(8)

      if ( h < 12 ) then
         ampm = 'AM'
      else if ( h == 12 ) then
         if ( n == 0 .and. s == 0 ) then
            ampm = 'Noon'
         else
            ampm = 'PM'
         end if
      else
         h = h - 12
         if ( h < 12 ) then
            ampm = 'PM'
         else if ( h == 12 ) then
            if ( n == 0 .and. s == 0 ) then
               ampm = 'Midnight'
            else
               ampm = 'AM'
            end if
         end if
      end if

      write ( *,'(a,1x,i2,1x,i4,2x,i2,a1,i2.2,a1,i2.2,a1,i3.3,1x,a)'
     &     ) trim ( month(m) ), d, y, h, ':', n, ':', s, '.', mm,
     &     trim ( ampm )

      return
      end subroutine timestamp

      end module constants
```

B.8 fort_rand.f

```
      module fort_rand
      implicit none
      contains

! Initialize FORTRAN random number generator
      subroutine rand_init(rseed)
      integer rseed,k,q,clock
      integer, dimension(:), allocatable :: seed
      call random_seed(size = q)
      allocate(seed(q))
      if (rseed.eq.0) then
!If 0 is passed for rseed, use clock instead
         call system_clock(count=clock)
      else
         clock=rseed
      endif
      seed = clock + 37 * (/ (k - 1, k = 1, q) /)
      call random_seed(put = seed)
      return
      end subroutine rand_init
```

```
! Function to call a random number from the subroutine
      real*8 function rnd()
      real*8 X
      call RANDOM_NUMBER(X)
      rnd=X
      return
      end function rnd

! Function to call a random number center around 0 [aka bound (-0.5,0.5)]
      real*8 function rndz()
      real*8 X
      call RANDOM_NUMBER(X)
      rndz=X-0.5
      return
      end function rndz


! Random integer from 1 to i
      integer function rndint(i)
      integer i
      real*8 X
      call RANDOM_NUMBER(X)
      rndint=int(X*i)+1
      return
      end function rndint


! Random number from Gaussian distribution
      real*8 function rndgauss(sigma,mean)
      real*8 sigma, mean,r,v1,v2,x
      r=2.0
      do while (r.ge.1.0)
         call random_number(v1)
         call random_number(v2)
         v1 = 2.0*v1-1.0
         v2 = 2.0*v2-1.0
         r=v1*v1+v2*v2
      enddo
      x=v1*dsqrt(-2.0*log(r)/r)
      rndgauss=mean+sigma*x
      return
      end function rndgauss


!! Generates random unit vector on a sphere
!!  Based on Algorithim 42 in Frenkel&Smit
      subroutine ran_nxyz(nv)
      real*8,dimension(3) :: nv !unit vector
      real*8 :: ransq,ran1,ran2,ranh
      ransq=2.
      do while (ransq.ge.1)
         call random_number(ran1)
         call random_number(ran2)
         ran1=1.-2.*ran1
         ran2=1.-2.*ran2
         ransq=ran1*ran1+ran2*ran2
      enddo
```

```
       ranh=2.*sqrt(1.-ransq)
       nv(1)=ran1*ranh
       nv(2)=ran2*ranh
       nv(3)=(1-2.*ransq)
       return
       end subroutine ran_nxyz


       end module fort_rand
```

C gradientInt.py code

This section contains the code developed for gradient-elution chromatography, Section 4.3. It was developed with Python and SciPy [306] with an object-oriented framework, allowing reusable scripts for different project, and easy extension of Python's many tools.

```
#!/usr/bin/env python
# GRADIENTINT.PY -- v1.2.0 beta
# Object-oriented framework for gradient elution integrator
# Consists of input and output objects user can create & modify
import sys, os
from scipy import *
from scipy.integrate import odeint, quad
from scipy.optimize import newton
import matplotlib as mpl
from matplotlib import pyplot as plot
from matplotlib import rc
import copy, random
from partitionFuncs import *

grid = 1000 # grid mesh for ODE solver, increase for accuracy, decrease
            # for speed
narrowPore = 10 # when is a narrow pore a narrow pore?
                # (when Rg/D/2 is <= to this value);
                # used in hybrid mode to determine switch over to
                #   asymptotic values.
debug = False # print data during integration
models  = {0:"Analytical", 1:"Near critical", 2:"General solution",
           3:"General/NC hybrid", 4:"Adsorption", 5:"Narrow"}
           # just a label for pretty formatting...
solvent = {0:"Constant", 1:"Linear Approx", 2:"Error Function"}
           # Available solvent models

# INPUT CLASSES:
# Polymer classes

class polymer:
# generic class, create it with poylmerType as name of K_function
    def __init__(self, polymerType, columnObj):
        # bookkeeping
        self.polymerModel = polymerType
        self.partitionFunc = "K_" + polymerType
        self.name = "polymer"
        self.column = columnObj
        self.MW = [3600] # list of molecular weights of polymers
        self.g  = [] # not yet used
        self.tgSoln = [] # contains solution to above
        self.Mi = [1] # MW of arms (normalized to 1)
        # model parameters (gradient)
        self.Fcr = [40.7]
        self.b = [1.0] # Radius of gyration coefficient
        self.s = [0.5] # Radius of gyration scaling
        self.dldF = [5e-5] # dlambda/dPhi
        self.solventModel = [1] # which solvent model to use
        # non-gradient parameters
        self.lmbda = [-1.0]
        self.qi = [0.0, 0.0] # functional group strength
        self.Kmax = 1e10
        self.erfRange = 100.

        # plotting parameters
        self.color = None
        self.marker = None
        self.style = ’-’
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    # generate a Markovian statistical copolymer with given parameters
    # Pi, lambda, dldF, solventModel must be lists the length of the
    #   number of species!
    # Pi is the overall ratio of chemical species,
    def stat_copolymer(Pi, vij, lmbda, dldF, solventModel, N, gChain):
        if self.polymerModel != ’copolymer’:
            print "Statistical copolymers must use the \’copolymer\’ type!"
            return
        gChain, backbone = generate_statistical_copolymer( Pi, vij,
                             range(len(lmbda)), N, gChain, silent=True )

        self.Mi = gChain
        self.lmbda = []
        self.dldF = []
        self.solventModel = []
        for i in backbone:
            self.lmbda.append( lmbda[i] )
            self.dldF.append( dldF[i] )
            self.solventModel( solventModel[i] )
        self.normalize(silent=True)

    def normalize(self, silent=False):
        self.Mi = array(self.Mi,dtype=float32)**2 /
                    sum(array(self.Mi,dtype=float32)**2)
        if not silent:
            print "Normalized MW per arm, ", self.Mi

    def Rg(self,M):
        return self.b[0] * M**(self.s[0])

    def Rg2(self,M):
        return self.Rg(M)**2

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.currentFg(tg,1.0)
    def info(self):
        # prints information about this instance
        print "\'polymer\' class -- %s"%self.name
        print " Model: %s (model=%i)"%(models[self.model],self.model)
        print " Column: %s (column=columnObject)"%(self.column.name)
        print " Scaling, R_g = b*M^s: b = %5.2f, s = %5.2f"%(self.b, self.s)
        print " Critical composition: %5.2f %% (Fcr=%5.2f)"%(self.Fcr,self.Fcr)
        print " Combined parameter: %5.3g 1/Da (A=%5.3g)"%(self.A,self.A)
        print " Interaction energy per composition: %6.4g kT/%% (dedF=%6.4g)" %
                (self.dedF,self.dedF)
        print " Interaction distance: %5.3f nm (a = %5.3f)"%(self.a,self.a)

    def tg(self, M):
        # solve for t_g for a certain MW
        xrange = linspace(0, 1, grid)
        t0 = 0
        ode = odeint(self.dtdx, t0, xrange, args=(M,), rtol=1e-5)
        return ode[-1][0]

    def K(self, t, x, M):
        # Current solvent strength
        F = self.column.currentFg(t[0],x)

        g = [2.0*self.Rg(i*M)/self.column.D for i in self.Mi]

        lmda = []
        for block in range(len(self.Mi)):
            # Constant, i.e. isocratic solvent model
            if self.solventModel[block] == 0:
                lmda.append( self.lmbda[block] )
            # Linear approximation
            elif self.solventModel[block] == 1:
                lmda.append( self.dldF[block]*(F - self.Fcr[block]) )
            elif self.solventModel[block] == 2:
                lmda.append( self.erfRange*erf(self.dldF[block] *
                               (F - self.Fcr[block])) )

        # return K from partitionFuncs
        Ksum = returnKvalue(self.partitionFunc, g, lmda, self.qi)
        #print min(Ksum, self.Kmax), lmda, g, F, t, x
        return min(Ksum, self.Kmax)

    def dtdx(self, t, x, M):
        # equation to integrate to find tg
        return self.column.t0 + self.K(t, x, M)*self.column.tP

class homopolymer:
    def __init__(self, columnObj):
        # bookkeeping
        self.model = 0
        self.name = "PS"
        self.column = columnObj
        self.MW = [3600]
        self.tgSoln = []
        self.Mi = [1]
        # model parameters
        self.f = 1
        self.A = 8.5e-5
        self.Fcr = 40.7
        self.a = 0.1 #nm
        self.b = 1.0
        self.s = 0.5
        self.dedF = 8.5e-5 * self.column.D *self.a / 2 / 5.0
        # plotting parameters
        self.color = None
        self.marker = None
        self.style = ’-’
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    def normalize(self):
        pass

    def calcA(self, silent=False):
        # calculate the fitting parameter A from other parameters
        self.A = 2*self.dedF*self.column.Fpr/self.column.D/self.a
        if not silent: print "A = %g"%self.A

    def Rg(self,M,f=1):
        return self.b * M**(self.s)

    def Rg2(self,M):
        return self.Rg(M)**2

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.Fi + self.column.dFdt*(tg -
                 (self.column.t0 + self.column.tP))

    def info(self):
        # prints information about this instance
        print "Homopolymer class -- %s"%self.name
        print " Model: %s (model=%i)"%(models[self.model],self.model)
        print " Column: %s (column=columnObject)"%(self.column.name)
        print " Scaling, R_g = b*M^s: b = %5.2f, s = %5.2f"%(self.b, self.s)
        print " Critical composition: %5.2f %% (Fcr=%5.2f)"%(self.Fcr,self.Fcr)
        print " Combined parameter: %5.3g 1/Da (A=%5.3g)"%(self.A,self.A)
        print " Interaction energy per composition: %6.4g kT/%% (dedF=%6.4g)" %
                (self.dedF,self.dedF)
        print " Interaction distance: %5.3f nm (a = %5.3f)"%(self.a,self.a)

    def tg(self, M):
        # solve for t_g for a certain MW
        Q = self.A*M
        if self.column.dFdt == 0: # isocratic case
            tg = self.column.t0 + self.column.tP * self.K(1.0, 1.0, M)
            # passed variables do not matter in isocratic case
            return tg
        if self.model in [0]:
            Fg = self.Fcr + (self.column.Fpr/Q)*log(1-exp(-Q) +
                 exp(Q/self.column.Fpr*(self.column.Fi-self.Fcr)-Q))
            return (Fg - self.column.Fi)/self.column.dFdt +
                     (self.column.t0 + self.column.tP)
        if self.model in [1,2,3]:
            xrange = linspace(0, 1, grid)
            t0 = 0
            return odeint(self.dtdx, t0, xrange, args=(M,))[-1][0]

    def K(self, t, x, M):
        F = min(100,self.column.Fi + self.column.dFdt*
                  (t - (self.column.t0 + self.column.tP)*x))
        g = 2.0*self.Rg(M)/self.column.D
        if self.model in [1] or (self.model == 3 and g > narrowPore):
            if self.column.Fpr == 0:
                Ksum = exp( 2*self.Rg(M)**2/self.column.D/self.a*self.dedF *
                            (self.Fcr - F) )
            else:
                Ksum = exp(self.A*M/self.column.Fpr*(self.Fcr - F))
        elif self.model in [2] or (self.model == 3 and g <= narrowPore):
            lmbda = float(self.dedF*(F - self.Fcr)*(self.column.D/2/self.a))
            Ksum = K_general(g, lmbda)
        elif self.model == 4: # Radke's model
            Ksum = K_adsorption(g, lmbda)
        elif self.model == 5: # Narrow pores
            Ksum = K_narrow(g, lmbda)
        else:
            print "Model %i not found in homopolymer class" % self.model
            sys.exit(1)
        if Ksum > Kmax: Ksum = Kmax
        return Ksum

    def dtdx(self, t, x, M):
        # equation to integrate to find tg
        return self.column.t0 + self.K(t, x, M)*self.column.tP

class functional:
    def __init__(self, columnObj):
        # bookkeeping
        self.model = 2
        self.name = "PS"
        self.column = columnObj
        self.MW = [3600]
        self.tgSoln = []
        self.Mi = [1]
        # model parameters
        self.f = 1
        self.A = 8.5e-5
        self.q = [1.0, 0.0]
        self.Fcr = 40.7
        self.a = 0.1 #nm
        self.b = 1.0
        self.s = 0.5
        self.dedF = 8.5e-5 * self.column.D *self.a / 2 / 5.0
        # plotting parameters
        self.color = None
        self.style = "-"
        self.marker = 'None'
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    def calcA(self):
        # calculate the fitting parameter A from other parameters
        self.A = 2*self.dedF*self.column.Fpr/self.column.D/self.a
        print "A = %g"%self.A

    def normalize(self):
        pass

    def Rg(self,M,i=1):
        return self.b * M**(self.s)

    def Rg2(self,M):
        return self.Rg(M)**2

    def info(self):
        # prints information about this instance
        print "Functional class -- %s"%self.name
        print " Model: %s (model=%i)"%(models[self.model],self.model)
        print " Column: %s (column=columnObject)"%(self.column.name)
        print " Scaling, R_g = b*M^s: b = %5.2f, s = %5.2f"%(self.b, self.s)
        print " Critical composition: %5.2f %% (Fcr=%5.2f)"%(self.Fcr,self.Fcr)
        print " Combined parameter: %5.3g 1/Da (A=%5.3g)"%(self.A,self.A)
        print " Interaction energy per composition: %6.4g kT/%% (dedF=%6.4g)" %
                (self.dedF,self.dedF)
        print " Interaction distance: %5.3f nm (a = %5.3f)"%(self.a,self.a)
        print " Functional parameter: %s (q = "%list2str(self.q),self.q,")"

    def tg(self, M):
        # solve for t_g for a certain MW
        Q = self.A*M
        if self.column.dFdt == 0: # isocratic case
            solveMe = lambda t_R: self.column.t0 + self.column.tP *
                                    self.K(t_R, 1.0, M) - t_R
            guess = 1
            t_R = newton(solveMe, guess)
            return t_R
        if self.model in [0]:
            print "No analytical expression exists for functionalized polymers."
            sys.exit(1)
        if self.model in [1,2,3]:
            xrange = linspace(0, 1, grid)
            return odeint(self.dtdx, 0, xrange, args=(M,))[-1][0]

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.Fi + self.column.dFdt*( tg -
                                  (self.column.t0 + self.column.tP) )

    def K(self, t, x, M):
        F = min(100,self.column.Fi + self.column.dFdt*( t -
                                       (self.column.t0 + self.column.tP)*x) )
        # current composition
        if self.model in [1]: # Near critical/narrow pore approximation
            if self.column.Fpr == 0:
                Ksum = exp(2*self.Rg(M)**2/self.column.D/self.a*self.dedF *
                           (self.Fcr - F)) * (1+self.q[0]*(1-(self.Fcr-F) *
                           self.dedF*self.column.D/6/self.a)) * (1+self.q[1] *
                           (1-(self.Fcr-F)*self.dedF*self.column.D/6/self.a))
            else:
                Ksum = exp(self.A*M/self.column.Fpr*(self.Fcr - F)) *
                           (1+self.q[0] * (1-(self.Fcr-F)*self.dedF *
                           self.column.D/6/self.a))*(1+self.q[1] *
                           (1-(self.Fcr-F)*self.dedF*self.column.D/6/self.a))

        elif self.model in [2]: # General solution
            g = 2.0*self.Rg(M)/self.column.D
            lmbda = float(self.dedF*(F - self.Fcr)*(self.column.D/2/self.a))
            pa = 0; pb = 0; pab = 0
            if self.q[0] != 0:
                pa = p_a(g,lmbda)
            if self.q[1] != 0:
                pb = p_a(g,lmbda)
            if prod(self.q) != 0:
                pab = p_ab(g,lmbda)
            Ksum = K_general(g,lmbda) + self.q[0]*pa + self.q[1]*pb +
                     prod(self.q)*pab

        elif self.model in [3]: # General/narrow pore hybrid
            g = 2.0*self.Rg(M)/self.column.D
            if (g <= 1):
                lmbda = float(self.dedF*(F - self.Fcr)*(self.column.D/2/self.a))
                pa = 0; pb = 0; pab = 0
                if self.q[0] != 0:
                    pa = p_a(g,lmbda)
                if self.q[1] != 0:
                    pb = p_a(g,lmbda)
                if prod(self.q) != 0:
                    pab = p_ab(g,lmbda)
                Ksum = K_general(g,lmbda) + self.q[0]*pa + self.q[1]*pb +
                         prod(self.q)*pab
            else:
                if self.column.Fpr == 0:
                    Ksum = exp( 2*self.Rg(M)**2/self.column.D/self.a*self.dedF *
                               (self.Fcr - F)) *  (1+self.q[0]*(1-(self.Fcr-F) *
                               self.dedF*self.column.D/6/self.a)) *
                               (1+self.q[1]*(1-(self.Fcr-F) * self.dedF *
                               self.column.D/6/self.a) )
                else:
                    Ksum = exp( self.A*M/self.column.Fpr*(self.Fcr - F)) *
                               (1+self.q[0] * (1-(self.Fcr-F)*self.dedF *
                               self.column.D/6/self.a))*(1+self.q[1] *
                            (1-(self.Fcr-F)*self.dedF*self.column.D/6/self.a) )

        else:
            print "Model %i not found in functional class" % self.model
            sys.exit(1)
        if Ksum > Kmax: Ksum = Kmax
        return Ksum

    def dtdx(self, t, x, M):
        # equation to integrate to find tg
        return self.column.t0 + self.K(t, x, M)*self.column.tP

class star_block:
    def __init__(self, columnObj):
        # bookkeeping
        self.model = 2
        self.name = "Star block-copolymer"
        self.column = columnObj
        self.MW = [3600]
        self.tgSoln = []
        # model parameters
        self.f = 4 # number of arms
        self.A = [8.5e-5, 8.5e-5, 8.5e-5, 8.5e-5] # lumped parameter
        self.Fcr = [40.7, 50.2, 40.7, 39.6]
          # Critical elution compositions for each arm
        self.Mi = [1.0, 1.0, 1.0, 1.0]
          # Relative weight of each arm, normalized to 1
        self.a = [0.1, 0.1, 0.1, 0.1] #nm
        self.b = [1.0, 1.0, 1.0, 1.0]
        self.s = [0.5, 0.5, 0.5, 0.5]
        self.dedF = [8.5e-6, 8.5e-6, 8.5e-6, 8.5e-6]
            # change of interaction energy per percent elutant, for each arm
        # plotting parameters
        self.color = None
        self.style = "-"
        self.marker = ’None’
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    def arms(self,arms):
        # resizes length of lists to specificed arms
        # overwrites existing values with first value!!
        self.f = arms
        # convert any existing lists to floats.
        # Assumes its already a float if not a list.
        if isinstance(self.A,list): self.A = self.A[0]
        if isinstance(self.Mi,list): self.Mi = self.Mi[0]
        if isinstance(self.Fcr,list): self.Fcr = self.Fcr[0]
        if isinstance(self.a,list): self.a = self.a[0]
        if isinstance(self.b,list): self.b = self.b[0]
        if isinstance(self.s,list): self.s = self.s[0]
        if isinstance(self.dedF,list): self.dedF = self.dedF[0]
        # Now set the list to proper size
        self.A = [self.A for i in range(arms)]
        self.Mi = [self.Mi for i in range(arms)]
        self.Fcr = [self.Fcr for i in range(arms)]
        self.a = [self.a for i in range(arms)]
        self.b = [self.b for i in range(arms)]
        self.s = [self.s for i in range(arms)]
        self.dedF = [self.dedF for i in range(arms)]

    def normalize(self):
        self.Mi = array(self.Mi,dtype=float32)**2 /
                         sum(array(self.Mi,dtype=float32)**2)
        print "Normalized MW per arm, ", self.Mi

    def calcA(self):
        # calculate the fitting parameter A from other parameters
        self.A = [2*self.dedF[i]*self.column.Fpr/self.column.D/self.a[i]
                   for i in range(self.f)]
        print "A = ", self.A

    def Rg(self,M,f):
        # radius of gyration of each arm
        return self.b[f] * M**(self.s[f])
```

```
def Rg2(self,M,f):
    # radius of gyration squared
    return self.Rg(M,f)**2

def info(self):
    # prints information about this instance
    print "Star block-copolymer class -- %s" % self.name
    print " Model: %s (model=%i)" % (models[self.model],self.model)
    print " Column: %s (column=columnObject)" % (self.column.name)
    print " Scaling per arm, R_g = b*M^s: b = %s, s = %s" %
            (list2str(self.b), list2str(self.s))
    print " Critical composition per arm: %s %% (Fcr=[%s])" %
            (list2str(self.Fcr),list2str(self.Fcr))
    print " Combined parameter: %s 1/Da (A=[%s])" %
            (list2str(self.A),list2str(self.A))
    print " Interaction energy per composition: %s kT/%% (dedF=[%s])" %
            (list2str(self.dedF),list2str(self.dedF))
    print " Interaction distance: %s nm (a = [%s])" %
            (list2str(self.a),list2str(self.a))

def tg(self, M):
    # solve for t_g for a certain MW
    #Q = [self.A[i]*M*Mi[i] for i in range(self.f)]
    if self.column.dFdt == 0: # isocratic case
        tg = self.column.t0 + self.column.tP * self.K(1.0, 1.0, M)
        return tg
    if self.model in [0]:
        print "No analytical expression exists for star polymers."
        sys.exit(1)
    if self.model in [1,2,3]:
        xrange = linspace(0, 1, grid)
        return odeint(self.dtdx, 0, xrange, args=(M,), rtol=1e-4)[-1][0]

def Vg(self, tg):
    # return V_g from inputed t_g
    return self.column.F * tg

def Fg(self, tg):
    # return Phi_G from inputed t_g
    return self.column.Fi + self.column.dFdt*(tg -
            (self.column.t0+self.column.tP))

def K(self, t, x, M):
    F = min(100, self.column.Fi + self.column.dFdt*(t -
            (self.column.t0 + self.column.tP)*x))
        # current composition
    Mi = M*array(self.Mi,dtype=float32)
    if self.model in [1]: # Near critical/narrow pore approximation
        if self.column.Fpr == 0:
            for i in range(self.f):
                Ksum = exp( 2*self.Rg2(Mi[i],i)/self.column.D/self.a[i] *
                            self.dedF[i] * (self.Fcr[i] - F) )
        else:
            for i in range(self.f):
                Ksum = exp(self.A[i]*Mi[i]/self.column.Fpr*(self.Fcr[i]-F))
    elif self.model in [2]: # General solution
        g = [2.0*self.Rg(Mi[i],i)/self.column.D for i in range(self.f)]
        lmbda = [ float(self.dedF[i]*(F-self.Fcr[i]) *
                    (self.column.D/2/self.a[i])) for i in range(self.f) ]
                Ksum = K_star_gen(g, lmbda, self.f)
            else:
                print "Model %i not found in star block-copolymer class"%self.model
                sys.exit(1)
            if Ksum > Kmax: Ksum = Kmax
            return Ksum

        def dtdx(self, t, x, M):
            # equation to integrate to find tg
            return self.column.t0 + self.K(t, x, M)*self.column.tP

class star_functional:
    def __init__(self, columnObj):
        # bookkeeping
        self.model = 2
        self.name = "PEG star"
        self.column = columnObj
        self.MW = [3600]
        self.tgSoln = []
        # model parameters
        self.f = 4 # number of arms
        self.A = [8.5e-5,8.5e-5,8.5e-5,8.5e-5]
        self.Mi = [1.0, 1.0, 1.0, 1.0]
          # Distribution of g^2 per arm. Will be normallized.
        self.q = [1.0, 0.0, 0.0, 1.0]
        self.Fcr = [40.7, 40.7, 40.7, 40.7]
        self.a = [0.1,0.1,0.1,0.1]
        self.b = [1.0,0.1,0.1,0.1]
        self.s = [0.5,0.5,0.5,0.5]
        self.dedF = [8.5e-6,8.5e-6,8.5e-6,8.5e-6]
        # plotting parameters
        self.color = None
        self.style = "-"
        self.marker = 'None'
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    def arms(self,arms):
        # resizes length of lists to specificed arms
        # overwrites existing values with first value!!
        self.f = arms
        # convert any existing lists to floats.
        # Assumes its already a float if not a list.
        if isinstance(self.A,list): self.A = self.A[0]
        if isinstance(self.Mi,list): self.Mi = self.Mi[0]
        if isinstance(self.q,list): self.q = self.q[0]
        if isinstance(self.Fcr,list): self.Fcr = self.Fcr[0]
        if isinstance(self.a,list): self.a = self.a[0]
        if isinstance(self.b,list): self.b = self.b[0]
        if isinstance(self.s,list): self.s = self.s[0]
        if isinstance(self.dedF,list): self.dedF = self.dedF[0]
        # Now set the list to proper size
        self.A = [self.A for i in range(arms)]
        self.q = [self.q for i in range(arms)]
        self.Mi = [self.Mi for i in range(arms)]
        self.Fcr = [self.Fcr for i in range(arms)]
        self.a = [self.a for i in range(arms)]
        self.b = [self.b for i in range(arms)]
        self.s = [self.s for i in range(arms)]
        self.dedF = [self.dedF for i in range(arms)]

    def calcA(self):
        # calculate the fitting parameter A from other parameters
        self.A = [ 2*self.dedF[i]*self.column.Fpr/self.column.D/self.a[i]
                   for i in range(self.f) ]
        print "A = ", self.A

    def normalize(self):
        self.Mi = array( self.Mi,dtype=float32)**2 /
                         sum(array(self.Mi,dtype=float32)**2 )
        print "Normalized MW per arm, ", self.Mi

    def Rg(self,M,f):
        return self.b[f] * M**(self.s[f])

    def Rg2(self,M,f):
        return self.Rg(M,f)**2

    def info(self):
        # prints information about this instance
        print "Functionalized star class -- %s"%self.name
        print " Model: %s (model=%i)"%(models[self.model],self.model)
        print " Column: %s (column=columnObject)"%(self.column.name)
        print " Scaling, R_g = b*M^s: b = %s, s = %s" %
                (list2str(self.b), list2str(self.s))
        print " Critical composition: %s %% (Fcr=[%s])" %
                (list2str(self.Fcr),list2str(self.Fcr))
        print " Combined parameter: %s 1/Da (A=[%s])" %
                (list2str(self.A),list2str(self.A))
        print " Interaction energy per composition: %s kT/%% (dedF=[%s])" %
                (list2str(self.dedF),list2str(self.dedF))
        print " Interaction distance: %s nm (a = [%s])" %
                (list2str(self.a),list2str(self.a))
        print " Functional parameter: %s (q = [%s])" %
                (list2str(self.q),list2str(self.q))

    def tg(self, M):
        # solve for t_g for a certain MW
        if self.column.dFdt == 0: # isocratic case
            tg = self.column.t0 + self.column.tP * self.K(1.0, 1.0, M)
            return tg
        if self.model in [0]:
            print "No analytical expression exists for functionalized polymers."
            sys.exit(1)
        if self.model in [1,2,3]:
            xrange = linspace(0, 1, grid)
            return odeint(self.dtdx, 0, xrange, args=(M,))[-1][0]

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.Fi + self.column.dFdt*(tg -
                (self.column.t0+self.column.tP))
    def K(self, t, x, M):
        F = min(100, self.column.Fi + self.column.dFdt*(t -
                (self.column.t0 + self.column.tP)*x))
        # current composition
        Mi = M*array(self.Mi,dtype=float32)
        if self.model in [1]: # Near critical/narrow pore approximation
            if self.column.Fpr == 0:
                Kprod = 1
                for i in range(self.f):
                    Kprod *= exp(2*self.Rg(Mi[i],i)**2/self.column.D/self.a[i] *
                                  self.dedF[i]*(self.Fcr[i] - F))*(1+self.q[i] *
                    (1-(self.Fcr[i]-F)*self.dedF[i] * self.column.D/6/self.a[i]))
            else:
                Kprod = 1
                for i in range(self.f):
                    Kprod *= exp(self.A[i]*M/self.column.Fpr*(self.Fcr[i]-F)) *
                                 (1+self.q[i]*(1-(self.Fcr[i]-F)*self.dedF[i] *
                                  self.column.D/6/self.a[i]))

        elif self.model in [2]: # General solution
            Kprod = 1
            for i in range(self.f):
                g = 2.0*self.Rg(Mi[i],i)/self.column.D
                lmbda = float( self.dedF[i]*(F - self.Fcr[i]) *
                               (self.column.D/2/self.a[i]) )
                pa = 0
                if self.q[i] != 0:
                    pa = p_a(g,lmbda)
                Kprod *= (K_general(g,lmbda) + self.q[i]*pa)

        elif self.model in [3]: # General/narrow pore hybrid
            print "Hybrid model not yet implimented for functionalized stars."
            sys.exit(1)

        else:
            print "Model %i not found in functional class" % self.model
            sys.exit(1)
        if Kprod > Kmax: Kprod = Kmax
        return Kprod

    def dtdx(self, t, x, M):
        # equation to integrate to find tg
        return self.column.t0 + self.K(t, x, M)*self.column.tP

class diblock:
    def __init__(self, columnObj):
        # bookkeeping
        self.model = 2
        self.name = "AB copolymer"
        self.column = columnObj
        self.MW = [3600]
        self.tgSoln = []
        # model parameters
        self.f = 2
        self.A = [8.5e-5,8.5e-5]
        self.Mi = [1.0, 1.0] # Distribution of M per arm. Will be normallized.
        self.Fcr = [40.7, 40.7]
        self.a = [0.1,0.1]
        self.b = [1.0,1.0]
        self.s = [0.5,0.5]
        self.dedF = [8.5e-6,8.5e-6]
        # plotting parameters
        self.color = None
        self.style = "-"
        self.marker = 'None'
        self.legend = 1
        print "Created new %s in %s"%(self.name, self.column.name)

    def normalize(self):
        self.Mi = array( self.Mi,dtype=float32)**2 /
                          sum(array(self.Mi,dtype=float32)**2 )
        print "Normalized MW per arm, ", self.Mi

    def calcA(self):
        # calculate the fitting parameter A from other parameters
        self.A = [ 2*self.dedF[i]*self.column.Fpr/self.column.D/self.a[i]
                    for i in range(self.f) ]
        print "A = ", self.A

    def Rg(self,M,f=2):
        return self.b[f] * M**(self.s[f])

    def Rg2(self,M,f=2):
        return self.Rg(M,f)**2

    def info(self):
        # prints information about this instance
        print "Diblock copolymer -- %s" % self.name
        print " Model: %s (model=%i)" % (models[self.model],self.model)
        print " Column: %s (column=columnObject)" % (self.column.name)
        print " Scaling, R_g = b*M^s: b = %s, s = %s" %
                (list2str(self.b), list2str(self.s))
        print " Critical composition: %s %% (Fcr=[%s])" %
                (list2str(self.Fcr), list2str(self.Fcr))
        print " Combined parameter: %s 1/Da (A=[%s])" %
                (list2str(self.A),list2str(self.A))
        print " Interaction energy per composition: %6s kT/%% (dedF=[%s])" %
                (list2str(self.dedF),list2str(self.dedF))
        print " Interaction distance: %s nm (a = [%s])" %
                (list2str(self.a),list2str(self.a))

    def tg(self, M, discard=True):
        # solve for t_g for a certain MW
        if self.column.dFdt == 0: # isocratic case
            tg = self.column.t0 + self.column.tP * self.K(1.0, 1.0, M)
            return tg
        if self.model in [0]:
            print "No analytical expression exists for copolymers."
            sys.exit(1)
        if self.model in [1,2,3]:
            xrange = linspace(0, 1, grid)
            odeRes = odeint(self.dtdx, 0, xrange, args=(M,))
        if discard:
            return odeRes[-1][0]
        else:
            return odeRes

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.Fi + self.column.dFdt*(tg -
                (self.column.t0+self.column.tP))

    def K(self, t, x, M):
        F = min( 100,self.column.Fi + self.column.dFdt*(t -
                  (self.column.t0 + self.column.tP)*x) )
        # current composition
        Mi = array(self.Mi,dtype=float32)*M
        if self.model in [1]: # Near critical/narrow pore approximation
            if self.column.Fpr == 0:
                Ksum = exp( sum([2*self.Rg(Mi[i],i)**2/self.column.D/self.a[i] *
                             self.dedF[i]*(self.Fcr[i] - F)
                             for i in range(self.f)]) )
            else:
                Ksum = exp( sum([self.A[i]*Mi[i]/self.column.Fpr*(self.Fcr[i]-F)
                             for i in range(self.f)]) )

        elif self.model in [2]: # General solution
            gi = [2.0*self.Rg(Mi[i],i)/self.column.D for i in range(self.f)]
            print "gi=",gi
            li = [float(self.dedF[i]*(F-self.Fcr[i])*(self.column.D/2/self.a[i]))
                    for i in range(self.f)]
            Ksum = K_copolymer(gi,li)
            if debug: print t,x,M,li,gi,Ksum

        else:
            print "Model %i not found in diblock copolymer class" % self.model
            sys.exit(1)
        if Ksum > Kmax: Ksum = Kmax
        return Ksum

    def dtdx(self, t, x, M):
        # equation to integrate to find tg
        return self.column.t0 + self.K(t, x, M)*self.column.tP

class column:
    def __init__(self):
        self.t0 = 0.5 # min
        self.tP = 0.5 # min
        self.Fi = [0] # %
        self.FL = [100] # %
        self.tF = [10.] # min
        self.flow = 1 #mL/min
        self.D = 10 #nm
        self.dFdt = [(self.FL[-1] - self.Fi[-1])/self.tF[-1]]
        self.Fpr = self.dFdt[-1]*self.tP
        self.name = "Nova-Pak Silica"
        self.solvent = "THF"
        print "Created new column %s"%self.name

    def info(self):
        print "Column %s"%self.name
        print " Void volume: %5.2f (t0=%5.2f)"%(self.t0*self.flow, self.t0)
        print " Pore volume: %5.2f (tP=%5.2f)"%(self.tP*self.flow, self.tP)
        print " Flow rate: %5.2f mL/min (flow=%5.2f)"%(self.flow,self.flow)
        print " Injection concentration: %5.2f %% (Fi=%5.2f)"%(self.Fi,self.Fi)
        print " Final concentration: %5.2f %% (FL=%5.2f)"%(self.FL,self.FL)
        print " Time of gradient run: %5.1f min (tF=%5.1f)"%(self.tF,self.tF)
        print " Mean pore diameter: %4.1f nm (D=%4.1f)"%(self.D,self.D)
        print " Solvent name: %s (solvent=’%s’)"%(self.solvent,self.solvent)

    def gradient(self, Fi, FL, tF):
        # add a linear gradient profile
        self.tF.append(tF) # min
        self.Fi.append(Fi)
        self.FL.append(FL)
        self.dFdt.append((FL - Fi)/tF)
        self.Fpr.append(self.dFdt[-1]*self.tP)
        print "Added gradient rate of %5.2f from %i to %i for column %s" %
                (self.dFdt[-1],Fi, FL, self.name)

    def clearGradients(self):
        self.tF = []
        self.Fi = []
        self.FL = []
        self.dFdt = []
        self.Fpr = []

    def currentFg(self,tg,x):
        running = 0
        last_tF = 0
        for i,t in enumerate(self.tF):
            running += t
            if (tg <= running):
                F = self.Fi[i] + self.dFdt[i]*((tg-last_tF) - x*(self.t0+self.tP))
                return F
            last_tF = t
        # tg > tF, return last given concentration
        return self.FL[-1]

    def return_Fg_list(self,tg):
        results = []
        for ts in tg:
            running = 0
            last_tF = 0
            thisF = -99
            for i,t in enumerate(self.tF):
                running += t
                if (ts <= running):
                    F = self.Fi[i]+self.dFdt[i]*((ts-last_tF)-(self.t0+self.tP))
                    thisF = F
                    break
                last_tF = t
            # tg > tF, return last given concentration
            if thisF == -99:
                thisF = self.FL[-1]
            # append results to array
            results.append(thisF)
        return results

    def isocratic(self, conc):
        self.Fi = self.FL = conc
        self.dFdt = self.Fpr = 0
        print "Set column %s as isocratic with Phi=%5.2f%%" %
                (self.name,self.Fi)

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.Fi + self.dFdt*(tg - (self.t0+self.tP))

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

class experdata:
    # Input of experimental data
    def __init__(self, columnObj, filename=None):
        self.column = columnObj
        self.b = 1.0
        self.s = 0.5
        self.color = ’black’
        self.style = ’’
        self.marker = ’o’
        self.name = "Experimental data"
        if filename == None:
            print "Type MW [space] Phi_g [Enter]. Second [Enter] ends input mode."
            rawInp = "999,111"
            self.ExpM = []; self.ExpF = [];
            while rawInp is not "":
                rawInp = raw_input(">")
                rawDat = rawInp.split()
                self.ExpM.append(float(rawDat[0]))
                self.ExpF.append(float(rawDat[1]))
        else:
            rawDat = loadtxt(filename)
            self.ExpM = rawDat[:,0]
            self.ExpF = rawDat[:,1]
        self.MW = self.ExpM
        self.tgSoln = self.tg(self.ExpF)
        print "Created new %s"%self.name

    def RSS(self,polymerObj):
        # Calculate the residual sum of squares
        #~ aCopy = copy.deepcopy(polymerObj)
        #~ aCopy.MW = self.ExpM
        #~ solver([aCopy], silent=True)
        #~ RSS = 0
        #~ for i in range(len(self.ExpM)):
            #~ Fg = aCopy.Fg(aCopy.tgSoln[i])
            #~ RSS += (self.ExpF[i] - Fg)**2
        #~ print "Calculated RSS = ", RSS
        #~ return RSS
        return 0


    def info():
        print "Experimental data -- %s" % self.name
        print " Scaling, R_g = b*M^s: b = %5.2f, s = %5.2f"%(self.b, self.s)

    def Rg(self,M):
        return self.b * M**(self.s)
    def Rg2(self,M):
        return self.Rg(M)**2

    def Vg(self, tg):
        # return V_g from inputed t_g
        return self.column.F * tg

    def Fg(self, tg):
        # return Phi_G from inputed t_g
        return self.column.Fi + self.column.dFdt*(tg -
                (self.column.t0+self.column.tP))

    def tg(self, Fg):
        return (Fg - self.column.Fi)/self.column.dFdt + self.column.t0 +
                 self.column.tP


# OUTPUT CLASSES:

# OUTPUT CLASSES:
class plotter:
    # general plotter: plots solution to integration in various units
    # At declaration, polymerObjs is a list of polymer objects to
    #   plot at the same time
    labels = {"M":r'$M$',"F":r'$\Phi_g$',"Rg":r'$R_g$',"Rg2":r'$R^2_g$',
              "V":r'$V_g$',"t":r'$t_g$',"g":r'$g$'}
    def __init__(self,polymerObjs):
        self.xunit = "M" # M, g, Rg or Rg2
        self.yunit = "F" # F, V or t
        self.xlimits = None
        self.ylimits = None
        self.type = "semilogx"
        self.polymerObjs = polymerObjs
        self.legend = True
        self.legendText = None
        self.legendtitle = None

    def draw(self):
        fig = plot.figure(1)
        axes = fig.add_subplot(111)
        for polymer in self.polymerObjs:
            plotColor = ""; plotStyle = ""
            if polymer.color: plotColor = ",color='"+polymer.color+"'"
            if polymer.style: plotStyle = ",ls='"+polymer.style+"'"
            if polymer.marker: plotStyle = ",marker='"+polymer.marker+"'"
            if self.xunit == "Rg":
                xaxis = "polymer.Rg(polymer.MW)"
            if self.xunit == "g":
                xaxis = "polymer.Rg(polymer.MW)/polymer.column.D"
            elif self.xunit == "Rg2":
                xaxis = "polymer.Rg2(polymer.MW)"
            elif self.xunit == "M":
                xaxis = "polymer.MW"
            else:
                print "Unrecognized plotter.xunit: %s" % self.xunit
                return 1
            if self.yunit == "F":
                yaxis = "polymer.column.return_Fg_list(polymer.tgSoln)"
            elif self.yunit == "V":
                yaxis = "polymer.Vg(polymer.tgSoln)"
            elif self.yunit == "t" or self.yunit == "tg" :
                yaxis = "polymer.tgSoln"
            else:
                print "Unrecognized plotter.yunit: %s" % self.yunit
                return 1
            if isinstance(polymer,experdata):
                plottype = "scatter"
            else:
                plottype = self.type
            plotting = "axes." + plottype + "("+xaxis+","+yaxis+
                         ",label=polymer.name" + plotColor + plotStyle + ")"
            exec plotting

        if self.xlimits: axes.set_xlim(self.xlimits)
        if self.ylimits: axes.set_ylim(self.ylimits)
        axes.set_xlabel(self.labels[self.xunit])
        axes.set_ylabel(self.labels[self.yunit])
        if self.legend: axes.legend(loc=0)
        if self.legendText: axes.legend(self.legendText,loc=0)
        if self.legendtitle: axes.legend(loc=0, title=self.legendtitle)
        plot.draw()

    def usetex():
        rc(’text’, usetex=True)
        params  = {’axes.labelsize’: 15,
                   ’text.fontsize’: 15,
                   ’xtick.labelsize’: 15,
                   ’ytick.labelsize’: 15,
                   ’legend.pad’: 0.2,
                   ’legend.fontsize’: 13,
                   ’lines.markersize’: 3,
                   ’font.size’: 18,
                   ’font.family’:’serif’,
                   ’font.weight’:’bolder’,
                   ’font.stretch’:’expanded’}
        pub  = {’axes.labelsize’: 10,
                   ’text.fontsize’: 10,
                   ’xtick.labelsize’: 10,
                   ’ytick.labelsize’: 10,
                   ’legend.pad’: 0.2,
                   ’legend.fontsize’: 10,
                   ’lines.markersize’: 3,
                   ’font.size’: 10,
                   ’font.family’:’serif’,
                   ’font.weight’:’bolder’,
                   ’font.stretch’:’expanded’}
        mpl.rcParams.update(params)

    def show(self):
        plot.show()


def fitter(polymerObj, expdataObj, fitVar =’ dedf’, fitRange = [0, 1],
            fitIter = 100, fitTol = 1e-3):
    # fit data to parameter. Could take a while depending on model choice
    fRzero = [min(fitRange),max(fitRange)] - average(fitRange)
    print fRzero
    fitTol = log10(min(fitRange))*fitTol
```

```
# save old MW stored in class
oldMW = polymerObj.MW
polymerObj.MW = expdataObj.MW

lastR2 = 0.0
bestR2 = 0.0
converged = False
oldVar = polymerObj.dedF
newVar = oldVar
bestdedF = oldVar
for i in range(fitIter):
    solver(polymerObj, silent=True)
    # calculate the error
    SSerr = 0
    SStot = 0
    for j in range(len(expdataObj.ExpM)):
        Fg = polymerObj.Fg(polymerObj.tgSoln[j])
        SStot += (expdataObj.ExpF[j] - polymerObj.Fcr)**2
        SSerr += (expdataObj.ExpF[j] - Fg)**2
    R2 = 1.0 - SSerr/SStot

    # set a floor value for R2
    if R2 < 0:
        R2 = 0
        oldVar = average(fitRange) # No fit, try random guess

    if (R2 > bestR2):
        bestR2 = R2
        bestdedF = oldVar
    else:
        oldVar = bestdedF

    # generate a new value to test
    newVar = oldVar + (1.0-R2)*random.uniform(fRzero[0],fRzero[1])
    while newVar < min(fitRange) and newVar > max(fitRange):
        newVar = oldVar + (1.0-R2)*random.uniform(fRzero[0],fRzero[1])

    if abs(oldVar-newVar) <= fitTol and R2 >= lastR2:
        converged = True
        break

    if (lastR2 < R2):
        oldVar = newVar
        lastR2 = R2

    polymerObj.dedF = newVar
    print "R2, dedF = ",R2,newVar

if (converged):
    print "After %i iterations, Converged with R2 = %5.3f"%(fitIter, R2)
else:
    R2 = bestR2
    polymerObj.dedF = bestdedF
    print "After %i iterations, did not converge. Using best R2 = %5.3f" %
            (fitIter, R2)
print "new dedF = ", polymerObj.dedF
polymerObj.calcA(silent=True)
polymerObj.MW = oldMW
```

```
class chromatogram:
    # virtual chromatogram, plots predicted retension from a 'sample' of polymers
    pass

class optimizer:
    pass

class overcolumn:
    # plots various information for a single MW, over the length of the column
    def __init__(self, polymerObj):
        self.polymer = polymerObj
        if isinstance(self.polymer.MW, list):
            self.MW = self.polymer.MW[-1]
            # take the last MW in list, if list is specified
        else:
            self.MW = self.polymer.MW
        # K, lambda, Phi,
        self.y1 = "K"
        self.y2 = "lambda"
    def F(self,t,x):
        return self.polymer.column.Fi + self.polymer.column.dFdt*(t -
                x*(self.polymer.column.t0 + self.polymer.column.tP))
    def show(self):
        soln = self.polymer.tg(self.MW,False)
        xgrid = linspace(0,1,grid)
        #deriv = [dtdx(t_g[i], x, MW, Phi_i, grad) for i,x in enumerate(xrange)]
        Kval = [self.polymer.K(soln[t][0], x, self.MW) for t,x in enumerate(xgrid)]
        F = [self.F(soln[t][0],x) for t,x in enumerate(xgrid)]
        eps = [[self.polymer.dedF[i]*(self.F(soln[t][0],x) - self.polymer.Fcr[i])
                    for i in range(self.polymer.f)] for t,x in enumerate(xgrid)]
        # plot using the following parameters
        fig = plot.figure()
        axes1 = fig.add_subplot(111)
        plot.ylabel(r'$\Phi$')
        plot.xlabel(r'$x$')
        axes2 = axes1.twinx()
        plot.ylabel(r'$(\epsilon-\epsilon_{cr})/kT$')
        axes1.plot(xgrid, F, label=r'$\Phi_g(t,x)$', color='red')
        #axes1.set_ylim(0,10)
        #axes1.plot(xrange, deriv, label=r'£dt/dx£')
        axes2.plot(xgrid, eps)
        #axes2.plot(xrange, Kgen, color='green', label='K (Gen)')
        axes1.legend(loc=6)
        axes2.legend(('A','B'),loc=5)
        plot.show()

def solver(listOfPolymers, silent=False):
    if not isinstance(listOfPolymers, list):
        listOfPolymers = [listOfPolymers]

    # updates the solution to t_g(M) for each polymer object in list
    for polymer in listOfPolymers:
        if not silent:
            print "Solving %i points for %s"%(len(polymer.MW),polymer.name)
        polymer.normalize()
        polymer.tgSoln = array([],dtype=float32)
        for i,M in enumerate(polymer.MW):
            polymer.tgSoln = append(polymer.tgSoln,polymer.tg(M))
    if not silent:
        print "Done."


def tabulator(polymerObjs):
    # prints solved info to terminal
    for polymer in polymerObjs:
        print polymer.name + " in " + polymer.column.name
        print "%10s %11s %11s %11s"%("MW","R_g,i","t_g","Phi_g")
        for i,M in enumerate(polymer.MW):
            print "%10.3g  %s  %10g  %10g"%(M,list2str([polymer.Rg(polymer.Mi[j],j)
             for j in range(len(polymer.Mi))]),polymer.tgSoln[i],
             polymer.Fg(polymer.tgSoln[i]))


def plotpartition(listOfKfuns,listOfLambdas):
    # plot the partition function, in terms of g=R/d and lambda=D/2a * (e-e_cr)/kT
    # Not really developed
    grange = linspace(0,3,100) # g = R/d
    lmbda = listOfLambdas
    colors = {0:"blue",1:"red",2:"green"}
    Kfuns = {0:"K_nearcrit",1:"K_general"}#,2:"K_AB",3:"K_star_gen"}
    for i,model in enumerate(listOfKfuns):
        thisColor = colors[i]
        for l in listOfLambdas:
            Kplot = []
            for g in grange:
                exec("Kplot.append("+Kfuns[model]+"("+repr(g)+","+repr(l)+"))")
            plot.plot(grange,Kplot,color=thisColor)
    plot.xlabel(r’$g=R_g/d$’)
    plot.ylabel(r’$K$’)
    plot.show()


def list2str(listIn, roundoff=2):
    formatStr = "%"+str(3+roundoff)+"."+str(roundoff)+"g"
    listIn = list(listIn)
    string = str(formatStr%listIn[0])
    for each in listIn[1:]:
        string = string + "," + str(formatStr%each)
    return string


if __name__ == ’__main__’:
    print "To use gradientInt.py, load it in an interactive enviroment or at ",
            "the start of a new script."
```

Curriculum Vita

Christopher J. Rasmussen

Education

Doctor of Philosophy in Chemical Engineering October, 2012
Rutgers, The State University of New Jersey, Piscataway, NJ

Bachelor of Science in Chemical Engineering May, 2006
Rutgers, The State University of New Jersey, Piscataway, NJ

Publications

1. C. J. Rasmussen and A. V. Neimark, "Calculation of chemical potential of polymers in molecular Monte Carlo simulations," *Molecular Simulation*, vol. in preparation, 2012
2. Y. Brun, B. McCauley, C. J. Rasmussen, and A. V. Neimark, "Optimization of gradient separation of complex polymers using statistical theory of interaction polymer chromatography," *Journal of Chromatography A*, vol. in preparation, 2012
3. C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, "Translocation dynamics of freely-jointed Lennard-Jones chains through a small opening into adsorbing pore," *Journal of Chemical Physics*, vol. in press, 2012
4. G. Y. Gor, C. J. Rasmussen, and A. V. Neimark, "Capillary condensation hysteresis in overlapping spherical pores: A Monte Carlo simulation study," *Langmuir*, vol. 28, pp. 12100–12107, Aug. 2012
5. C. J. Rasmussen, G. Y. Gor, and A. V. Neimark, "Monte Carlo simulation of cavitation in pores with nonwetting defects," *Langmuir*, vol. 28, no. 10, pp. 4702–4711, 2012
6. C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, "Calculation of chemical potentials of chain molecules by the incremental gauge cell method," *Journal of Chemical Physics*, vol. 135, p. 214109, Dec. 2011
7. C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, "Monte Carlo simulation of polymer adsorption," *Adsorption*, vol. 17, no. 1, pp. 265–271, 2011
8. C. J. Rasmussen, A. Vishnyakov, M. Thommes, B. M. Smarsly, F. Kleitz, and A. V. Neimark, "Cavitation in metastable liquid nitrogen confined to nanoscale pores," *Langmuir*, vol. 26, pp. 10147–10157, June 2010

Bibliography

[1] C. J. Rasmussen, A. Vishnyakov, M. Thommes, B. M. Smarsly, F. Kleitz, and A. V. Neimark, "Cavitation in metastable liquid nitrogen confined to nanoscale pores," *Langmuir*, vol. 26, pp. 10147–10157, June 2010.

[2] C. J. Rasmussen, G. Y. Gor, and A. V. Neimark, "Monte Carlo simulation of cavitation in pores with nonwetting defects," *Langmuir*, vol. 28, no. 10, pp. 4702–4711, 2012.

[3] G. Y. Gor, C. J. Rasmussen, and A. V. Neimark, "Capillary condensation hysteresis in overlapping spherical pores: A Monte Carlo simulation study," *Langmuir*, vol. 28, pp. 12100–12107, Aug. 2012.

[4] Y. Brun, B. McCauley, C. J. Rasmussen, and A. V. Neimark, "Optimization of gradient separation of complex polymers using statistical theory of interaction polymer chromatography," *Journal of Chromatography A*, vol. in preparation, 2012.

[5] T. Young, "An essay on the cohesion of fluids," *Philosophical Transactions of the Royal Society of London*, vol. 95, pp. 65–87, Jan. 1805.

[6] P. S. Laplace, *Traité de mécanique céleste. Supplément au Livre X*, vol. 4. Paris: Gauthier-Villars, 1805.

[7] S. K. Kumar and R. Krishnamoorti, "Nanocomposites: Structure, phase behavior, and properties," *Annual Review of Chemical and Biomolecular Engineering*, vol. 1, no. 1, pp. 37–58, 2010.

[8] B. Widom, "Some topics in the theory of fluids," *Journal of Chemical Physics*, vol. 39, pp. 2808–2812, Dec. 1963.

[9] A. V. Neimark and A. Vishnyakov, "Gauge cell method for simulation studies of phase transitions in confined systems," *Physical Review E*, vol. 62, p. 4611, Oct. 2000.

[10] A. V. Neimark and A. Vishnyakov, "A simulation method for the calculation of chemical potentials in small, inhomogeneous, and dense systems," *Journal of Chemical Physics*, vol. 122, pp. 234108–11, June 2005.

[11] S. K. Kumar, I. Szleifer, and A. Z. Panagiotopoulos, "Determination of the chemical potentials of polymeric systems from Monte Carlo simulations," *Physical Review Letters*, vol. 66, p. 2935, June 1991.

[12] H. Pasch and B. Trathnigg, *HPLC of polymers.* Berlin: Springer-Verlag, 1999.

[13] Y. Brun and P. Alden, "Gradient separation of polymers at critical point of adsorption," *Journal of Chromatography A*, vol. 966, pp. 25–40, Aug. 2002.

[14] Y. Brun, "The mechanism of copolymer retention in interactive polymer chromatography. i. critical point of adsorption for statistical copolymers.," *Journal of Liquid Chromatography & Related Technologies*, vol. 22, p. 3027, Dec. 1999.

[15] Y. Brun, "The mechanism of copolymer retention in interactive polymer chromatography. II. gradient separation.," *Journal of Liquid Chromatography & Related Technologies*, vol. 22, p. 3067, Dec. 1999.

[16] R. Zsigmondy *Z. anorg. Chem.*, vol. 71, p. 356, 1911.

[17] E. O. Kraemer, *Treatise on Physical Chemistry.* New York: Van Nostrand, 1931.

[18] S. J. Gregg and K. S. W. Sing, *Adsorption, Surface Area and Porosity.* New York: Academic Press, 1982.

[19] F. Rouquerol, J. Rouquerol, and K. S. W. Sing, *Adsorption by Powders and Porous Solids: Methodology and Applications.* San Diego, CA: Academic Press, 1999.

[20] D. H. Everett, *The Solid-Gas Interface.* New York: Dekker, 1967.

[21] A. V. Neimark, P. I. Ravikovitch, and A. Vishnyakov, "Adsorption hysteresis in nanopores," *Physical Review E*, vol. 62, pp. 1493–1496, Aug. 2000.

[22] K. Morishige, "Adsorption hysteresis in ordered mesoporous silicas," *Adsorption*, vol. 14, pp. 157–163, June 2008.

[23] P. I. Ravikovitch, G. L. Haller, and A. V. Neimark, "Density functional theory model for calculating pore size distributions: pore structure of nanoporous catalysts," *Advances in Colloid and Interface Science*, vol. 76-77, pp. 203–226, July 1998.

[24] J. S. Beck, J. C. Vartuli, W. J. Roth, M. E. Leonowicz, C. T. Kresge, K. D. Schmitt, C. T. W. Chu, D. H. Olson, E. W. Sheppard, S. B. McCullen, J. B. Higgins, and J. L. Schlenker, "A new family of mesoporous molecular sieves prepared with liquid crystal templates," *Jounral of the American Chemical Society*, vol. 114, pp. 10834–43, Dec. 1992.

[25] D. Zhao, J. Feng, Q. Huo, N. Melosh, G. H. Fredrickson, B. F. Chmelka, and G. D. Stucky, "Triblock copolymer syntheses of mesoporous silica with periodic 50 to 300 angstrom pores.," *Science*, vol. 279, p. 548, Jan. 1998.

[26] S. Brunauer, P. H. Emmett, and E. Teller, "Adsorption of gases in multimolecular layers," *Journal of the American Chemical Society*, vol. 60, pp. 309–319, Feb. 1938.

[27] P. I. Ravikovitch and A. V. Neimark, "Density functional theory of adsorption in spherical cavities and pore size characterization of templated nanoporous silicas with cubic and three-dimensional hexagonal structures," *Langmuir*, vol. 18, pp. 1550–1560, Mar. 2002.

[28] P. I. Ravikovitch and A. V. Neimark, "Characterization of nanoporous materials from adsorption and desorption isotherms," *Colloids and Surfaces A*, vol. 187-188, pp. 11–21, 2001.

[29] A. V. Neimark and P. I. Ravikovitch, "Capillary condensation in MMS and pore structure characterization," *Microporous and Mesoporous Materials*, vol. 44-45, pp. 697–707, Apr. 2001.

[30] ISO 15901-3:2007, *Pore size distribution and porosity of solid materials by mercury porosimetry and gas adsorption – Part 3: Analysis of micropores by gas adsorption.* ISO, Geneva, Switzerland.

[31] J. Rouquerol, D. Avnir, C. W. Fairbridge, D. H. Everett, J. M. Haynes, N. Pernicone, J. D. F. Ramsay, K. S. W. Sing, and K. K. Unger, "Recommendations for the characterization of porous solids," *Pure and Applied Chemistry*, vol. 66, no. 8, pp. 1739–1758, 1994.

[32] G. Mason, "The effect of pore space connectivity on the hysteresis of capillary condensation in adsorption–desorption isotherms," *Journal of Colloid and Interface Science*, vol. 88, pp. 36–46, July 1982.

[33] G. C. Wall and R. J. C. Brown, "The determination of pore-size distributions from sorption isotherms and mercury penetration in interconnected pores: The application of percolation theory," *Journal of Colloid and Interface Science*, vol. 82, pp. 141–149, July 1981.

[34] H. Liu, L. Zhang, and N. A. Seaton, "Sorption hysteresis as a probe of pore structure," *Langmuir*, vol. 9, pp. 2576–2582, Oct. 1993.

[35] P. I. Ravikovitch and A. V. Neimark, "Experimental confirmation of different mechanisms of evaporation from ink-bottle type pores: Equilibrium, pore blocking, and cavitation," *Langmuir*, vol. 18, pp. 9830–9837, Dec. 2002.

[36] M. Thommes, B. Smarsly, M. Groenewolt, P. I. Ravikovitch, and A. V. Neimark, "Adsorption hysteresis of nitrogen and argon in pore networks and characterization of novel micro- and mesoporous silicas," *Langmuir*, vol. 22, pp. 756–764, Jan. 2006.

[37] P. J. Flory, *Statistical Mechanics of Chain Molecules*. New York: Oxford University Press, 1988.

[38] M. Rubinstein and R. H. Colby, *Polymer Physics*. New Tork: Oxford University Press, 2003.

[39] W. W. Graessley, R. C. Hayward, and G. S. Grest, "Excluded-volume effects in polymer solutions. 2. comparison of experimental results with numerical simulation data," *Macromolecules*, vol. 32, pp. 3510–3517, May 1999.

[40] P. I. Ravikovitch, A. Vishnyakov, and A. V. Neimark, "Density functional theories and molecular simulations of adsorption and phase transitions in nanopores," *Physical Review E*, vol. 64, p. 011602, 2001.

[41] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*. Academic Press, 2002.

[42] J.-P. Hansen and L. Verlet, "Phase transitions of the Lennard-Jones system," *Physical Review*, vol. 184, no. 1, p. 151, 1969.

[43] G. C. Boulougouris, I. G. Economou, and D. N. Theodorou, "On the calculation of the chemical potential using the particle deletion scheme.," *Molecular Physics*, vol. 96, pp. 905–913, Mar. 1999.

[44] W. C. Swope and H. C. Andersen, "A computer simulation method for the calculation of chemical potentials of liquids and solids using the bicanonical ensemble," *Journal of Chemical Physics*, vol. 102, no. 7, p. 2851, 1995.

[45] C. H. Bennett, "Efficient estimation of free energy differences from Monte Carlo data," *Journal of Computational Physics*, vol. 22, pp. 245–268, Oct. 1976.

[46] A. M. Ferrenberg and R. H. Swendsen, "Optimized Monte Carlo data analysis," *Physical Review Letters*, vol. 63, no. 12, p. 1195, 1989.

[47] G. M. Torrie and J. P. Valleau, "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling," *Journal of Computational Physics*, vol. 23, pp. 187–199, Feb. 1977.

[48] I. Nezbeda and J. A. Kolafa, "A new version of the insertion particle method for determining the chemical potential by Monte Carlo simulation," *Molecular Physics*, vol. 5, pp. 391—403, 1991.

[49] A. Vishnyakov and A. V. Neimark, "Multicomponent gauge cell method," *Journal of Chemical Physics*, vol. 130, pp. 224103–9, June 2009.

[50] A. Neimark and A. Vishnyakov, "Phase transitions and criticality in small systems: Vapor-liquid transition in nanoscale spherical cavities," *Journal of Physical Chemistry B*, vol. 110, pp. 9403–9412, May 2006.

[51] A. Vishnyakov and A. Neimark, "Studies of liquid-vapor equilibria, criticality, and spinodal transitions in nanopores by the gauge cell Monte Carlo simulation method," *Journal of Physical Chemistry B*, vol. 105, pp. 7009–7020, July 2001.

[52] A. Vishnyakov and A. V. Neimark, "Monte Carlo simulation test of pore blocking effects," *Langmuir*, vol. 19, pp. 3240–3247, Apr. 2003.

[53] A. Vishnyakov and A. V. Neimark, "Nucleation of liquid bridges and bubbles in nanoscale capillaries," *Journal of Chemical Physics*, vol. 119, pp. 9755–9764, Nov. 2003.

[54] A. V. Neimark and A. Vishnyakov, "Monte Carlo simulation study of droplet nucleation," *Journal of Chemical Physics*, vol. 122, pp. 174508–11, May 2005.

[55] A. V. Neimark and A. Vishnyakov, "The birth of a bubble: A molecular simulation study.," *Journal of Chemical Physics*, vol. 122, p. 054707, Feb. 2005.

[56] P. Kowalczyk, A. Ciach, and A. V. Neimark, “Adsorption-induced deformation of microporous carbons: Pore size distribution effect,” *Langmuir*, vol. 24, pp. 6603–6608, July 2008.

[57] F. Zheng, X. Zhang, and W. Wang, “Macrophase and microphase separations for surfactants adsorbed on solid surfaces: A gauge cell Monte Carlo study in the lattice model,” *Langmuir*, vol. 24, pp. 4661–4669, May 2008.

[58] M. Jorge and N. A. Seaton, “Long-range interactions in Monte Carlo simulation of confined water,” *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, vol. 100, no. 13, p. 2017, 2002.

[59] J. Jiang and S. I. Sandler, “Capillary phase transitions of linear and branched alkanes in carbon nanotubes from molecular simulation,” *Langmuir*, vol. 22, no. 17, pp. 7391–7399, 2006.

[60] J. Jiang, S. Sandler, and B. Smit, “Capillary phase transitions of n-alkanes in a carbon nanotube,” *Nano Letters*, vol. 4, pp. 241–244, Feb. 2004.

[61] J. Mota and I. Esteves, “Simplified gauge-cell method and its application to the study of capillary phase transition of propane in carbon nanotubes,” *Adsorption*, vol. 13, pp. 21–32, Feb. 2007.

[62] W. L. Jorgensen, J. D. Madura, and C. J. Swenson, “Optimized intermolecular potential functions for liquid hydrocarbons,” *Journal of the American Chemical Society*, vol. 106, no. 22, pp. 6638–6646, 1984.

[63] M. G. Martin and J. I. Siepmann, “Transferable potentials for phase equilibria. 1. united-atom description of n-alkanes,” *Journal of Physical Chemistry B*, vol. 102, pp. 2569–2577, Apr. 1998.

[64] E. F. Casassa, “Equilibrium distribution of flexible polymer chains between a macroscopic solution phase and small voids,” *Journal of Polymer Science Part B: Polymer Letters*, vol. 5, pp. 773–778, Sept. 1967.

[65] M. Muthukumar, “Polymer translocation through a hole.,” *Journal of Chemical Physics*, vol. 111, p. 10371, Dec. 1999.

[66] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, “Testing a new Monte Carlo algorithm for protein folding,” *Proteins: Structure, Function, and Bioinformatics*, vol. 32, no. 1, p. 52–66, 1998.

[67] K. Binder and W. Paul, “Monte Carlo simulations of polymer dynamics: Recent advances,” *Journal of Polymer Science Part B: Polymer Physics*, vol. 35, pp. 1–31, Jan. 1997.

[68] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[69] A. Panagiotopoulos, “Direct determination of phase coexistence properties of fluids by Monte Carlo simulation in a new ensemble,” *Molecular Physics*, vol. 61, pp. 813–826, 1987.

[70] J. J. de Pablo and F. A. Escobedo, “Monte Carlo methods for polymeric systems,” in *Advances in Chemical Physics* (I. Prigogine and S. A. Rice, eds.), p. 337–367, John Wiley & Sons, Inc., 2007.

[71] E. Leontidis, B. M. Forrest, A. H. Widmann, and U. W. Suter, “Monte Carlo algorithms for the atomistic simulation of condensed polymer phases,” *Journal of the Chemical Society, Faraday Transactions*, vol. 91, no. 16, p. 2355, 1995.

[72] D. Frenkel, G. C. A. M. Mooij, and B. Smit, “Novel scheme to study structural and thermal properties of continuously deformable molecules,” *Journal of Physics: Condensed Matter*, vol. 4, pp. 3053–3076, Mar. 1992.

[73] K. G. Honnell, C. K. Hall, and R. Dickman, “On the pressure equation for chain molecules,” *Journal of Chemical Physics*, vol. 87, pp. 664–674, July 1987.

[74] R. Dickman and C. K. Hall, “High density Monte Carlo simulations of chain molecules: Bulk equation of state and density profile near walls,” *Journal of Chemical Physics*, vol. 89, pp. 3168–3174, Sept. 1988.

[75] K. G. Honnell and C. K. Hall, “Exact equations of state for one-dimensional chain fluids,” *Journal of Statistical Physics*, vol. 61, pp. 803–842, Nov. 1990.

[76] K. K. Mon and R. B. Griffiths, “Chemical potential by gradual insertion of a particle in Monte Carlo simulation,” *Physical Review A*, vol. 31, pp. 956–959, Feb. 1985.

[77] M. Müller and W. Paul, “Measuring the chemical potential of polymer solutions and melts in computer simulations,” *Journal of Chemical Physics*, vol. 100, pp. 719–724, Jan. 1994.

[78] I. Carmesin and K. Kremer, “The bond fluctuation method: a new effective algorithm for the dynamics of polymers in all spatial dimensions,” *Macromolecules*, vol. 21, no. 9, pp. 2819–2823, 1988.

[79] N. B. Wilding and M. Müller, “Accurate measurements of the chemical potential of polymeric systems by Monte Carlo simulation,” *Journal of Chemical Physics*, vol. 101, no. 5, p. 4324, 1994.

[80] S. K. Kumar, “The chain length dependence of the chemical potentials of macromolecular systems at zero density: Exact calculations and Monte Carlo simulations,” *Journal of Chemical Physics*, vol. 96, pp. 1490–1497, Jan. 1992.

[81] T. Spyriouni, I. G. Economou, and D. N. Theodorou, “Thermodynamics of chain fluids from atomistic simulation: A test of the chain increment method for chemical potential,” *Macromolecules*, vol. 30, no. 16, pp. 4744–4755, 1997.

[82] D. Frenkel and B. Smit, “Unexpected length dependence of the solubility of chain molecules,” *Molecular Physics*, vol. 75, pp. 983–988, Apr. 1992.

[83] B. Smit, G. C. A. M. Mooij, and D. Frenkel, “Comment on “Determination of the chemical potential of polymeric systems from Monte Carlo simulations”,” *Physical Review Letters*, vol. 68, p. 3657, June 1992.

[84] N. B. Wilding, M. Müller, and K. Binder, “Chain length dependence of the polymer–solvent critical point parameters,” *Journal of Chemical Physics*, vol. 105, p. 802, July 1996.

[85] P. Grassberger, “Pruned-enriched rosenbluth method: Simulations of theta polymers of chain length up to 1 000 000,” *Physical Review E*, vol. 56, no. 3, p. 3682, 1997.

[86] P. Grassberger and R. Hegger, “Simulations of theta-polymers in 2 dimensions,” *Journal De Physique I*, vol. 5, pp. 597–606, May 1995.

[87] L. F. Vega, A. Z. Panagiotopoulos, and K. E. Gubbins, “Chemical potentials and adsorption isotherms of polymers confined between parallel plates,” *Chemical Engineering Science*, vol. 49, pp. 2921–2929, Sept. 1994.

[88] N. Kacker, J. D. Weinhold, and S. K. Kumar, “Thermodynamic properties of a coarse-grained model of hydrocarbon polymers. computer simulations on articulated chain structures,” *Journal of the Chemical Society, Faraday Transactions*, vol. 91, no. 16, p. 2457, 1995.

[89] Y. J. Sheng, A. Z. Panagiotopoulos, S. K. Kumar, and I. Szleifer, “Monte Carlo calculation of phase equilibria for a bead-spring polymeric model,” *Macromolecules*, vol. 27, pp. 400–406, Jan. 1994.

[90] J. J. de Pablo, M. Laso, and U. W. Suter, “Estimation of the chemical potential of chain molecules by simulation,” *Journal of Chemical Physics*, vol. 96, pp. 6157–6162, Apr. 1992.

[91] J. Siepmann and D. Frenkel, “Configurational bias Monte-Carlo - a new sampling scheme for flexible chains,” *Molecular Physics*, vol. 75, no. 1, pp. 59–70, 1992.

[92] M. N. Rosenbluth and A. W. Rosenbluth, “Monte Carlo calculation of the average extension of molecular chains,” *Journal of Chemical Physics*, vol. 23, pp. 356–359, Feb. 1955.

[93] J. I. Siepmann, "A method for the direct calculation of chemical potentials for dense chain systems," *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, vol. 70, no. 6, p. 1145, 1990.

[94] K. Kremer and K. Binder, "Monte Carlo simulation of lattice models for macromolecules," *Computer Physics Reports*, vol. 7, pp. 259–310, June 1988.

[95] J. Batoulis and K. Kremer, "Statistical properties of biased sampling methods for long polymer chains," *Journal of Physics A: Mathematical and General*, vol. 21, pp. 127–146, Jan. 1988.

[96] G. C. A. M. Mooij and D. Frenkel, "The overlapping distribution method to compute chemical potentials of chain molecules," *Journal of Physics: Condensed Matter*, vol. 6, no. 21, pp. 3879–3888, 1994.

[97] S. Consta, N. B. Wilding, D. Frenkel, and Z. Alexandrowicz, "Recoil growth: An efficient simulation method for multi-polymer systems," *Journal of Chemical Physics*, vol. 110, no. 6, p. 3220, 1999.

[98] S. Consta, T. J. H. Vlugt, J. W. Hoeth, B. Smit, and D. Frenkel, "Recoil growth algorithm for chain molecules with continuous interactions," *Molecular Physics*, vol. 97, pp. 1243–1254, Dec. 1999.

[99] B. Smit, "Grand canonical Monte Carlo simulations of chain molecules: adsorption isotherms of alkanes in zeolites," *Molecular Physics*, vol. 85, no. 1, p. 153, 1995.

[100] B. Smit, S. Karaborni, and J. I. Siepmann, "Computer simulations of vapor–liquid phase equilibria of n-alkanes," *Journal of Chemical Physics*, vol. 102, pp. 2126–2140, Feb. 1995.

[101] M. D. Macedonia and E. J. Maginn, "A biased grand canonical Monte Carlo method for simulating adsorption using all-atom and branched united atom models," *Molecular Physics*, vol. 96, pp. 1375–1390, May 1999.

[102] M. Laso, J. J. de Pablo, and U. W. Suter, "Simulation of phase equilibria for chain molecules," *Journal of Chemical Physics*, vol. 97, no. 4, pp. 2817–2819, 1992.

[103] B. L. Severson and R. Q. Snurr, "Monte Carlo simulation of n-alkane adsorption isotherms in carbon slit pores," *Journal of Chemical Physics*, vol. 126, pp. 134708–7, Apr. 2007.

[104] F. Porcheron, B. Rousseau, and A. H. Fuchs, "Structure of ultra-thin confined alkane films from Monte Carlo simulations," *Molecular Physics*, vol. 100, pp. 2109–2119, July 2002.

[105] F. Porcheron, B. Rousseau, M. Schoen, and A. H. Fuchs, "Structure and solvation forces in confined alkane films," *Physical chemistry chemical physics*, vol. 3, no. 7, pp. 1155–1159, 2001.

[106] M. Martin and J. Siepmann, "Novel configurational-bias Monte Carlo method for branched molecules. transferable potentials for phase equilibria. 2. united-atom description of branched alkanes," *Journal of Physical Chemistry B*, vol. 103, pp. 4508–4517, May 1999.

[107] G. Garberoglio, "Boltzmann bias grand canonical Monte Carlo," *Journal of Chemical Physics*, vol. 128, Apr. 2008.

[108] F. Ganazzoli and G. Raffaini, "Computer simulation of polypeptide adsorption on model biomaterials," *Physical Chemistry Chemical Physics*, vol. 7, no. 21, p. 3651, 2005.

[109] F. T. Wall and J. J. Erpenbeck, "New method for the statistical computation of polymer dimensions," *Journal of Chemical Physics*, vol. 30, pp. 634–637, Mar. 1959.

[110] R. Grishman, "Mean square endpoint separation of off-lattice self-avoiding walks," *Journal of Chemical Physics*, vol. 58, pp. 220–225, Jan. 1973.

[111] T. Garel and H. Orland, "Guided replication of random chain: a new Monte Carlo method," *Journal of Physics A: Mathematical and General*, vol. 23, pp. L621–L626, June 1990.

[112] P. G. Higgs and H. Orland, "Scaling behavior of polyelectrolytes and polyampholytes: Simulation by an ensemble growth method," *Journal of Chemical Physics*, vol. 95, pp. 4506–4518, Sept. 1991.

[113] B. Velikson, T. Garel, J. C. Niel, H. Orland, and J. C. Smith, "Conformational distribution of heptaalanine: Analysis using a new Monte Carlo chain growth method," *Journal of Computational Chemistry*, vol. 13, no. 10, p. 1216–1233, 1992.

[114] R. Hegger and P. Grassberger, "Chain polymers near an adsorbing surface," *Journal of Physics A: Mathematical and General*, vol. 27, pp. 4069–4081, June 1994.

[115] P. Grassberger and R. Hegger, "Monte Carlo simulations of off-lattice polymers," *Journal of Physics: Condensed Matter*, vol. 7, no. 16, pp. 3089–3097, 1995.

[116] P. Grassberger, "Recursive sampling of random walks: self-avoiding walks in disordered media," *Journal of Physics A: Mathematical and General*, vol. 26, no. 5, pp. 1023–1036, 1993.

[117] P. Grassberger, "Monte Carlo simulation of 3D self-avoiding walks," *Journal of Physics A: Mathematical and General*, vol. 26, no. 12, pp. 2769–2776, 1993.

[118] P. Grassberger, H. Frauenkron, and W. Nadler, "PERM: a Monte Carlo strategy for simulating polymers and other things," in *Monte Carlo approach to biopolymers and protein folding* (P. Grassberger, G. Barkema, and W. Nadler, eds.), pp. 301–315, Singapore: World Scientific, 1998.

[119] N. Combe, T. J. H. Vlugt, P. R. T. Wolde, and D. Frenkel, "Dynamic pruned-enriched rosenbluth method," *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, vol. 101, no. 11, p. 1675, 2003.

[120] J. Jiang, Y. Huang, and J. Wu, "Can the pruned-enriched method be used for the simulation of fluids?," *Journal of Statistical Physics*, vol. 136, pp. 984–988, Sept. 2009.

[121] H. Frauenkron, U. Bastolla, E. Gerstner, P. Grassberger, and W. Nadler, "New Monte Carlo algorithm for protein folding," *Physical Review Letters*, vol. 80, pp. 3149–3152, Apr. 1998.

[122] H.-P. Hsu and P. Grassberger, "A review of Monte Carlo simulations of polymers with PERM," *Journal of Statistical Physics*, vol. 144, pp. 597–637, July 2011.

[123] F. A. Escobedo and J. J. de Pablo, "Monte Carlo simulation of the chemical potential of polymers in an expanded ensemble," *Journal of Chemical Physics*, vol. 103, no. 7, pp. 2703–2710, 1995.

[124] A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, "New approach to Monte Carlo calculation of the free energy: Method of expanded ensembles," *Journal of Chemical Physics*, vol. 96, pp. 1776–1783, Feb. 1992.

[125] F. A. Escobedo and J. J. de Pablo, "Expanded grand canonical and gibbs ensemble Monte Carlo simulation of polymers.," *Journal of Chemical Physics*, vol. 105, no. 10, p. 4391, 1996.

[126] F. Escobedo, "Simulation of bulk, confined, and polydisperse systems. II. application to chain systems," *Journal of Chemical Physics*, vol. 115, pp. 5653–5661, Sept. 2001.

[127] F. A. Escobedo, "Simulation of bulk, confined, and polydisperse systems. i. a unified methodological framework.," *Journal of Chemical Physics*, vol. 115, no. 12, 2001.

[128] F. A. Escobedo and C. R. A. Abreu, "On the use of transition matrix methods with extended ensembles," *Journal of Chemical Physics*, vol. 124, p. 104110, Mar. 2006.

[129] H. Meirovitch, "Entropy, pressure, and chemical potential of multiple chain systems from computer simulation. i. application of the scanning method," *Journal of Chemical Physics*, vol. 97, p. 5803, Oct. 1992.

[130] H. Meirovitch, "Entropy, pressure, and chemical potential of multiple chain systems from computer simulation. II. application of the metropolis and the hypothetical scanning methods," *Journal of Chemical Physics*, vol. 97, p. 5816, Oct. 1992.

[131] H. Meirovitch, "Simulation of the chemical potential of polymers," *Computational and Theoretical Polymer Science*, vol. 8, no. 1-2, pp. 219–227, 1998.

[132] S. Cheluvaraja and H. Meirovitch, “Calculation of the entropy and free energy of peptides by molecular dynamics simulations using the hypothetical scanning molecular dynamics method,” *Journal of Chemical Physics*, vol. 125, p. 024905, July 2006.

[133] K. Shing and K. Gubbins, “The chemical potential in dense fluids and fluid mixtures via computer simulation,” *Molecular Physics*, vol. 46, no. 5, pp. 1109–1128, 1982.

[134] G. C. Boulougouris, I. G. Economou, and D. N. Theodorou, “Calculation of the chemical potential of chain molecules using the staged particle deletion scheme.,” *Journal of Chemical Physics*, vol. 115, p. 8231, Nov. 2001.

[135] L. R. Dodd and D. N. Theodorou, “Analytical treatment of the volume and surface area of molecules formed by an arbitrary collection of unequal spheres intersected by planes,” *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, vol. 72, no. 6, p. 1313, 1991.

[136] G. P. Lithoxoos, L. D. Peristeras, G. C. Boulougouris, and I. G. Economou, “Monte Carlo simulation of carbon monoxide, carbon dioxide and methane adsorption on activated carbon,” *Molecular Physics*, vol. 110, no. 11-12, pp. 1153–1160, 2012.

[137] J. W. McBain, “An explanation of hysteresis in the hydration and dehydration of gels,” *Journal of the American Chemical Society*, vol. 57, pp. 699–700, Apr. 1935.

[138] L. H. Cohan, “Sorption hysteresis and the vapor pressure of concave surfaces,” *Journal of the American Chemical Society*, vol. 60, pp. 433–435, Feb. 1938.

[139] M. Thommes, “Physical adsorption characterization of order and amorphous mesoporous materials,” in *Nanoporous Materials, Science & Engineering* (G. Q. Lu and X. S. Zhao, eds.), p. 317, Imperial College Press, 2004.

[140] C. T. Kresge, M. E. Leonowicz, W. J. Roth, J. C. Vartuli, and J. S. Beck, “Ordered mesoporous molecular-sieves synthesized by a liquid-crystal template mechanism,” *Nature*, vol. 359, pp. 710–712, Oct. 1992.

[141] A. V. Neimark, “Percolation theory of capillary hysteresis phenomena and its application for characterization of porous solids,” *Studies in Surface Science and Catalysis*, vol. 62, p. 67, 1991.

[142] P. Van Der Voort, P. I. Ravikovitch, K. P. De Jong, M. Benjelloun, E. Van Bavel, A. H. Janssen, A. V. Neimark, B. M. Weckhuysen, and E. F. Vansant, “A new templated ordered structure with combined micro- and mesopores and internal silica nanocapsules,” *Journal of Physical Chemistry B*, vol. 106, pp. 5873–5877, June 2002.

[143] A. V. Neimark *Acad. Sci. USSR (Doklady Akademii Nauk SSSR)*, vol. 273, p. 384, 1983.

[144] A. V. Neimark *Russian J. Phys. Chem. (Z. Fiz. Khim.)*, vol. 60, p. 1745, 1986.

[145] F. R. Young, *Cavitation.* World Scientific Publishing Company, 1 ed., Mar. 1999.

[146] Y. A. Pishchalnikov, O. A. Sapozhnikov, M. R. Bailey, J. C. Williams Jr., R. O. Cleveland, L. A. Crum, A. P. Evan, and J. A. McAteer, “Cavitation bubble cluster activity in the breakage of kidney stones by lithotripter shock waves,” *Journal of Endourology / Endourological Society*, vol. 17, p. 435–446, Sept. 2003.

[147] C. C. Coussios and R. A. Roy, “Applications of acoustics and cavitation to noninvasive therapy and drug delivery,” *Annual Review of Fluid Mechanics*, vol. 40, pp. 395–420, Jan. 2008.

[148] T. Terahara, S. Mitragotri, and R. Langer, “Porous resins as a cavitation enhancer for low-frequency sonophoresis,” *Journal of Pharmaceutical Sciences*, vol. 91, pp. 753–759, Mar. 2002.

[149] S. Eslava, M. R. Baklanov, A. V. Neimark, F. Iacopi, C. E. A. Kirschhock, K. Maex, and J. A. Martens, “Evidence of large voids in pure-silica-zeolite low-k dielectrics synthesized by spin-on of nanoparticle suspensions,” *Advanced Materials*, vol. 20, no. 16, pp. 3110–3116, 2008.

[150] Y. Sakamoto, M. Kaneda, O. Terasaki, D. Y. Zhao, J. M. Kim, G. Stucky, H. J. Shin, and R. Ryoo, "Direct imaging of the pores and cages of three-dimensional mesoporous materials," *Nature*, vol. 408, no. 6811, pp. 449–453, 2000.

[151] A. H. Schoen, *Infinite periodic minimal surfaces without self-intersections*. NASA technical note, NASA TN D-5541, National Aeronautics and Space Administration, 1970.

[152] O. C. Gobin, Y. Wan, D. Zhao, F. Kleitz, and S. Kaliaguine, "Mesostructured silica SBA-16 with tailored intrawall porosity part 1: Synthesis and characterization," *Journal of Physical Chemistry C*, vol. 111, pp. 3053–3058, Feb. 2007.

[153] P. Van Der Voort, M. Benjelloun, and E. F. Vansant, "Rationalization of the synthesis of SBA-16: controlling the micro- and mesoporosity," *Journal of Physical Chemistry B*, vol. 106, pp. 9027–9032, Sept. 2002.

[154] C.-F. Cheng, Y.-C. Lin, H.-H. Cheng, and Y.-C. Chen, "The effect and model of silica concentrations on physical properties and particle sizes of three-dimensional SBA-16 nanoporous materials," *Chemical Physics Letters*, vol. 382, pp. 496–501, Dec. 2003.

[155] F. Kleitz, T. Czuryszkiewicz, L. A. Solovyov, and M. Linden, "X-ray structural modeling and gas adsorption analysis of cagelike SBA-16 silica mesophases prepared in a F127/Butanol/H2O system," *Chemistry of Materials*, vol. 18, pp. 5070–5079, Oct. 2006.

[156] C. Yang, W. Schmidt, and F. Kleitz, "Pore topology control of three-dimensional large pore cubic silica mesophases," *Journal of Materials Chemistry*, vol. 15, no. 48, pp. 5112–5114, 2005.

[157] A. Thomas, H. Schlaad, B. Smarsly, and M. Antonietti, "Replication of lyotropic block copolymer mesophases into porous silica by nanocasting: Learning about finer details of polymer self-assembly," *Langmuir*, vol. 19, pp. 4455–4459, May 2003.

[158] D. Kuang, T. Brezesinski, and B. Smarsly, "Hierarchical porous silica materials with a trimodal pore system using surfactant templates," *Journal of the American Chemical Society*, vol. 126, no. 34, pp. 10534–10535, 2004.

[159] O. Sel, D. Kuang, M. Thommes, and B. Smarsly, "Principles of hierarchical meso- and macropore architectures by liquid crystalline and polymer colloid templating," *Langmuir*, vol. 22, pp. 2311–2322, Feb. 2006.

[160] M. Kuemmel, D. Grosso, C. Boissière, B. Smarsly, T. Brezesinski, P. A. Albouy, H. Amenitsch, and C. Sanchez, "Thermally stable nanocrystalline gamma-alumina layers with highly ordered 3D mesoporosity13," *Angewandte Chemie International Edition*, vol. 44, no. 29, pp. 4589–4592, 2005.

[161] S. Inagaki, S. Guan, Y. Fukushima, T. Ohsuna, and O. Terasaki, "Novel mesoporous materials with a uniform distribution of organic groups and inorganic oxide in their frameworks," *Journal of the American Chemical Society*, vol. 121, pp. 9611–9614, Oct. 1999.

[162] P. Ravikovitch, A. Vishnyakov, R. Russo, and A. Neimark, "Unified approach to pore size characterization of microporous carbonaceous materials from n2, ar, and CO2 adsorption isotherms," *Langmuir*, vol. 16, pp. 2311–2320, Mar. 2000.

[163] P. I. Ravikovitch, D. Wei, W. T. Chueh, G. L. Haller, and A. V. Neimark, "Evaluation of pore structure parameters of MCM-41 catalyst supports and catalysts by means of nitrogen and argon adsorption," *Journal of Physical Chemistry B*, vol. 101, pp. 3671–3679, May 1997.

[164] A. V. Neimark, P. I. Ravikovitch, M. Grün, F. Schüth, and K. K. Unger, "Pore size analysis of MCM-41 type adsorbents by means of nitrogen and argon adsorption,," *Journal of Colloid and Interface Science*, vol. 207, pp. 159–169, Nov. 1998.

[165] J. K. Johnson, J. A. Zollweg, and K. E. Gubbins, "The Lennard-Jones equation of state revisited," *Molecular Physics*, vol. 78, no. 3, pp. 591–618, 1993.

[166] G. E. Norman and V. S. Filinov, "Investigation of phase transitions by the Monte Carlo method," *High Temp. (USSR)*, vol. 7, pp. 216–222, 1969.

[167] A. V. Neimark, P. I. Ravikovitch, and A. Vishnyakov, "Inside the hysteresis loop: Multiplicity of internal states in confined fluids," *Physical Review E*, vol. 65, p. 031505, Mar. 2002. Copyright (C) 2009 The American Physical Society; Please report any problems to prola@aps.org.

[168] M. Thommes, R. Köhn, and M. Fröba, "Sorption and pore condensation behavior of nitrogen, argon, and krypton in mesoporous MCM-48 silica materials," *Journal of Physical Chemistry B*, vol. 104, no. 33, pp. 7932–7943, 2000.

[169] M. Thommes, R. Köhn, and M. Fröba, "Sorption and pore condensation behavior of pure fluids in mesoporous MCM-48 silica, MCM-41 silica, SBA-15 silica and controlled-pore glass at temperatures above and below the bulk triple point," *Applied Surface Science*, vol. 196, pp. 239–249, Aug. 2002.

[170] J. W. Gibbs, *The Scientific Papers of J. Willard Gibbs*. New York: Dover, 1961.

[171] M. Mecke, J. Winkelmann, and J. Fischer, "Molecular dynamics simulation of the liquid–vapor interface: The Lennard-Jones fluid," *Journal of Chemical Physics*, vol. 107, pp. 9264–9270, Dec. 1997.

[172] S. Punnathanam and D. S. Corti, "Homogeneous bubble nucleation in stretched fluids: Cavity formation in the superheated Lennard-Jones liquid," *Industrial & Engineering Chemistry Research*, vol. 41, pp. 1113–1121, Mar. 2002.

[173] V. G. Baidakov, *Explosive Boiling of Superheated Cryogenic Liquid*. Weinheim, Germany: WILEY-VCH GmbH & Co. KGaA, 2007.

[174] C. G. V. Burgess and D. H. Everett, "The lower closure point in adsorption hysteresis of the capillary condensation type," *Journal of Colloid and Interface Science*, vol. 33, pp. 611–614, Aug. 1970.

[175] O. Kadlec and M. M. Dubinin, "Comments on the limits of applicability of the mechanism of capillary condensation," *Journal of Colloid and Interface Science*, vol. 31, pp. 479–489, Dec. 1969.

[176] C. Reichenbach, G. Kalies, D. Enke, and D. Klank, "Cavitation and pore blocking in nanoporous glasses," *Langmuir*, vol. 27, no. 17, pp. 10699–10704, 2011.

[177] A. Grosman and C. Ortega, "Cavitation in metastable fluids confined to linear mesopores," *Langmuir*, vol. 27, pp. 2364–2374, Mar. 2011.

[178] C. Fan, D. D. Do, and D. Nicholson, "On the cavitation and pore blocking in slit-shaped ink-bottle pores," *Langmuir*, vol. 27, pp. 3511–3526, Apr. 2011.

[179] P. A. Monson, "Fluids confined in porous materials: Towards a unified understanding of thermodynamics and dynamics," *Chemie Ingenieur Technik*, vol. 83, pp. 143–151, Jan. 2011.

[180] P. T. M. Nguyen, D. D. Do, and D. Nicholson, "On the cavitation and pore blocking in cylindrical pores with simple connectivity," *Journal of Physical Chemistry B*, vol. 115, no. 42, pp. 12160–12172, 2011.

[181] K. Morishige, M. Tateishi, F. Hirose, and K. Aramaki, "Change in desorption mechanism from pore blocking to cavitation with temperature for nitrogen in ordered silica with cagelike pores," *Langmuir*, vol. 22, pp. 9220–9224, Oct. 2006.

[182] A. Sarkar, S. R. Chaudhuri, S. Wang, F. Kirkbir, and H. Murata, "Drying of alkoxide gels – observation of an alternate phenomenology," *Journal of Sol-Gel Science and Technology*, vol. 2, no. 1-3, pp. 865–870, 1994.

[183] G. W. Scherer and D. M. Smith, "Cavitation during drying of a gel," *Journal of Non-Crystalline Solids*, vol. 189, pp. 197–211, Sept. 1995.

[184] M. Parlar and Y. Yortsos, "Nucleation and pore geometry effects in capillary desorption processes in porous media," *Journal of Colloid and Interface Science*, vol. 132, pp. 425–443, Oct. 1989.

[185] M. Parlar and Y. Yortsos, "Percolation theory of vapor adsorption—desorption processes in porous materials," *Journal of Colloid and Interface Science*, vol. 124, pp. 162–176, July 1988.

[186] J. Esparza, M. Ojeda, A. Campero, G. Hernández, C. Felipe, M. Asomoza, S. Cordero, I. Kornhauser, and F. Rojas, "Development and sorption characterization of some model mesoporous and microporous silica adsorbents," *Journal of Molecular Catalysis A: Chemical*, vol. 228, pp. 97–110, Mar. 2005.

[187] S. Haffer, M. Tiemann, and M. Fröba, "Periodic mesoporous organosilica (PMO) materials with uniform spherical core-shell structure," *Chemistry - A European Journal*, vol. 16, pp. 10447–10452, Sept. 2010.

[188] K. Morishige and M. Ishino, "Lower closure point of adsorption hysteresis in ordered mesoporous silicas," *Langmuir*, vol. 23, no. 22, pp. 11021–11026, 2007.

[189] H.-J. Woo, F. Porcheron, and P. A. Monson, "Modeling desorption of fluids from disordered mesoporous materials," *Langmuir*, vol. 20, pp. 4743–4747, May 2004.

[190] B. Coasne, K. E. Gubbins, and R. J.-M. Pellenq, "A grand canonical Monte Carlo study of adsorption and capillary phenomena in nanopores of various morphologies and topologies: Testing the BET and BJH characterization methods," *Particle & Particle Systems Characterization*, vol. 21, pp. 149–160, Aug. 2004.

[191] B. Libby and P. A. Monson, "Adsorption/Desorption hysteresis in inkbottle pores: A density functional theory and Monte Carlo simulation study," *Langmuir*, vol. 20, no. 10, pp. 4289–4294, 2004.

[192] L. Sarkisov and P. A. Monson, "Hysteresis in Monte Carlo and molecular dynamics simulations of adsorption in porous materials," *Langmuir*, vol. 16, pp. 9857–9860, Dec. 2000.

[193] L. Sarkisov and P. A. Monson, "Modeling of adsorption and desorption in pores of simple geometry using molecular dynamics," *Langmuir*, vol. 17, pp. 7600–7604, Nov. 2001.

[194] B. Coasne, A. Galarneau, F. Di Renzo, and R. J. M. Pellenq, "Effect of morphological defects on gas adsorption in nanoporous silicas," *Journal of Physical Chemistry C*, vol. 111, no. 43, pp. 15759–15770, 2007.

[195] S. Naumov, A. Khokhlov, R. Valiullin, J. Kärger, and P. A. Monson, "Understanding capillary condensation and hysteresis in porous silicon: Network effects within independent pores," *Physical Review E*, vol. 78, p. 060601, Dec. 2008.

[196] D. Wallacher, N. Künzner, D. Kovalev, N. Knorr, and K. Knorr, "Capillary condensation in linear mesopores of different shape," *Physical Review Letters*, vol. 92, p. 195704, May 2004.

[197] B. Coasne, A. Grosman, C. Ortega, and M. Simon, "Adsorption in noninterconnected pores open at one or at both ends: A reconsideration of the origin of the hysteresis phenomenon," *Physical Review Letters*, vol. 88, p. 256102, June 2002.

[198] S. Naumov, R. Valiullin, J. Kärger, and P. A. Monson, "Understanding adsorption and desorption processes in mesoporous materials with independent disordered channels," *Physical Review E*, vol. 80, no. 3, p. 031607, 2009.

[199] P. Kumar, T. Hofmann, K. Knorr, P. Huber, P. Scheib, and P. Lemmens, "Tuning the pore wall morphology of mesoporous silicon from branchy to smooth, tubular by chemical treatment," *Journal of Applied Physics*, vol. 103, p. 024303, Jan. 2008.

[200] P. G. Debenedetti, *Metastable Liquids*. Princeton University Press, Dec. 1996.

[201] J. Frenkel, *Kinetic Theory of Liquids*. Peter Smith Publisher, Incorporated, Jan. 1984.

[202] I. J. Ford, "Statistical mechanics of nucleation: a review," *Proceedings of the Institution of Mechanical Engineers Part C-Journal of Mechanical Engineering Science*, vol. 218, pp. 883–899, Aug. 2004.

[203] R. C. Tolman, "The effect of droplet size on surface tension," *Journal of Chemical Physics*, vol. 17, pp. 333–337, Mar. 1949.

[204] D. Do and H. Do, "Appropriate volumes for adsorption isotherm studies: The absolute void volume, accessible pore volume and enclosing particle volume," *Journal of Colloid and Interface Science*, vol. 316, pp. 317–330, Dec. 2007.

[205] M. S. A. Baksh and R. T. Yang, "Model for spherical cavity radii and potential functions of sorbates in zeolites," *AIChE Journal*, vol. 37, no. 6, pp. 923–930, 1991.

[206] R. Goldman, *Rethinking Quaternions: Theory and Computation.* Morgan & Claypool Publishers, Aug. 2010.

[207] C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, "Monte Carlo simulation of polymer adsorption," *Adsorption*, vol. 17, no. 1, pp. 265–271, 2011.

[208] C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, "Calculation of chemical potentials of chain molecules by the incremental gauge cell method," *Journal of Chemical Physics*, vol. 135, p. 214109, Dec. 2011.

[209] A. V. Neimark and A. Vishnyakov, "Vapor-to-droplet transition in a Lennard-Jones fluid: Simulation study of nucleation barriers using the ghost field method," *Journal of Physical Chemistry B*, vol. 109, pp. 5962–5976, Mar. 2005.

[210] M. Haahr, "Random.org: True random number service."

[211] M. Schoen, C. Rhykerd, J. Cushman, and D. Diestler, "Slit-pore sorption isotherms by the grand-canonical Monte Carlo method," *Molecular Physics*, vol. 66, pp. 1171–1182, Apr. 1989.

[212] J. R. Errington, "Evaluating surface tension using grand-canonical transition-matrix Monte Carlo simulation and finite-size scaling," *Physical Review E*, vol. 67, p. 012102, Jan. 2003.

[213] B. Chen, J. I. Siepmann, K. J. Oh, and M. L. Klein, "Aggregation-volume-bias Monte Carlo simulations of vapor-liquid nucleation barriers for Lennard-Jonesium," *Journal of Chemical Physics*, vol. 115, pp. 10903–10913, Dec. 2001.

[214] A. Luzar and K. Leung, "Dynamics of capillary evaporation. i. effect of morphology of hydrophobic surfaces," *Journal of Chemical Physics*, vol. 113, pp. 5836–5844, Oct. 2000.

[215] H. Cochard, "Cavitation in trees," *C. R. Physique*, vol. 7, pp. 1018—1026, 2006.

[216] D. Or and M. Tuller, "Cavitation during desaturation of porous media under tension," *Water Resources Research*, vol. 38, p. 1061, May 2002.

[217] P. I. Ravikovitch, A. V. Neimark, A. Sayari, and M. Jaroniec, "Calculations of pore size distributions in nanoporous materials from adsorption and desorption isotherms," in *Nanoporous Materials II Proceedings of the 2nd Conference on Access in Nanoporous Materials*, vol. Volume 129, pp. 597–606, Elsevier, 2000.

[218] T. Horikawa, D. Do, and D. Nicholson, "Capillary condensation of adsorbates in porous materials," *Advances in Colloid and Interface Science*, vol. 169, pp. 40–58, Nov. 2011.

[219] P. Monson, "Understanding adsorption/desorption hysteresis for fluids in mesoporous materials using simple molecular models and classical density functional theory," *Microporous and Mesoporous Materials*, vol. 160, pp. 47–66, Sept. 2012.

[220] J. Edison and P. Monson, "Dynamic mean field theory of condensation and evaporation in model pore networks with variations in pore size," *Microporous and Mesoporous Materials*, vol. 154, pp. 7–15, May 2012.

[221] C. J. Gommes, "Adsorption, capillary bridge formation, and cavitation in SBA-15 corrugated mesopores: A Derjaguin–Broekhoff–de boer analysis," *Langmuir*, vol. 28, pp. 5101–5115, Mar. 2012.

[222] J. R. Matos, M. Kruk, L. P. Mercuri, M. Jaroniec, L. Zhao, T. Kamiyama, O. Terasaki, T. J. Pinnavaia, and Y. Liu, "Ordered mesoporous silica with large cage-like pores: Structural identification and pore connectivity design by controlling the synthesis temperature and time," *Journal of the American Chemical Society*, vol. 125, pp. 821–829, Jan. 2003.

[223] W. Fan, M. A. Snyder, S. Kumar, P.-S. Lee, W. C. Yoo, A. V. McCormick, R. Lee Penn, A. Stein, and M. Tsapatsis, "Hierarchical nanofabrication of microporous crystals with ordered mesoporosity," *Nature Materials*, vol. 7, p. 984, Dec. 2008.

[224] G. Y. Gor, M. Thommes, K. A. Cychosz, and A. V. Neimark, "Quenched solid density functional theory method for characterization of mesoporous carbons by nitrogen adsorption," *Carbon*, vol. 50, pp. 1583–1590, Apr. 2012.

[225] K. A. Cychosz, X. Guo, W. Fan, R. Cimino, G. Y. Gor, M. Tsapatsis, A. V. Neimark, and M. Thommes, "Characterization of the pore structure of three-dimensionally ordered mesoporous carbons using high resolution gas sorption," *Langmuir*, vol. in press, Aug. 2012.

[226] A. Wilke and J. Weber, "Mesoporous polymer networks-ultraporous DVB resins by hard-templating of close-packed silica spheres," *Macromolecular Rapid Communications*, vol. 33, pp. 785–790, May 2012. PMID: 22351298.

[227] Z. Liu, L. Herrera, V. T. Nguyen, D. D. Do, and D. Nicholson, "A Monte Carlo scheme based on mid-density in a hysteresis loop to determine equilibrium phase transition," *Molecular Simulation*, vol. 37, pp. 932–939, Sept. 2011.

[228] L. D. Gelb, "Modeling amorphous porous materials and confined fluids," *MRS Bulletin*, vol. 34, no. 08, pp. 592–601, 2009.

[229] F. Casanova, C. E. Chiang, A. M. Ruminski, M. J. Sailor, and I. K. Schuller, "Controlling the role of nanopore morphology in capillary condensation," *Langmuir*, vol. 28, pp. 6832–6838, May 2012.

[230] J. Harris and S. A. Rice, "A lattice model of a supported monolayer of amphiphile molecules: Monte Carlo simulations," *Journal of Chemical Physics*, vol. 88, pp. 1298–1306, Jan. 1988.

[231] G. C. A. M. Mooij and D. Frenkel, "A systematic optimization scheme for configurational bias Monte Carlo," *Molecular Simulation*, vol. 17, no. 1, p. 41, 1996.

[232] S. Kumar, I. Szleifer, and A. Panagiotopoulos, "Kumar, Szleifer, and Panagiotopoulos reply," *Physical Review Letters*, vol. 68, p. 3658, 1992.

[233] S. K. Kumar, "A modified real particle method for the calculation of the chemical potentials of molecular systems," *Journal of Chemical Physics*, vol. 97, no. 5, pp. 3550–3556, 1992.

[234] M. Jorge and N. A. Seaton, "Molecular simulation of phase coexistence in adsorption in porous solids," *Molecular Physics: An International Journal at the Interface Between Chemistry and Physics*, vol. 100, no. 24, p. 3803, 2002.

[235] L. Dodd and D. Theodorou, "Atomistic Monte Carlo simulation and continuum mean field theory of the structure and equation of state properties of alkane and polymer melts," in *Atomistic Modeling of Physical Properties* (L. Monnerie and U. Suter, eds.), vol. 116 of *Advances in Polymer Science*, pp. 249–281, Springer Berlin / Heidelberg, 1994.

[236] J. Gao and J. H. Weiner, "Contribution of covalent bond force to pressure in polymer melts," *Journal of Chemical Physics*, vol. 91, no. 5, pp. 3168–3173, 1989.

[237] F. T. Wall and F. Mandel, "Macromolecular dimensions obtained by an efficient Monte Carlo method without sample attrition," *Journal of Chemical Physics*, vol. 63, no. 11, p. 4592, 1975.

[238] S. K. Kumar, M. Vacatello, and D. Y. Yoon, "Off-lattice Monte Carlo simulations of polymer melts confined between two plates," *Journal of Chemical Physics*, vol. 89, no. 8, p. 5206, 1988.

[239] C. D. Wick and J. I. Siepmann, "Self-adapting fixed-end-point configurational-bias Monte Carlo method for the regrowth of interior segments of chain molecules with strong intramolecular interactions," *Macromolecules*, vol. 33, no. 19, pp. 7207–7218, 2000.

[240] L. R. Dodd, T. D. Boone, and D. N. Theodorou, "A concerted rotation algorithm for atomistic Monte-Carlo simulation of polymer melts and glasses," *Molecular Physics*, vol. 78, pp. 961–996, Mar. 1993.

[241] K. Kremer, A. Baumgärtner, and K. Binder, “Collapse transition and crossover scaling for self-avoiding walks on the diamond lattice,” *Journal of Physics A: Mathematical and General*, vol. 15, no. 9, pp. 2879–2897, 1982.

[242] A. Baumgärtner, “Statics and dynamics of the freely jointed polymer chain with Lennard-Jones interaction,” *Journal of Chemical Physics*, vol. 72, pp. 871–879, Jan. 1980.

[243] B. Li, N. Madras, and A. Sokal, “Critical exponents, hyperscaling, and universal amplitude ratios for two- and three-dimensional self-avoiding walks,” *Journal of Statistical Physics*, vol. 80, no. 3, pp. 661–754, 1995.

[244] L. D. Gelb, K. E. Gubbins, R. Radhakrishnan, and M. Sliwinska-Bartkowiak, “Phase separation in confined systems,” *Reports on Progress in Physics*, vol. 62, no. 12, pp. 1573–1659, 1999.

[245] W. Humphrey, A. Dalke, and K. Schulten, “VMD: visual molecular dynamics,” *Journal of Molecular Graphics*, vol. 14, pp. 33–38, Feb. 1996.

[246] J. M. Caillol, “Critical-point of the Lennard-Jones fluid: A finite-size scaling study,” *Journal of Chemical Physics*, vol. 109, no. 12, p. 4885, 1998.

[247] H. Flyvbjerg and H. G. Petersen, “Error estimates on averages of correlated data,” *Journal of Chemical Physics*, vol. 91, pp. 461–466, July 1989.

[248] D. Kent, R. Muller, A. Anderson, W. Goddard, and M. Feldmann, “Efficient algorithm for ”on-the-fly” error analysis of local or distributed serially correlated data,” *Journal of Computational Chemistry*, vol. 28, pp. 2309–2316, Nov. 2007.

[249] M. G. Martin and A. L. Frischknecht, “Using arbitrary trial distributions to improve intramolecular sampling in configurational-bias Monte Carlo,” *Molecular Physics*, vol. 104, pp. 2439–2456, Aug. 2006.

[250] M. Muthukumar, *Polymer Translocation.* Taylor & Francis US, May 2011.

[251] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell, Fourth Edition.* Garland Science, 4 ed., Mar. 2002.

[252] S. Casjens, *Virus Structure and Assembly.* Jones & Bartlett Pub, 1 ed., Jan. 1985.

[253] G. Schatz and B. Dobberstein, “Common principles of protein translocation across membranes,” *Science*, vol. 271, p. 1519, Mar. 1996.

[254] A. Meller, “Dynamics of polynucleotide transport through nanometre-scale pores,” *Journal of Physics: Condensed Matter*, vol. 15, pp. R581–R607, May 2003.

[255] H. W. C. Postma, “Rapid sequencing of individual DNA molecules in graphene nanogaps,” *Nano Letters*, vol. 10, no. 2, pp. 420–425, 2010.

[256] C. A. Merchant, K. Healy, M. Wanunu, V. Ray, N. Peterman, J. Bartel, M. D. Fischbein, K. Venta, Z. Luo, A. T. C. Johnson, and M. Drndic?, “DNA translocation through graphene nanopores,” *Nano Letters*, vol. 10, no. 8, pp. 2915–2921, 2010.

[257] S. Alapati, D. V. Fernandes, and Y. K. Suh, “Numerical and theoretical study on the mechanism of biopolymer translocation process through a nano-pore,” *Journal of Chemical Physics*, vol. 135, pp. 055103–055103–11, Aug. 2011.

[258] M. Bernaschi, S. Melchionna, S. Succi, M. Fyta, and E. Kaxiras, “Quantized current blockade and hydrodynamic correlations in biopolymer translocation through nanopores: Evidence from multiscale simulations,” *Nano Letters*, vol. 8, no. 4, pp. 1115–1119, 2008.

[259] P. Park and W. Sung, “Polymer translocation induced by adsorption.,” *Journal of Chemical Physics*, vol. 108, p. 3013, Feb. 1998.

[260] K. Kiran Kumar and K. L. Sebastian, “Adsorption-assisted translocation of a chain molecule through a pore,” *Physical Review E*, vol. 62, p. 7536, Nov. 2000.

[261] S. Yang and A. V. Neimark, "Adsorption-driven translocation of polymer chain into nanopores," *Journal of Chemical Physics*, vol. 136, pp. 214901–214901–15, June 2012.

[262] A. Milchev, K. Binder, and A. Bhattacharya, "Polymer translocation through a nanopore induced by adsorption: Monte Carlo simulation of a coarse-grained model," *Journal of Chemical Physics*, vol. 121, no. 12, pp. 6042–6051, 2004.

[263] W. Sung and P. J. Park, "Polymer translocation through a pore in a membrane," *Physical Review Letters*, vol. 77, p. 783, July 1996.

[264] D. K. Lubensky and D. R. Nelson, "Driven polymer translocation through a narrow pore," *Biophysical Journal*, vol. 77, pp. 1824–1838, Oct. 1999.

[265] J. Chuang, Y. Kantor, and M. Kardar, "Anomalous dynamics of translocation," *Physical Review E*, vol. 65, p. 011802, Dec. 2001.

[266] S. Mirigian, Y. Wang, and M. Muthukumar, "Translocation of a heterogeneous polymer," *Journal of Chemical Physics*, vol. 137, pp. 064904–064904–17, Aug. 2012.

[267] C. Kong and M. Muthukumar, "Polymer translocation through a nanopore. II. excluded volume effect," *Journal of Chemical Physics*, vol. 120, pp. 3460–3466, Feb. 2004.

[268] R. Metzler and J. Klafter, "When translocation dynamics becomes anomalous," *Biophysical Journal*, vol. 85, pp. 2776–2779, Oct. 2003.

[269] V. V. Lehtola, R. P. Linna, and K. Kaski, "Unforced polymer translocation compared to the forced case," *Physical Review E*, vol. 81, p. 031803, Mar. 2010.

[270] D. Wei, W. Yang, X. Jin, and Q. Liao, "Unforced translocation of a polymer chain through a nanopore: The solvent effect," *Journal of Chemical Physics*, vol. 126, p. 204901, May 2007.

[271] V. V. Lehtola, R. P. Linna, and K. Kaski, "Critical evaluation of the computational methods used in the forced polymer translocation," *Physical Review E*, vol. 78, p. 061803, Dec. 2008.

[272] P. Tian and G. D. Smith, "Translocation of a polymer chain across a nanopore: A brownian dynamics simulation study," *Journal of Chemical Physics*, vol. 119, no. 21, p. 11475, 2003.

[273] J. L. A. Dubbeldam, V. G. Rostiashvili, A. Milchev, and T. A. Vilgis, "Fractional brownian motion approach to polymer translocation: The governing equation of motion," *Physical Review E*, vol. 83, p. 011802, Jan. 2011.

[274] J. L. A. Dubbeldam, V. G. Rostiashvili, A. Milchev, and T. A. Vilgis, "Forced translocation of a polymer: Dynamical scaling versus molecular dynamics simulation," *Physical Review E*, vol. 85, p. 041801, Apr. 2012.

[275] F. Kapahnke, U. Schmidt, D. W. Heermann, and M. Weiss, "Polymer translocation through a nanopore: The effect of solvent conditions," *Journal of Chemical Physics*, vol. 132, p. 164904, Apr. 2010.

[276] S.-S. Chern, A. E. Cárdenas, and R. D. Coalson, "Three-dimensional dynamic Monte Carlo simulations of driven polymer transport through a hole in a wall," *Journal of Chemical Physics*, vol. 115, p. 7772, Oct. 2001.

[277] J. K. Wolterink, G. T. Barkema, and D. Panja, "Passage times for unbiased polymer translocation through a narrow pore," *Physical Review Letters*, vol. 96, p. 208301, May 2006.

[278] J. L. A. Dubbeldam, A. Milchev, V. G. Rostiashvili, and T. A. Vilgis, "Polymer translocation through a nanopore: A showcase of anomalous diffusion," *Physical Review E*, vol. 76, p. 010801, July 2007.

[279] D. B. Wells, V. Abramkina, and A. Aksimentiev, "Exploring transmembrane transport through alpha-hemolysin with grid-steered molecular dynamics," *Journal of Chemical Physics*, vol. 127, p. 125101, Sept. 2007. PMID: 17902937.

[280] A. Milchev, "Single-polymer dynamics under constraints: scaling theory and computer experiment," *Journal of Physics: Condensed Matter*, vol. 23, p. 103101, Mar. 2011.

[281] M. Muthukumar, "Polymer escape through a nanopore," *Journal of Chemical Physics*, vol. 118, p. 5174, Mar. 2003.

[282] P.-G. de Gennes, *Scaling concepts in polymer physics.* Cornell University Press, 1979.

[283] J. Li and D. S. Talaga, "The distribution of DNA translocation times in solid-state nanopores," *Journal of Physics: Condensed Matter*, vol. 22, p. 454129, Nov. 2010.

[284] W. Radke, "Chromatography of polymers," in *Macromolecular Engineering* (Krzysztoftyjaszewski, Y. Gnanou, and L. Leibler, eds.), p. 1881–1936, Wiley-VCH Verlag GmbH & Co. KGaA, 2007.

[285] A. Bhattacharya, W. H. Morrison, K. Luo, T. Ala-Nissila, S.-C. Ying, A. Milchev, and K. Binder, "Scaling exponents of forced polymer translocation through a nanopore," *European Physical Journal E – Soft Matter*, vol. 29, no. 4, pp. 423–429, 2009.

[286] O. V. Krasilnikov, C. G. Rodrigues, and S. M. Bezrukov, "Single polymer molecules in a protein nanopore in the limit of a strong polymer-pore attraction," *Physical Review Letters*, vol. 97, p. 018301, July 2006.

[287] A. Milchev, J. L. A. Dubbeldam, V. G. Rostiashvili, and T. A. Vilgis, "Polymer translocation through a nanopore: A showcase of anomalous diffusion," *Annals of the New York Academy of Sciences*, vol. 1161, pp. 95–104, Apr. 2009.

[288] A. J. Storm, J. H. Chen, H. W. Zandbergen, and C. Dekker, "Translocation of double-strand DNA through a silicon oxide nanopore," *Physical Review E*, vol. 71, p. 051903, May 2005.

[289] A. J. Storm, C. Storm, J. Chen, H. Zandbergen, J.-F. Joanny, and C. Dekker, "Fast DNA translocation through a solid-state nanopore," *Nano Lett.*, vol. 5, no. 7, pp. 1193–1197, 2005.

[290] A. J. P. Martin *Biochemical Society Symposia*, vol. 3, pp. 4—10, 1949.

[291] E. F. Casassa and Y. Tagami, "An equilibrium theory for exclusion chromatography of branched and linear polymer chains," *Macromolecules*, vol. 2, pp. 14–26, Jan. 1969.

[292] A. A. Gorbunov and A. M. Skvortsov, "Statistical properties of confined macromolecules," *Advances in Colloid and Interface Science*, vol. 62, pp. 31–108, Nov. 1995.

[293] A. Skvortsov and A. Gorvunov, "Adsorption effects in the chromatography of polymers," *Journal of Chromatography A*, vol. 358, pp. 77–83, 1986.

[294] A. Gorbunov and B. Trathnigg, "Theory of liquid chromatography of mono- and difunctional macromolecules: I. studies in the critical interaction mode," *Journal of Chromatography A*, vol. 955, pp. 9–17, Apr. 2002.

[295] A. A. Gorbunov and A. M. Skvortsov *Vysokomol. Soedin. Ser. A (U.S.S.R.)*, vol. 30, p. 453, 1988.

[296] A. A. Gorbunov and A. V. Vakhrushev, "Theory of chromatographic separation of linear and star-shaped binary block-copolymers," *Journal of Chromatography A*, vol. 1064, pp. 169–181, Feb. 2005.

[297] J. Ennis, L. Sjöström, T. Åkesson, and B. Jönsson, "Surface interactions in the presence of polyelectrolytes. a simple theory," *Langmuir*, vol. 16, no. 18, pp. 7116–7125, 2000.

[298] Y. Brun and P. Foster, "Characterization of synthetic copolymers by interaction polymer chromatography: Separation by microstructure," *Journal of Separation Science*, vol. 33, pp. 3501–3510, Nov. 2010.

[299] A. A. Gorbunov and A. V. Vakhrushev, "Theory of chromatography of linear and cyclic polymers with functional groups," *Polymer*, vol. 45, pp. 7303–7315, Sept. 2004.

[300] M. A. Bashir and W. Radke, “Comparison of retention models for polymers: 1. poly(ethylene glycol)s,” *Journal of Chromatography A*, vol. 1131, pp. 130–141, Oct. 2006.

[301] M. A. Bashir and W. Radke, “Predicting the chromatographic retention of polymers: Poly(methyl methacrylate)s and polyacryate blends,” *Journal of Chromatography A*, vol. 1163, pp. 86–95, Sept. 2007.

[302] M. A. Bashir and W. Radke, “Predicting the chromatographic retention of polymers: Application of the polymer model to poly(styrene/ethylacrylate)copolymers,” *Journal of Chromatography A*, vol. 1225, pp. 107–112, Feb. 2012.

[303] K. Matyjaszewski, Y. Gnanou, and L. Leibler, *Macromolecular Engineering: Precise Synthesis, Materials Properties, Applications.* Wiley-VCH, 1 ed., Apr. 2007.

[304] L. R. Snyder, J. J. Kirkland, and J. L. Glajch, *Practical HPLC Method Development, 2nd Edition.* Wiley-Interscience, 2nd ed., Mar. 1997.

[305] P. Schoenmakers, H. Billiet, R. Tussen, and L. De Galan, “Gradient selection in reversed-phase liquid chromatography,” *Journal of Chromatography A*, vol. 149, pp. 519–537, Feb. 1978.

[306] E. Jones, T. Oliphant, P. Peterson, and others, “SciPy: open source scientific tools for Python,” 2001.

[307] J. Ennis and B. Jönsson, “Interactions between surfaces in the presence of ideal adsorbing block copolymers,” *Journal of Physical Chemistry B*, vol. 103, no. 12, pp. 2248–2255, 1999.

[308] C. J. Rasmussen and A. V. Neimark, “Calculation of chemical potential of polymers in molecular Monte Carlo simulations,” *Molecular Simulation*, vol. in preparation, 2012.

[309] C. J. Rasmussen, A. Vishnyakov, and A. V. Neimark, “Translocation dynamics of freely-jointed Lennard-Jones chains through a small opening into adsorbing pore,” *Journal of Chemical Physics*, vol. in press, 2012.