Spring 5-2021

Modelling the Filling of Methane in Heterogenous Pore Networks

Samuel Bonnevie

MODELLING THE FILLING OF METHANE IN HETEROGENEOUS PORE NETWORKS

by

Samuel W. Bonnevie

A Thesis Submitted in Partial Fulfillment
of the Requirements for a Degree with Honors
(Chemistry and Chemical Engineering)

The Honors College

University of Maine

May 2021

Advisory Committee:
Brian Frederick, Associate Professor of Chemistry, Co-Advisor
François Amar, Professor of Chemistry, Co-Advisor
William DeSisto, Professor of Chemical Engineering
Melissa Ladenheim, Associate Dean of the Honors College
Thomas Schwartz, Associate Professor of Chemical Engineering

ABSTRACT

In the field of heterogeneous catalysis, there is great interest in the transport properties of ordered mesoporous materials such as SBA-15, but inverting quasi-elastic neutron scattering data for materials with a distribution of pore sizes such as SBA-15 is an ill-posed problem. This project aimed to generate an idealized model of methane adsorption in the pores of SBA-15 so that in the future, molecular dynamics simulations can be used to study diffusion. By sampling over a canonical ensemble using the Metropolis Monte Carlo Method and using Widom's insertion method alongside Vaitheeswaran and Rasaiah's insertion/removal method to calculate the chemical potential, isotherms comparable to those generated by methane porosimetry measurements can be produced. Plots of chemical potential vs. number of molecules were used to show that the simulation data is reproducible to 2% relative standard deviation, as well as to build an understanding of the mechanism of pore filling and how it is affected by simulation conditions.

ACKNOWLEDGEMENTS

to Brian, for his endless patience and brilliant explanation,

to François, for his beautiful mind and beautiful vocabulary,

to Ben, for his dedication to helping and his quick smile,

to Melissa, for setting me straight and steering me forward,

to Abby, for her amazing art and eagerness to help,

to the rest of my family, for their infinite love and support,

to my team, for being my big family up at school,

to Jay, for his incredible wisdom and for making me think this was a good idea,

to my fellow Chemistry students, who made this year a blast,

and to Moxie, for always being the happiest when I come home.

special thanks goes out to the funders and institutional support of the project: the Center for Undergraduate Research, the Maine Space Grant Consortium, the Advanced Computing Group, the Departments of Chemistry and Chemical Engineering, and past support from the Forest Bioproducts Research Institute, the Frontier Institute for Research in Sensor Technology, and the U.S. Department of Energy.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. General Motivation

The ability to characterize the diffusion of a fluid in any heterogeneous material is useful in a broad range of disciplines. Fracking involves the diffusion of natural gas out of porous rock. If drilling engineers have a picture of how much gas will diffuse into the pipeline, how quickly, and from how far away, they can set up drilling positions to gather the most gas in the smallest amount of time. In catalysis, the effective reaction rate is often limited by the rate of diffusion of the reactants to catalyst sites. Therefore, by increasing the rate of diffusion in the system, the effectiveness of the catalyst is directly increased. Additionally, with enough improvement in catalyst performance, biofuel production can be extended to a wider range of biomass sources and made more profitable.

B. Environmental Motivation

Fossil fuels, or non-renewable carbon sources, currently constitute the majority of energy production and consumption in the United States; in 2019, they made up 80.1% of overall energy production and 80.0% of overall energy consumption.[1] However, reserves of fossil fuels are by nature dwindling and will eventually have to be replaced. Ahlbrandt and McCabe[2] speculate that out of the three trillion barrels of oil estimated to be left in the world, about 24% have been produced and 29% have been discovered. To mitigate the inevitable reduction in energy availability as these reserves disappear, there has been increasing interest in renewable sources of energy. Biofuel production is among the most

attractive options for multiple reasons. First, biofuel is nearly infinitely renewable: it can be made from almost any plant matter, such as lignin, algae, switchgrass, and corn stover, among many others.[3-8] Second, because the production of biofuel resources (e.g., plants and bacteria) actively sequesters carbon from the atmosphere, biofuel is much more environmentally friendly than fossil fuels. Finally, unlike electrically-based energy storage, biofuels are compatible with existing energy production systems, from generators to internal combustion engines.[9]

C. Catalytic Pyrolysis

One of the most effective techniques for the generation of biofuel and biofuel precursors is fast pyrolysis. This method can be used to thermochemically transform lignin and cellulose, among other biological molecules, into liquid bio-oil.[3,10] To further increase the quality of the bio-oil, catalytic pyrolysis can be implemented, wherein a catalyst is introduced during the fast pyrolysis process. This induces catalytic reactions between the bio-oil constituents, increasing yield and heating value.[8,10,11] In catalytic pyrolysis, the size of the pores in the catalyst or catalyst support plays an important role in determining the rate of reaction and the quality of the products. Pyrolysis oil inherently contains compounds with a large distribution of molecular weights, some very high. In traditional microporous catalysts such as zeolites with pore diameters of about 5-13 Å, these high molecular weight compounds are excluded from pores, limiting their reaction to the external surface of the catalyst particles. In addition to reducing yield and product quality, this can also cause spontaneous polymerization of the bio-oil constituents.[12] Instead, the use of Ordered Mesoporous Materials (OMMs) as catalyst supports is greatly favored. Their ordered framework, high surface area, large pore size and large pore

volumes allow high diffusion rates to a large number of catalyst sites.[13] This increase in diffusion rate increases the effective reaction rate and allows the reaction of larger molecular weight compounds, preventing catastrophic polymerization.[3] Many other processes involving reactions using OMMs as a support in this way have been studied, including the Fischer-Tropsch reaction, steam reforming, and enantioselective reactions.[3,14-16] Of particular interest to many studying OMM-supported catalysis has been the silica SBA-15, as it has a 2D hexagonal array of uniformly distributed large mesopores, as illustrated in Figure 1.1. These mesopores are tailorable from 30-300 Å and aid in diffusion, give it a large specific surface area, and increase its thermal and mechanical stability relative to other OMMs.[13]

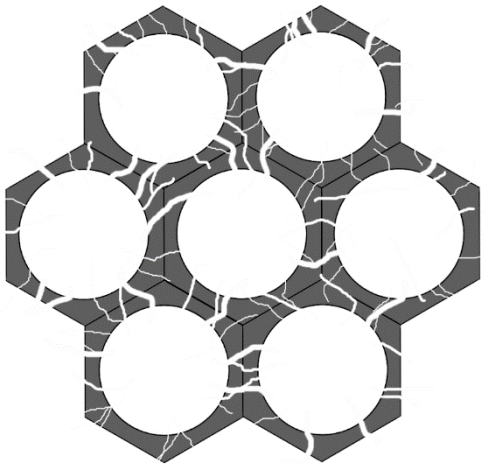

Figure 1.1. Artistic representation of the micropore-mesopore network of SBA-15.

D. SBA-15 Synthesis

Like many porous ceramics, SBA-15 is formed by generating silica around a template, then removing the template. SBA-15 in particular uses a solution of amphiphilic polymers consisting of ethylene and propylene oxide units such as $EO_{20}PO_{70}EO_{20}$ or Pluronic P123. These polymers form cylindrical micelles that in low ratios of EO to PO

assemble into a hexagonal pattern. It is predicted that especially at low temperatures, some EO blocks partially unravel, sticking into the spaces between micelles. Silica is precipitated into the solution, forming an amorphous solid around the micelles.[17] The template is then removed via calcination in air, leaving a two-dimensional hexagonal array of large mesopores with interconnecting micropores resulting from the unraveled micelles. The size of the mesopores can be controlled by altering synthesis conditions and brought up to 300 Å with the use of a swelling agent. Morphology can also be changed by altering synthesis conditions. The microporosity is also tailorable and can be controlled using varying heating methods and solvents during the synthesis process.[13]

E. Diffusion in SBA-15

To improve the selectivity and efficiency of SBA-15, it is necessary to improve the understanding of its diffusion characteristics. A significant amount of research has been done characterizing the diffusion in porous materials, but little research has been done to describe comprehensively the diffusion within materials like SBA-15 that have a *broad distribution* of pore sizes.[18-21] In materials with a single pore size, Quasi-Elastic Neutron Scattering data can be directly inverted to find the diffusion constant. If no neutrons scattered off moving molecules in a sample, the width of the elastic peak would only be due to the instrumental limit of the monochromator of the energy distribution of the neutron beam. However, if there are molecules moving in the sample, the neutrons either gain or lose energy based on the energy and character of the movement, broadening the peak, which is generally referred to as the "quasi-elastic" peak. Figure 1.2 shows measurements of methane in SBA-15 by Pollock.[22] This broadening can be attributed to

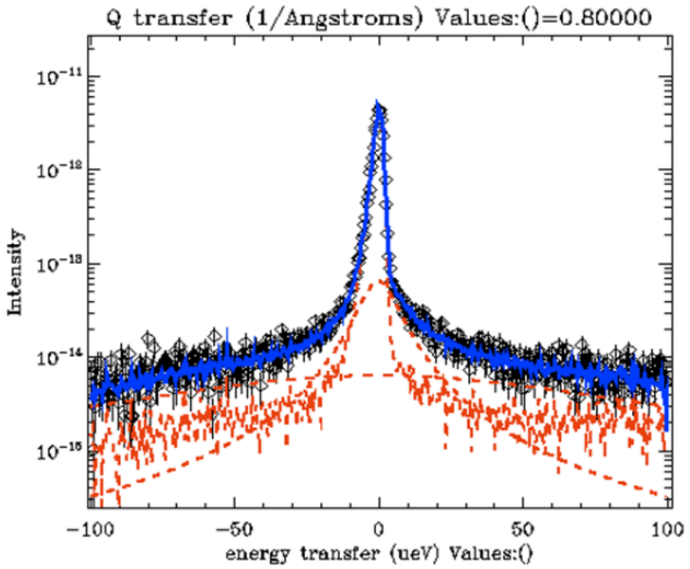

Figure 1.2. QENS data showing the broadening of the elastic peak. Reproduced from Pollock.[22]

translational and rotational motion of the molecules within the sample. In ideal systems with a single pore size, the Knudsen diffusion constant in the sample can be calculated from the dependence of the broadening with momentum transfer.[21] Doing this for materials with a distribution of pore sizes is an ill-posed problem: there is no one diffusion constant for the material because the diffusion constant varies with pore size. Therefore, the neutron scattering data cannot be directly inverted. Instead, molecular dynamics simulations can be used to predict QENS peak broadening, and diffusion characteristics can be elucidated from there. This thesis presents work that is a continuation of Monte Carlo simulations started by Pollock, with contributions from York and Walden.

F. Pore Structure Determination

Before molecular dynamics simulations can be performed, it is necessary to define a realistic structural model of SBA-15. To do that, it was first necessary to fully understand the pore structure of SBA-15. Extending the work of other researchers,[13,17,23] Pollock et

al.[24] did measurements using nitrogen and argon porosimetry, X-ray diffraction, and Contrast Matching Small Angle Neutron Scattering (CM-SANS). They confirmed that SBA-15 is made up of a 2D hexagonal arrangement of mesopores, interconnected by micropores, and furthered research by determining that because of the size of the smallest micropores, about 5.7 Å, the shape of the micropores was more likely to be described by the Saito-Foley[25] description of cylindrical pores than the Horvath-Kawazoe[26] description of slit-shaped pores, as illustrated in Figure 1.3. They also determined that the micropores are distributed uniformly throughout the mesopore network, rather than in a corona around the mesopores as previously thought.[27]

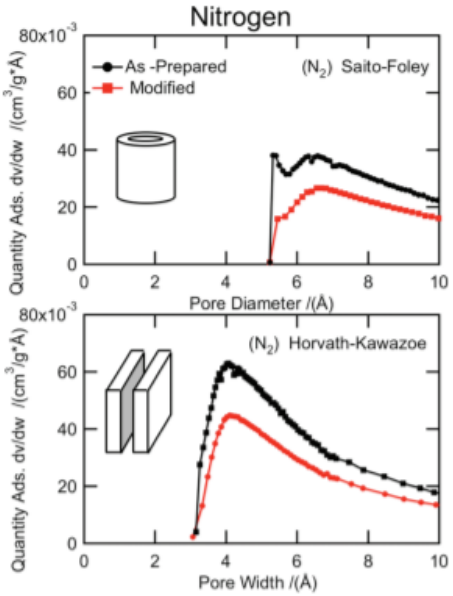

Figure 1.3. Analysis of nitrogen adsorption isotherms for pores of the given shape showing that the pore size distribution for cylindrical pores is consistent with the 5.7 Å diameter found in CM-SANS measurements. Reproduced from Pollock, et al.[24]

In Chapter II, we demonstrate how we generate an idealized model of methane adsorbing into the heterogeneous pore structure of SBA-15 using spherical and smeared Lennard-Jones potentials. We then sample the canonical ensemble using the Metropolis

Monte Carlo method and calculate the chemical potential from both the Widom insertion method and the Vaitheeswaran-Rasaiah insertion/deletion method. From the chemical potential and from the ratios of pore filling probabilities, we are able to generate isotherms comparable to the methane porosimetry measurements. In Chapter III, we analyze the reproducibility of the methods relative to repetitions and to each other, and find that except at very high filling, the data was highly reproducible. We explain the behavior of the chemical potential vs. the number of molecules in the pore at different temperatures, pore radii, pore lengths, and report the presence of discontinuities potentially indicating metastable configurations in low temperature runs. Chapter IV presents conclusions and prospects for future work.

II. THEORETICAL METHODS

A. Idealized Model of Methane in SBA-15 Pores

The first step in the generation of data for this project was to create an idealized model of the species present in the methane porosimetry measurements. Bhattacharya et al. have modeled adsorption in a structure with more atomistic detail,[28] however their model is not useful for the averaging over sizes used in this project. Therefore, we assumed that the microporous structure of SBA-15 can be simulated by generating data from each of a range of pore sizes individually. As mentioned previously, predictions from the method of Saito and Foley[25] and CM-SANS measurements by Pollock, et al.[24] indicated that the pores were cylindrical in shape. Though the pore walls are made up of discrete silicon and oxygen atoms, for the purposes of this project we used an idealized model of the pore. This was also the technique used by Pollock, et al.[24] when doing the Neimark method[29] NLDFT analysis used in the background of this project. The unraveled ethylene oxide polymers that are theorized to template the microporous structure of SBA-15 are of relatively high aspect ratio, so end effects were assumed to be negligible. To remove these effects while maintaining a finite volume, the cylindrical pores were treated with periodic boundary conditions along the z axis of the pore. Under these conditions, a move that brings a molecule outside the bounds of the pore a certain distance in the z direction causes it to enter the other end of the pore the same distance, preserving the x and y coordinates.

All neutral molecules experience fluctuations in their electron cloud that can create a slight dipole. At a certain distance from other neutral molecules, this dipole can induce a dipole in those molecules, creating a slight attractive force between the molecules.[30] When the molecules are brought closer together, the electron clouds interact strongly enough to repel each other. This attractive/repulsive interaction can be modeled by a Lennard-Jones expression

$$u(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{2.1}$$

where ϵ is the depth of the potential well or the strength of the maximum attractive force, *r* is the center-to-center distance between the molecules, and σ is the non-trivial point at which the attractive and repulsive forces are equal, as illustrated in Figure 2.1. The methane molecules in the simulations were modeled using a spherical Lennard-Jones potential. Values for ϵ (used in the form $\frac{\epsilon}{k_B}$ with a value of 148 K) and σ (3.73 Å) were obtained from the molecular dynamics work done by Goodbody et al.,[31] whose work produced diffusion coefficients for methane within silicalite that matched experimental PFG NMR data.

For the interaction of the molecule with the wall of the pore, the Lennard-Jones potential (shown in Figure 2.1) was integrated over the entire inner surface of the cylindrical pore, as shown in Figure 2.2. This potential was modelled directly after the work done by Tjatjopolous et al..[32] The depth of the potential well of the smeared Lennard-Jones potential depends on the radius, but the parameters ϵ and σ were constant ($\epsilon = 133\ K$and $\sigma = 3.21$ Å).[31] Note because we are using Kelvin units for our energy parameters, we will often report thermodynamic quantities in units of Kelvins, and in these units,

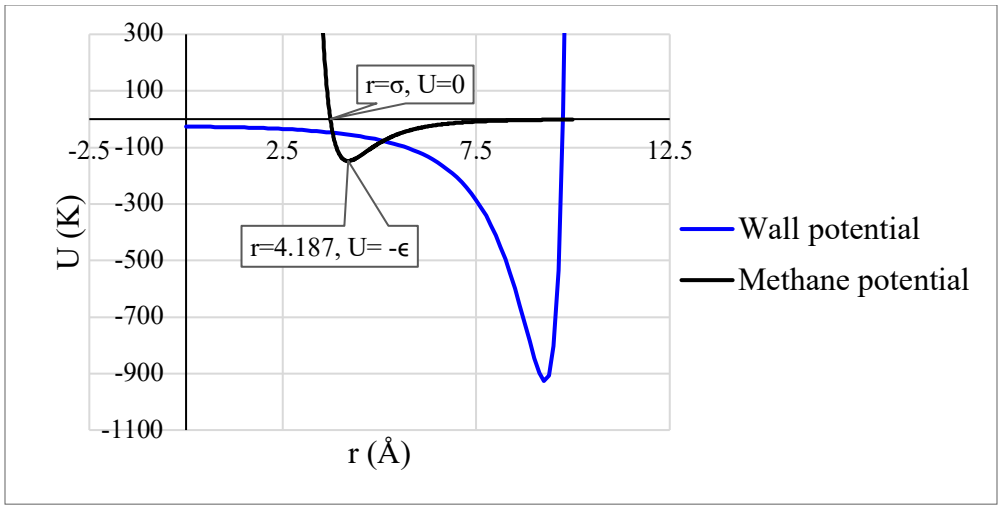

Figure 2.1. Graph of methane-methane Lennard-Jones potential showing significance of epsilon and sigma. Overlaid upon graph of methane-wall potential for 12.5 Å pore.

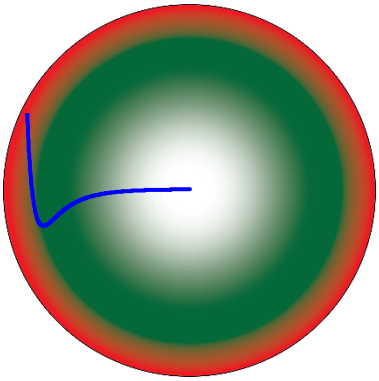

Figure 2.2. Artistic representation of the cross-section of a cylindrical pore with a smeared Lennard-Jones potential, overlayed with a radial plot. Red represents repulsive force, green represents attractive force, and the tint represents the strength of the force. Courtesy of Abby Bonnevie.

Boltzmann's constant is unity. The length of the pore (that is, the furthest z distance a molecule could travel before reentering the other end of the pore due to the periodic boundary conditions)was chosen to be 10σ, or 37.3 Å. This value was chosen because at 5σ distance from the center of a Lennard-Jones methane, the attractive force is only 0.03% ϵ, which is negligible for our purposes. If the pore was shorter than 10σ, non-negligible forces could wrap around the end of the pore and add to forces from the other side of the same molecule, even causing the molecule to attract itself. A longer pore would require more molecules to reach saturation, unnecessarily increasing computation time.

B. Canonical Ensemble

To generate data describing an experimentally equivalent macroscopic sample in equilibrium, we used the canonical ensemble. The canonical ensemble represents real systems by averaging thermodynamic properties over a large number of mechanically isolated systems that have a constant and equal number of particles, volume, and temperature. By averaging the potential energy of each system in the ensemble, an internal energy which is equivalent to that of a macroscopic sample can be found. This in turn can be used to obtain other thermodynamic properties of the ensemble using the canonical partition function,

$$Q(N,V,T) \equiv \frac{1}{N!\sqrt{h^2/(2\pi m k_B T)}^{3N}} \int d\boldsymbol{r^N} exp[-\beta U(\boldsymbol{r^N})], \tag{2.2}$$

where N is the number of molecules in each system, V is the volume of the configuration space, T is the temperature of each system, m is the mass of each molecule, $\boldsymbol{r}^N$ is the configuration of N molecules in 3N dimensional space, β is $\frac{1}{k_B T}$ and U is the total potential energy of the system in the configuration $\boldsymbol{r}^N$ relative to the isolated ideal gas phase where U=0.

C. Monte Carlo Method

To sample from the ensemble, we used the Metropolis Monte Carlo method[33] of exploring the configuration space of the pore: at a given N, the molecules are arranged randomly inside the pore, and the energy of a given molecule is calculated from its interactions with all other molecules using the Lennard-Jones potential and interactions with the pore wall using the smeared Lennard-Jones potential. The molecule is then

moved a random distance on each axis, up to a maximum distance of 0.2 Å in each direction. This distance was chosen to maximize program efficiency: longer moves are less likely to be accepted and are therefore unproductive, while shorter moves require more trials to facilitate equilibration. Moves on the x and y axes outside of a radius 1.5 Å smaller than the radius of the pore were excluded, as between this radius and the radius of the pore the energy of a molecule in that position would be so high that the probability of it being there is negligible, and data from those positions do not contribute to further calculations.

After the molecule is moved, the energy of the configuration is recalculated, and the change in energy, ΔU, is found. If ΔU<0, i.e. if the move brought the configuration to a lower energy state, the move is accepted. If ΔU>0, the move is accepted with a probability $P = exp(-\beta \Delta U)$. Whether or not to accept the move is determined by comparing that probability to a random number between 0 and 1. If the random number is less than the exponential, the move is accepted and vice versa. Every N+10 accepted moves, the configuration is considered a new system for the purpose of canonical averages. This method generates configurations with energy U_r in the Boltzmann distribution $p_r \propto exp(-\beta U_r)$, where p_r is the probability of configuration r.

D. Calculation of Chemical Potential: Widom Insertion Method

Chemical potential is a particularly useful value for this project because two phases in equilibrium with each other share the same chemical potential. That is, in the methane porosimetry measurements, when the chamber had reached equilibrium the adsorbed phase and the bulk phase had the same chemical potential, and it being a gas the value for

the bulk phase was defined. Therefore, the chemical potential can indirectly be used as a quantitative measure of the accuracy of simulations of equilibria. It can be related directly to Helmholtz energy:

$$\mu = \left(\frac{\delta A}{\delta n}\right)_{T,V} \tag{2.3}$$

and to the canonical partition function:

$$\mu = -k_B T \frac{\delta lnQ}{\delta n}. \tag{2.4}$$

As shown in Equation 2.4, in principle, the chemical potential can be found by solving the canonical partition function directly. However, direct evaluation of Q is only possible for a gas.[34] For the high density conditions found in our ensembles involving molecule-pore interactions, the Widom insertion method[35] provides a means to obtain the excess chemical potential $\mu_{ex} = \mu_N - \mu_{id}$, where μ_N is the chemical potential of N molecules in the pore and the chemical potential of an ideal gas, $\mu_{id} = RTln\left(\frac{P}{P^0}\right)$. In this method, a pore of N molecules is first equilibrated using Monte Carlo moves. Then, random positions are chosen to insert an additional molecule (5N times in our simulations). Before and after each insertion, the energy of the system is recorded so that the change in system energy due to the addition, $\Delta U = U_{N+1} - U_N$, can be found. These values are stored in an array, which can be plotted as a histogram of the frequency of insertion energy, $F_{ins}(\Delta U)$, vs. ΔU, known as the insertion histogram. Monte Carlo moves are then performed until another N+10 moves have been accepted and the system is considered in a new configuration. The insertion process is repeated and Monte Carlo moves continue for a total of 10^6 attempted moves. This method was used to calculate the excess chemical potential based on the equation[36]

$$\mu_{ex} = -k_B T ln \int d\boldsymbol{s}_{N+1} \langle exp(-\beta \Delta U) \rangle_{\boldsymbol{N}} \quad (2.5)$$

where μ_{ex} is the excess chemical potential, $\boldsymbol{s}_{N+1}$ is set of coordinates for the insertion of the $(N+1)^{th}$ molecule, and $\langle ... \rangle_N$ is the exponential of the change in energy due to the addition of a molecule, averaged over the configurations of the N molecules before the addition of the $(N+1)^{th}$ molecule. Figure 2.3A illustrates a typical histogram, $F_{ins}(\Delta U)$ vs. ΔU, typically containing 250,000 insertion values. Normalization of the insertion histogram gives the probability distribution, $p_{ins,N}(\Delta U)$, as shown in Figure 2.3B.

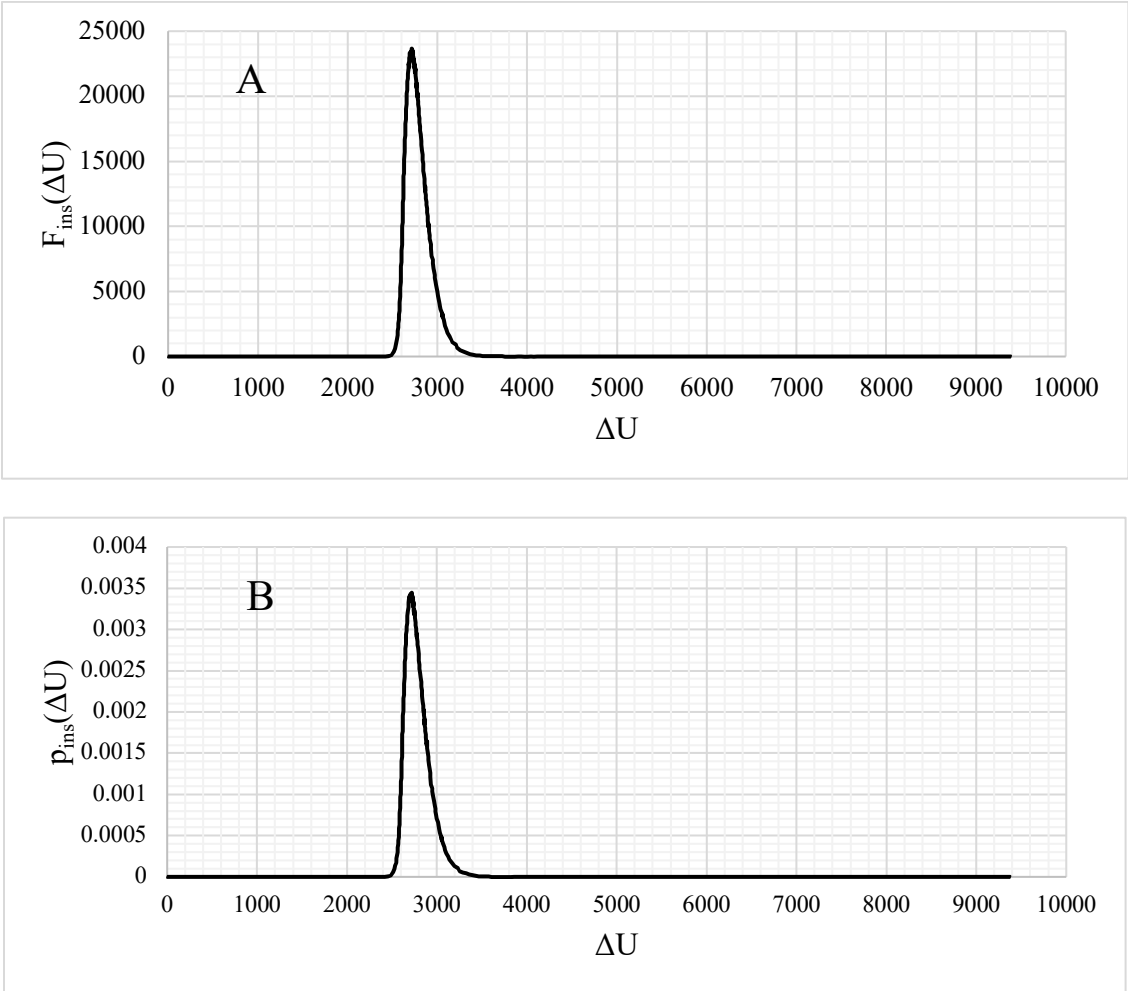

Figure 2.3. Example of A) an insertion histogram of insertion frequency vs. insertion energy and B) the corresponding normalized probability distribution.

E. Calculation of Chemical Potential: Vaitheeswaran-Rasaiah Method

In addition to using the Widom insertion method, we calculated the chemical potential via a method developed by Vaitheeswaran and Rasaiah.[37] This method was chosen as it provides better signal-to-noise, and there was a desire to exemplify more applications for the method. It utilizes the same technique for inserting molecules as the Widom method, but adds a removal piece: before the insertion of a molecule 5N times, each of the N molecules already in the pore is removed and replaced one at a time, and the change in system energy is again recorded. These values are accumulated in a histogram of the frequency of removal energy, $F_{rem}(\Delta U)$ vs. ΔU, where $\Delta U = U_N - U_{N-1}$. Normalization of the histogram gives the removal probability distribution, $p_{rem}(\Delta U)$, in the same manner as Figure 2.3B. Vaitheeswaran and Rasaiah show that the ratio of the distributions of insertion and removal energies is equal to

$$\frac{p_{ins,N}(\Delta U)}{p_{rem,N+1}(\Delta U)} = e^{\beta \Delta U} \langle e^{-\beta(U_{N+1}-U_N)} \rangle_N \tag{2.6}$$

where $p_{ins,N}$ is the distribution of insertion energies with N molecules in the pore and $p_{rem,N+1}$ is the distribution of removal energies with N+1 molecules. From consecutive simulations with N and N+1 molecules in the pore, the insertion probability and the removal probability provide the ratio on the left side of eqn. 2.6. Taking the logarithm,

$$ln\left(\frac{p_{ins,N}(\Delta U)}{p_{rem,N+1}(\Delta U)}\right) = \beta \Delta U + ln\langle \dots \rangle, \tag{2.6A}$$

provides the basis for obtaining the configuration integral $\langle \dots \rangle$ from simulation. By fitting the natural log of the distribution ratio as a function of ΔU to a straight line, the y-intercept of the fit will be the average $\langle \dots \rangle_N$. This in turn is shown[37] to be equal to the

exponential of the excess chemical potential:

$$exp(-\beta\mu_N^{ex}) = \langle exp[-\beta(U_{N+1} - U_N)]\rangle_N. \quad (2.7)$$

F. Generation of Isotherms

Vaitheeswaran and Rasaiah also show that the ratio of the probability of finding N+1 molecules in the configuration space, P(N+1), to the probability of finding N molecules, P(N), is

$$\frac{P(N+1)}{P(N)} = \frac{\rho V}{N+1} e^{-\beta(\mu_N^{ex} - \mu_{bulk}^{ex})} \quad (2.8)$$

where ρ is the number density of the bulk phase, V is the volume of the configuration space, and μ_{bulk}^{ex} is the excess chemical potential of the bulk fluid. In the case of our ensemble, the bulk fluid is an ideal gas, so this term disappears, leaving only the excess chemical potential. Easily converting the density of the bulk phase to pressure results in a function of probability similar to the probability of finding N molecules in a pore at a given pressure. However, the output is only probability ratios, not probabilities. To find the probabilities, we first arbitrarily set the probability of finding 0 molecules in the pore, P(0), to a very small number. Using the calculated ratios, we then found the probabilities up to P(N). The probabilities were then summed and normalized to unity. The average of the probability distribution was $\langle N \rangle$. By plotting $\langle N \rangle$ versus bulk pressure, we created isotherms as shown in Figure 2.4. However, the isotherm for each pore radius was only representative of a sample with pores of that single radius. Therefore, we summed and weighted each isotherm based on the pore size distribution found by Pollock et al.,[24] and from that data constructed an overall isotherm, as seen in Figure 2.5, which is comparable to the data produced during the methane porosimetry measurements.

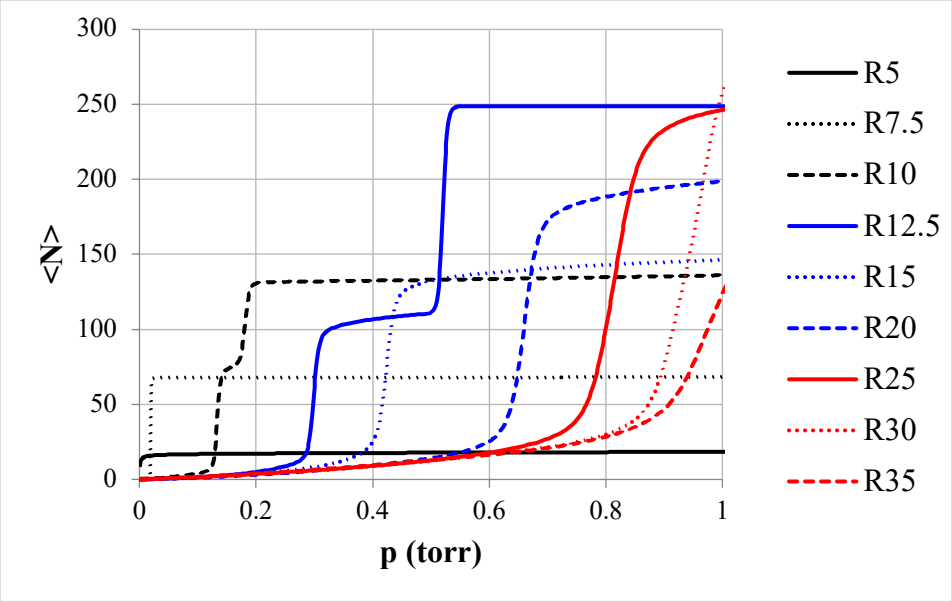

Figure 2.4. Plot of average number of molecules vs. bulk pressure for various pore radii. Adapted from Amar et al.[38]

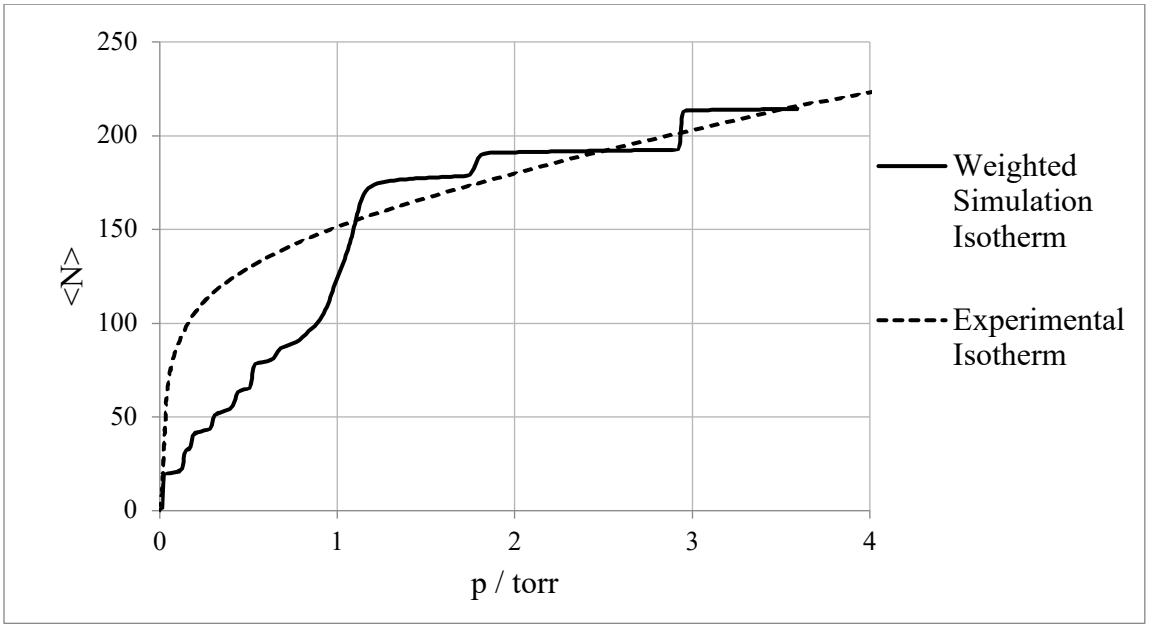

Figure 2.5. Overall isotherm from summed and weighted individual radius isotherms. Overlaid upon experimental isotherm from the same pressure range. Adapted from Amar et al.[38]

G. Computational Method

The code for the molecular Monte Carlo simulations was developed in such a way as to maximize equilibration without unnecessarily increasing computation time. The program can be broken down into runs, passes, moves, insertions, and deletions. Each run consists of one million passes. Each pass, there is one attempt to move every molecule. After ten accepted moves, every molecule is removed one at a time and the change in the system energy for removal of that molecule is calculated and binned into the removal frequency histogram (written to a text file). Then, a molecule is inserted into a random location and that change in energy is binned. Since there are an infinite number of places to insert a molecule, more insertions can be made, so in our simulations five times the number of molecules in the pore were inserted and the energy changes binned into the insertion frequency histogram. Attempts are then made once again to move molecules, and the process is repeated until one million passes have been completed. When all of this is finished for a system of a given N, a molecule is inserted into the position that was found during the insertion process to have the lowest energy, and the pass/insertion/deletion cycle is completed again. At the beginning of the next cycle, since the final molecule insertion may have upset the equilibrium, 50,000 additional passes are performed to equilibrate the configuration space. The runs begin nominally at N=0 and end at a user determined N value. To record the insertion and removal energies, bins for insertions and removals are created with limits based on estimates of the lowest possible energy of the system and its absolute value. Bins are then created at and in between those limits, incrementing by 1 K. Each time a particular insertion or removal energy is found, the event is then recorded as a 1 in its respective bin, and the sums of each bin normalized to

unity constitute the probability of the canonical system being at any given energy between the maximum and minimum. The outputs of the program (the generation of which is briefly outlined in Figure 2.6) are *energy* files, *histo* files, *m* files, *.dat* files, *meth* files, *snap* files, *.log* files, and error files. *Energy* files contain the energy of the configuration after each pass. *Histo* files contain the binned insertion and deletion energies. *M* files contain the xyz coordinates of each molecule in the pore after the millionth pass. The *.dat* files are generated during program startup, and direct the values of variables for the main program as well as directing the addition and retrieval of data into and out of the correct files. *Meth* files contain the xyz coordinates of the least energetic insertion and direct the startup coordinates of the next run. *Snap* files collect the xyz coordinates of the configuration once every 10,000 passes. The *.log* files print out various information about the run, including the pore volume, temperature, number density, and $\beta\mu_N^{ex}$ for the run calculated using the Widom insertion method. Error files generated by the operating system contained any errors encountered during the run. Representative sections of examples of *energy, histo, m, .dat, meth, log,* and *snap* files as well as the python code and .pbs script can be found in Appendix A, and the "MCPore" simulation code can be found in Appendix B.

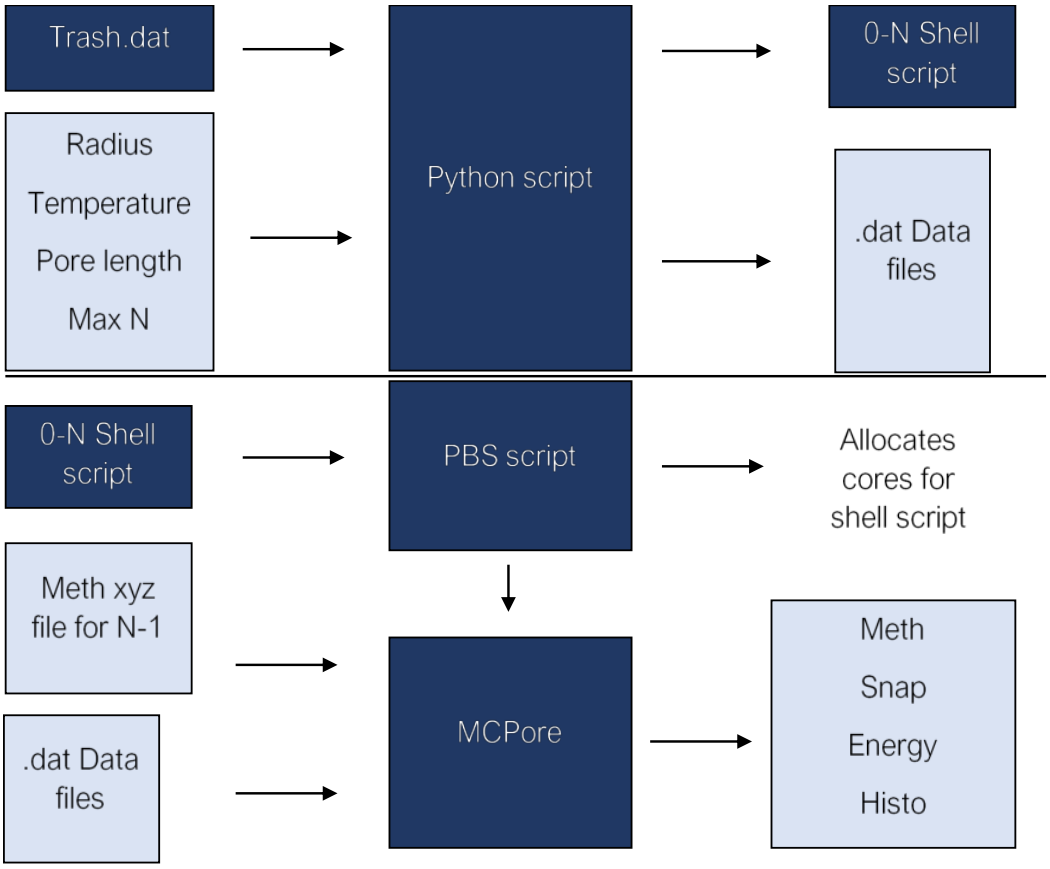

Figure 2.6. Flow chart describing process of running simulations. Adapted from Benjamin Walden.

The calculations were performed on the University of Maine Advanced Computing Group Cluster, which contains 72 Supermicro nodes and 2464 Intel Haswell/Broadwell cores at 2.5/2.4GHz. A remote connection was made via PuTTY, an SSH client, and files were retrieved either via PuTTY or via FileZilla, an FTP client.

III. RESULTS AND DISCUSSION

A. Reproducibility

Before any other analysis of the data could be done, it was necessary to gain a perspective on the reproducibility of the data between runs and between methods, i.e. the Widom insertion method and the Vaitheeswaran-Rasaiah insertion/deletion method. Two sets of data for a pore with radius 12.5 Å and temperature 77 K already existed, and two more were run. All four data sets were processed using the Widom insertion method. As shown in Figure 3.1, the values of βμ (or $\frac{1}{k_B T}$ times excess chemical potential) were very reproducible from N=0 to about N=130, at which point the statistical fluctuations

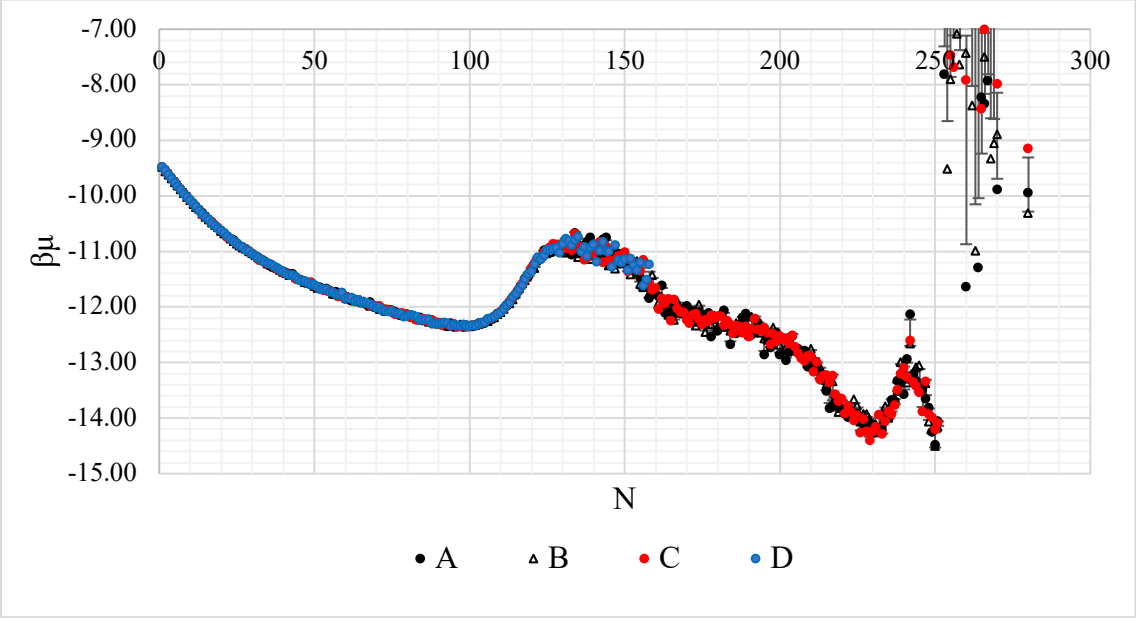

Figure 3.1. Plot of beta times the excess chemical potential vs. N for four runs at a radius of 12.5 Å, temperature 77 K, and pore length 10σ. Error bars represent one standard deviation.

increased. As the number of molecules in the pore reached N=250, however, the βμ

values varied widely. Figure 3.1 contains error bars at one standard deviation which are not visible until about N=240.

Figure 3.2A also demonstrates the high reproducibility of the data: from N=0 to about N=125, the relative standard deviation stayed below a quarter of a percent, and from N=125 to N=250 the average was about 0.75%.

As shown in Figure 3.2B, beyond N=250 the relative standard deviation quickly increased, becoming as much 12 times the value of βμ. This is because as filling increased, the number of low energy insertion locations decreased. As a result, the chance that a molecule was inserted into a high energy location increased. Because the chemical

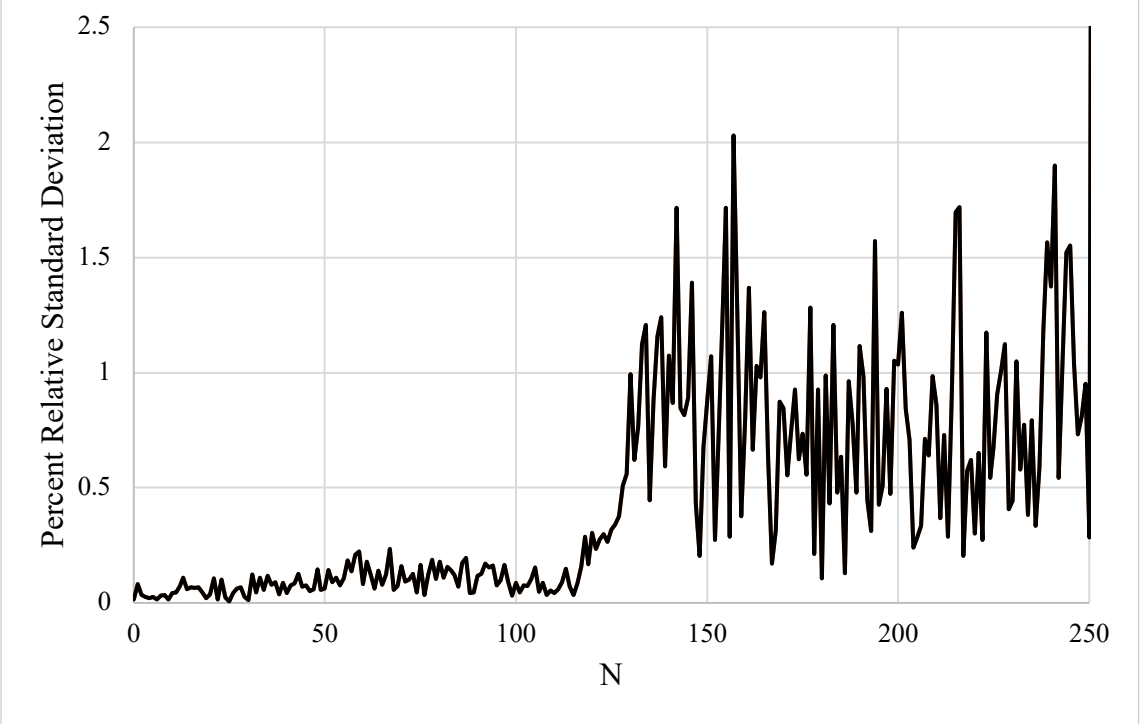

Figure 3.2A. Percent relative standard deviation vs. N for four simulation runs at pore radius 12.5 Å and 77 K.

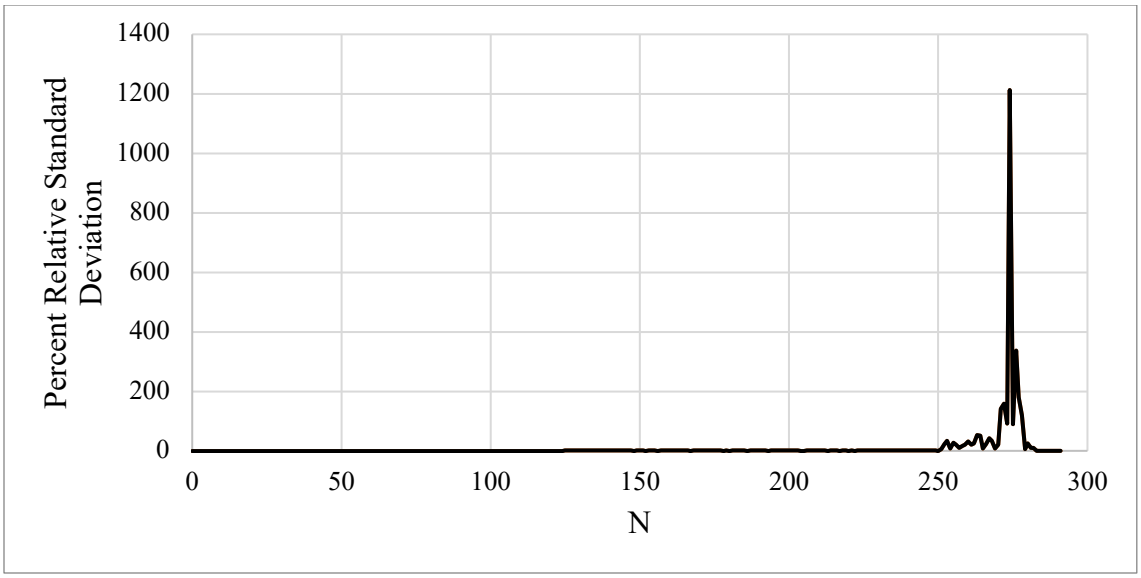

Figure 3.2B. Percent relative standard deviation up to N=300.

potential is calculated in the form $\mu = \sum U_i exp(-\beta U_i)$, the contribution to chemical potential from these high energy insertions should be negligible. However, since the total number of available insertion locations with energy below the upper limit is so low at high N, the sampling is sparse and leads to large statistical fluctuations. The insertion histograms in Figures 3.3A and B demonstrate this: at fillings significantly less than saturation, there are few high energy insertions. Near saturation, that is, at about 230 molecules, there are many more high energy insertions, but they take up a small fraction of total insertions, and therefore contribute little to the chemical potential. However, as shown in Figure 3.3B, as the pore approaches 270 molecules high energy insertions take up a higher and higher fraction of total insertions, eventually contributing strongly to the chemical potential despite their low weight. From this data it was clear that up until the βμ data begins to spike erratically, the data is consistent, but beyond that point it is not useful for analysis.

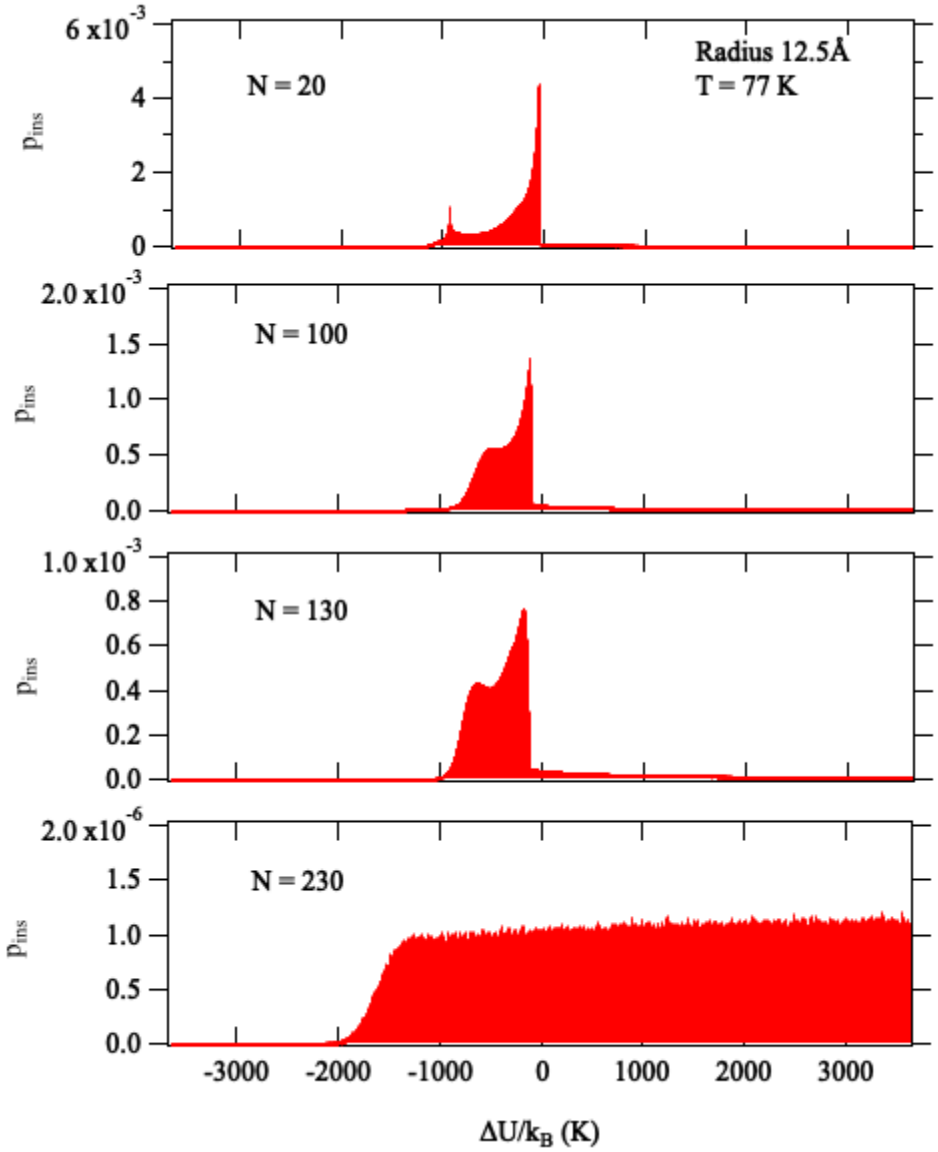

Figure 3.3A. Insertion histograms at notable filling stages, showing extremely low fraction of high energy insertions at low filling stages.

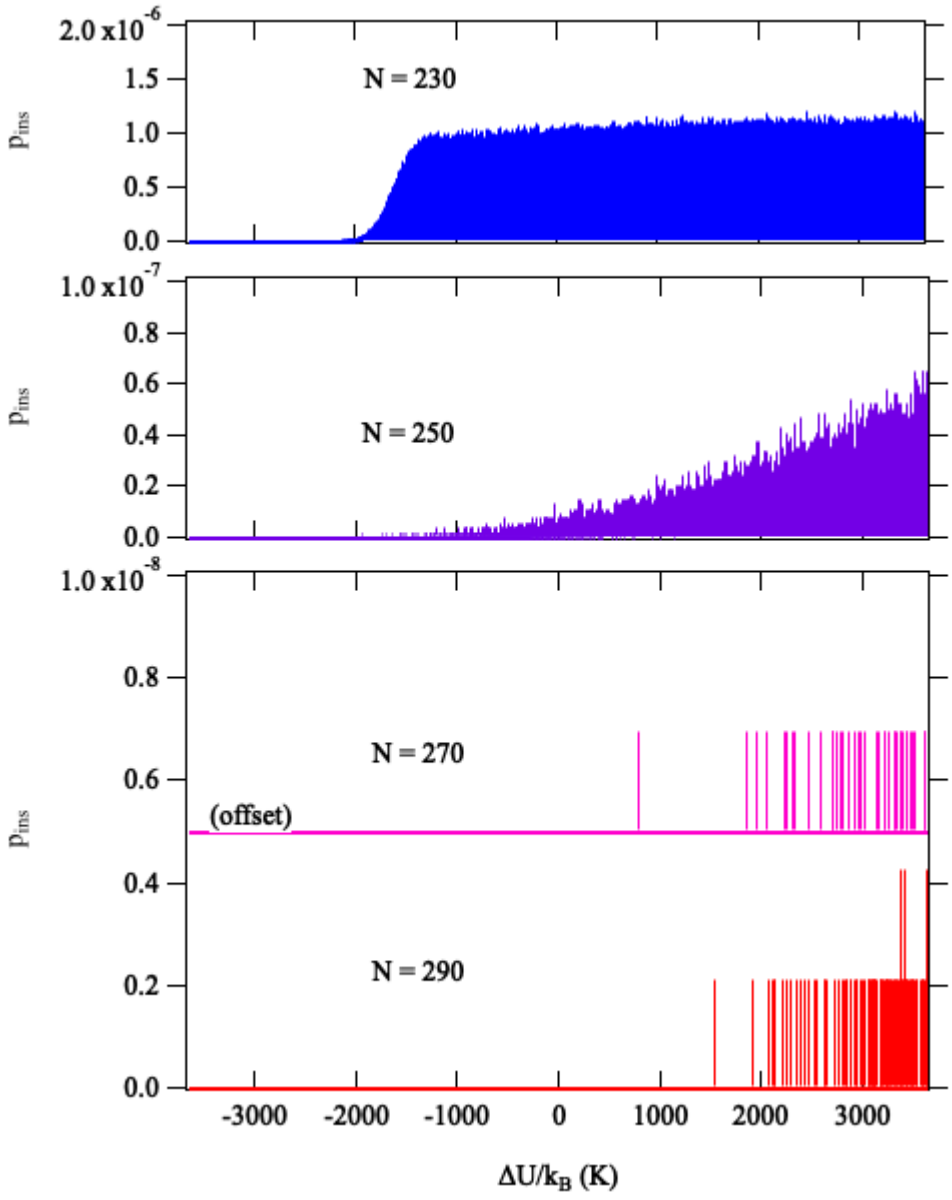

Figure 3.3B. Insertion histograms at pore fillings near saturation, demonstrating extreme bias towards high insertion energy. Data is for a pore with radius 12.5 Å at 77 K.

In addition to good reproducibility between runs using the same method of calculating the chemical potential, Figure 3.4 demonstrates that the Widom method and the Vaitheeswaran-Rasaiah method both generate data in high agreement.

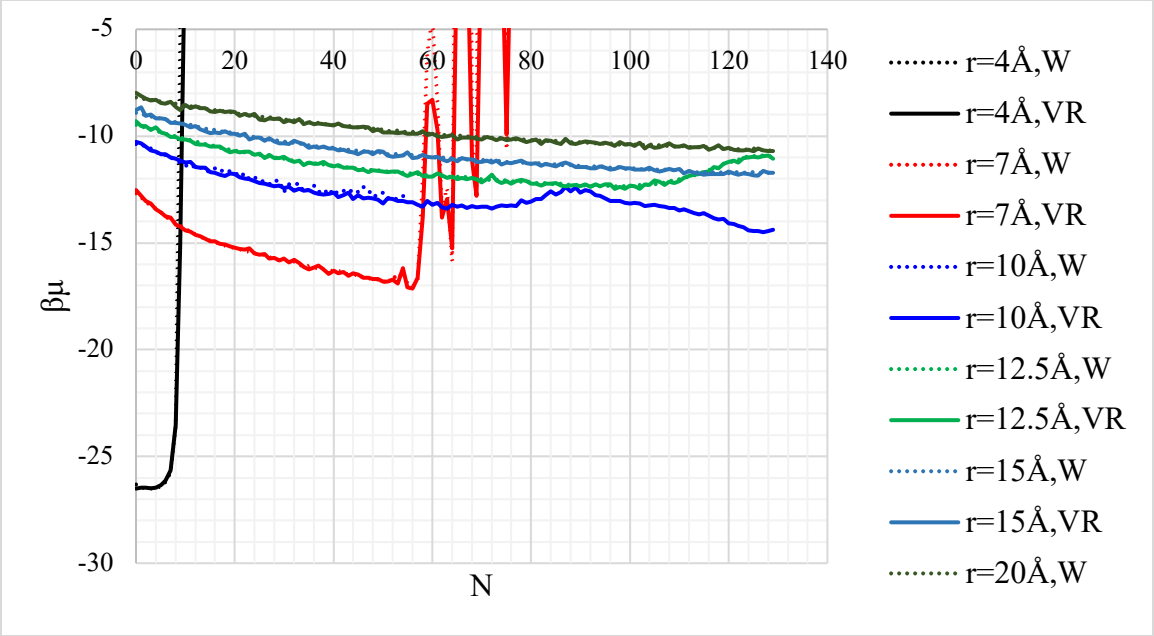

Figure 3.4. Plot of beta times excess chemical potential vs. N for various pore radii, calculated using the Widom method (W) and with the Vaitheeswaran-Rasaiah method (VR).

B. Pore Filling

As the dependence of βμ on N in Figure 3.1 shows, there is considerable structure and we now look at the factors that contribute to this dependence.

Relation of Pore Filling to Chemical Potential Trends

Our next interest was in determining the mechanism by which the pores filled, and how that process could be linked to key points in the chemical potential (or βμ) plot. As seen in Figure 3.5 below, from the initial value to most values past point E, there is a general increase. This is because relative to the slight decrease in energy from inserting a

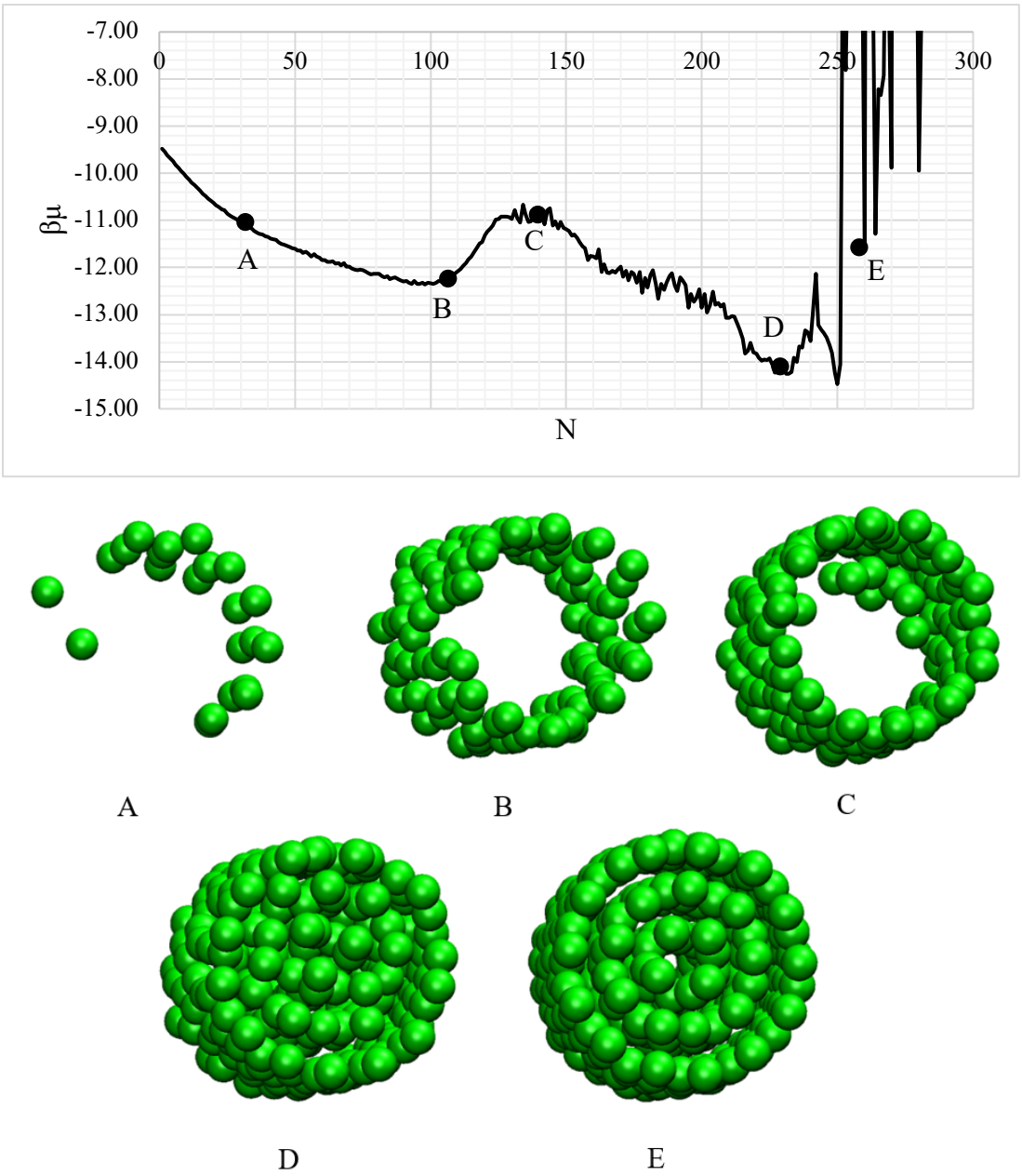

Figure 3.5. Plot of beta times excess chemical potential for a 12.5 Å radius pore at 77 K with labels at notable filling stages, together with snapshots of the pore at the indicated fillings as captured on VMD.[39]

molecule next to an empty (or almost empty, as in point A) pore wall, inserting a molecule into the highly crowded configuration space present at point E will likely result in an energy increase due to repulsive forces from closely neighboring molecules. However, Figure 3.5 also demonstrates that not only is there a significant decrease in βμ

as N is incremented slowly, but that decrease is interrupted by a local maximum at point C. By plotting the coordinates of the molecules in snap files for runs at those fillings in VMD, we were able to form a theory for the origin of the effect shown in Figure 3.5. At point A, the pore is just beginning to fill. The more combined molecule-molecule and molecule-wall forces an inserted particle can interact with, the lower its energy and the more likely its configuration. Therefore, as picture A shows, molecules will initially tend to cluster against the wall of the pore and next to other molecules. As each molecule is added, the volume of favorable interaction increases, again increasing the probability of such configurations and driving the excess chemical potential more negative. Eventually, the entire wall of the pore is covered in a layer of molecules, as shown in picture B. At this point, the molecules against the wall begin to act as if they were the wall of the pore, decreasing the chance of an inserted particle interacting simultaneously with the wall and other molecules. As a result, the average energy of insertion increases (becomes less negative). Because of this, $\beta\mu$ also becomes more positive, as shown in between points B and C. However, at this point the local maximum does not exceed the starting y value, as the pore has effectively grown smaller, increasing an inserted molecule's interaction with both sides of the pore as well as with the molecules making up the new "pseudopore." At point C, molecules begin filling in along the wall of the inner layer. This again begins increasing the potential for simultaneous molecule-molecule interactions, which causes $\beta\mu$ to become more negative. At point D, the pore has become nearly saturated, so that an inserted molecule is surrounded on all sides by attractive forces but is not so crowded as to feel repulsive forces. This configuration allows for the lowest possible insertion energy for a pore of this radius, which in turn drives the chemical potential to an overall

minimum. However, as molecules are added, the chance of interactions being entirely attractive decreases, as molecules have to be pushed closer together to fit more. By point E, only highly optimized configurations have enough space that inserting a molecule decreases the energy, as shown by picture E.

Effect of Pore Radius

Having described the pore filling process, we now present data to understand the differences in the filling of pores of different sizes. There are three dependencies of filling on pore size: the initial chemical potential, the rate at which the pore becomes saturated, and the location of local minima and maxima. As the curvature of the pore increases (or the radius decreases), a molecule feels increasing attraction not just from the wall it is tangent to but from its sides and even from across the pore's diameter. Likewise, as the curvature decreases, the molecule feels attraction decreasing to an asymptote, at which point the infinite radius pore is topologically identical to a flat plane. This behavior sets a minimum bound for the initial excess chemical potential at that of the smallest possible pore a molecule can fit in and a maximum bound at the excess chemical potential due to insertion onto a pore of infinite radius. The second dependency is trivial: the excess chemical potential becomes erratic sooner for pores of a smaller radius because it requires fewer molecules to fill the pore. The final dependency is that of the location of the local minima and maxima. As shown in Figure 3.6, on a plot of $\beta\mu$ vs. N the local minima and maxima occur later for pores of higher radii. This is because more molecules are required to fill in the space along the wall of the pore. Therefore, there exist insertion sites where a molecule can feel simultaneous molecule-molecule and

molecule-wall interaction up to a higher N. Molecules are more likely to be inserted in these low energy sites, so the average energy stays low and with it the chemical potential.

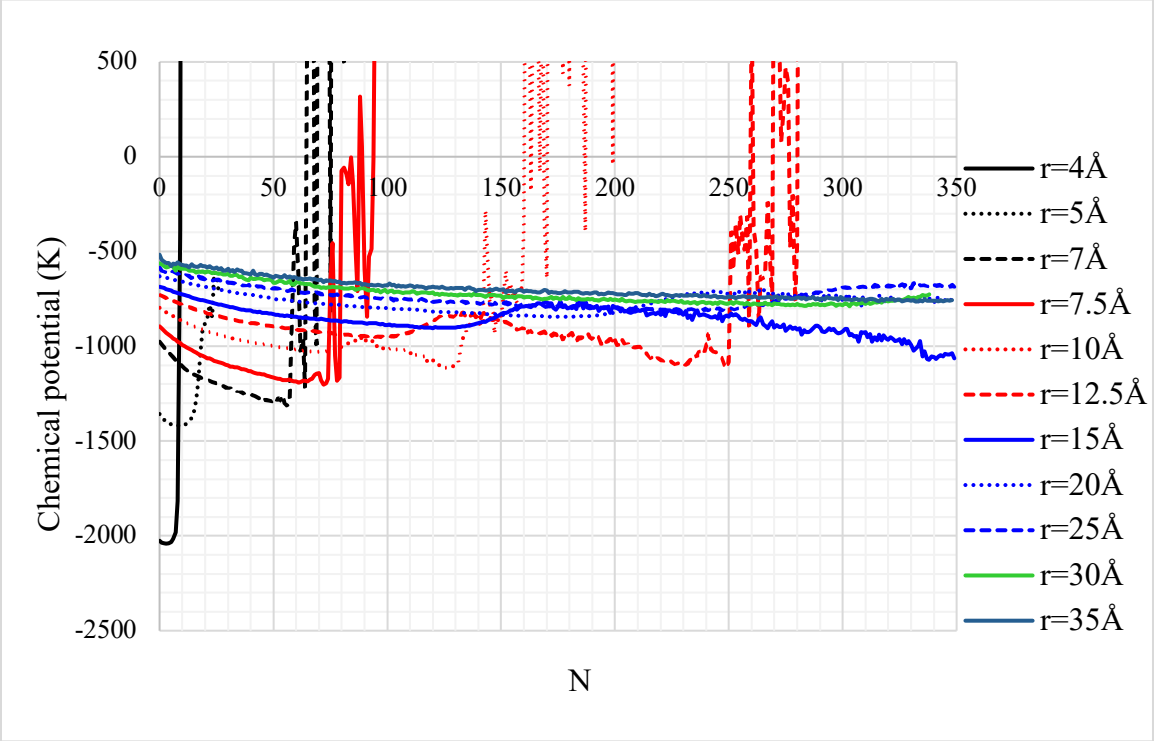

Figure 3.6. Plot of chemical potential vs. number of molecules for pores of various radii at 77K.

Mass Density Trend

Further data is presented to compare the density of molecules in the pores compared to the liquid, solid, and gas densities of methane. As shown in Figure 3.7, the density of methane in the pore does not come close to the liquid density of methane until the pore is very saturated. However, the snapshots from pore fillings significantly less than that, such as in Figure 3.5, show an organization not present in the gas phase. Therefore, the phase of the adsorbed methane cannot be positively identified as being liquid or gas.

In the plot of βμ vs. mass density, pores of a larger diameter experience a local maximum at a lower density than smaller diameter pores. This is because as the radius of the pore

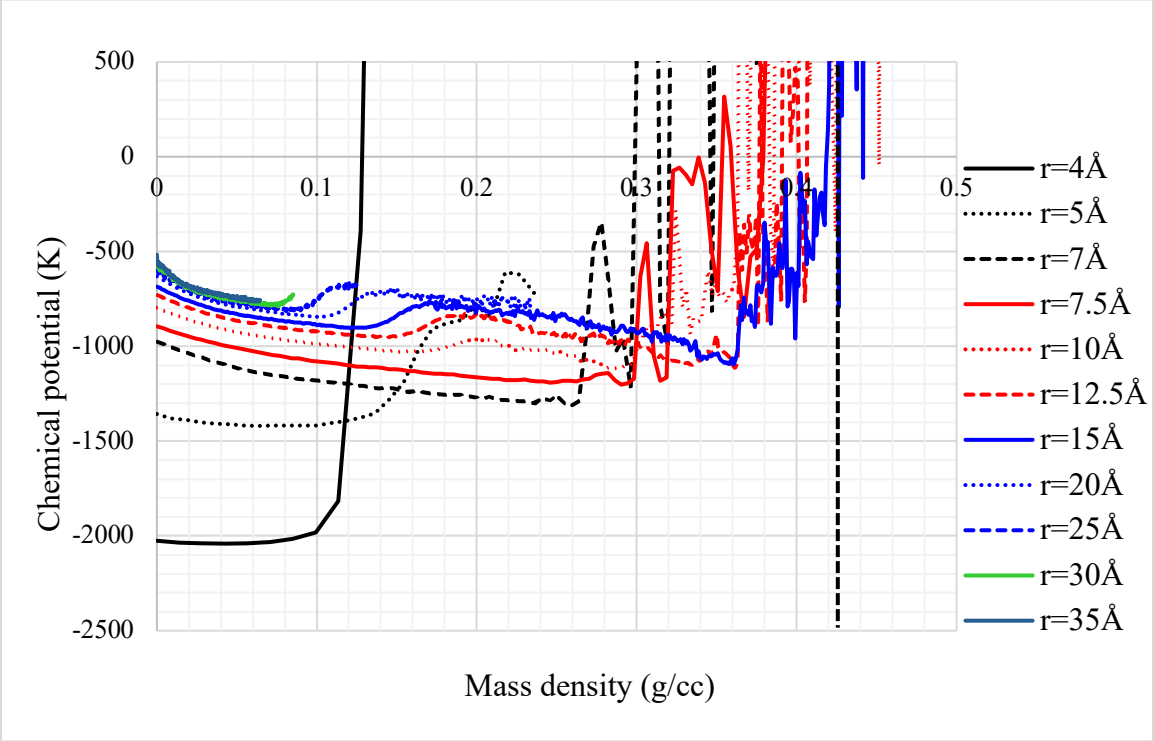

Figure 3.7. Plot of beta times excess chemical potential vs. density in gm/cc. The dashed vertical line at x=0.424 gm/cc represents the liquid density of methane.

increases, the less the volume of the molecules against the wall take up as a fraction of the total volume of the pore. Notably, this is not the case for much smaller pores, such as r = 4 Å. This is because as the total volume of a pore decreases, the efficiency with which it can be packed with spherical methane molecules also decreases. The pore quickly becomes saturated, such that inserting a molecule raises the energy and with it the chemical potential.

Effect of Temperature and Pore Length

One of the most critical parts of this project is the minimization of computing time. In general, 15 Å pores take only about half an hour to complete. However, increasing the number of molecules exponentially increases computational time, so 25 Å pores can take up to a week. Minimizing computing time both saves money and allows progress to be made much more quickly. One method we tested for minimizing computing time was reduction of the length of the pore. This method would directly reduce the number of molecules in the pore at the cost of potentially introducing greater error in the calculation of system energies. As described previously, reducing the volume of a pore decreases its sphere packing efficiency so that it becomes saturated at a smaller N relative to its volume. In addition, while U is 0.03% ϵ at 5σ, it is 0.5% ϵ at 3σ, or the cutoff tested for a shorter 6σ pore. While still low, that error would add up much more quickly, potentially affecting data. To investigate the effect of the periodic boundary condition on the excess chemical potential, simulations were run at 40, 70, and 100 K for periodic boundary lengths 6, 10, and 14 σ, using a cutoff of the Lennard-Jones potential of 3σ in the 6σ pore length and 5σ for the 10 and 14σ pores. By plotting βμ vs. N divided by the volume of the pore as shown in Figures 3.8A and B, we were able to negate the direct effects of changing the volume of the pore on the excess chemical potential, and instead elucidate the effects of changing the cutoff. The effects at 77 K and 100 K were negligible. The plots did show a shift in the chemical potential values at 40 K and length 6σ compared to 10σ. We postulate that the shift may be due to a problem with packing methanes. This could explain why it occurred at the lowest of the temperatures: the simulations at other pore radii had just enough energy to fit one more molecule in before the discontinuity

occurred. However, more investigation is needed in the matter. Figure 3.8B also shows the effect of temperature on the excess chemical potential: the higher the temperature, the more energy the molecules have to desorb, and therefore the higher the escaping tendency and the more positive the excess chemical potential.

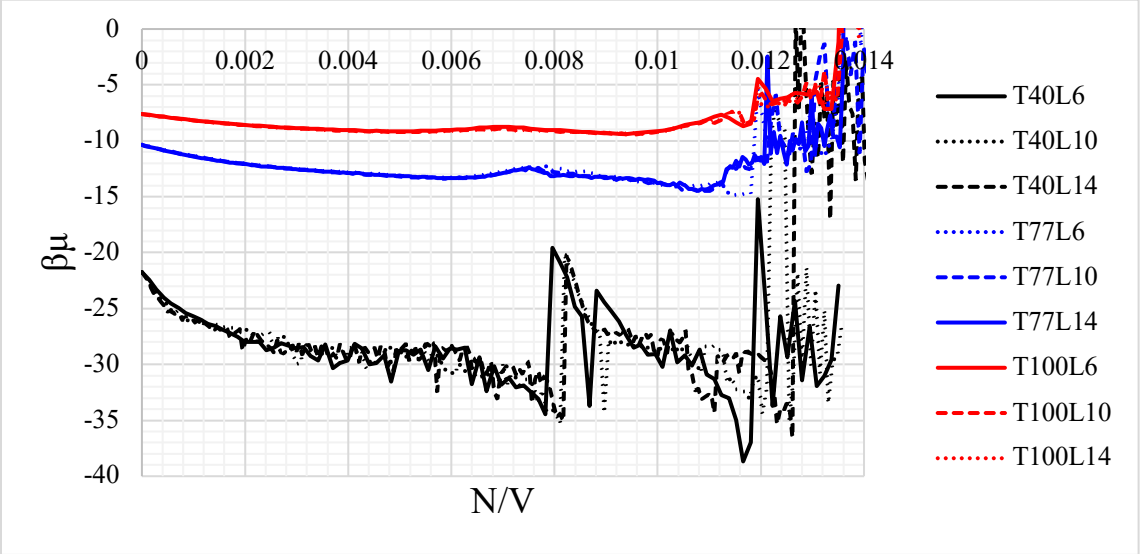

Figure 3.8A. Plot of beta times the excess chemical potential vs. N divided by the volume of the pore. Plots at temperatures 40, 77, and 100 K and for pores of length 6σ, 10σ, and 14σ, all at a radius of 10 Å.

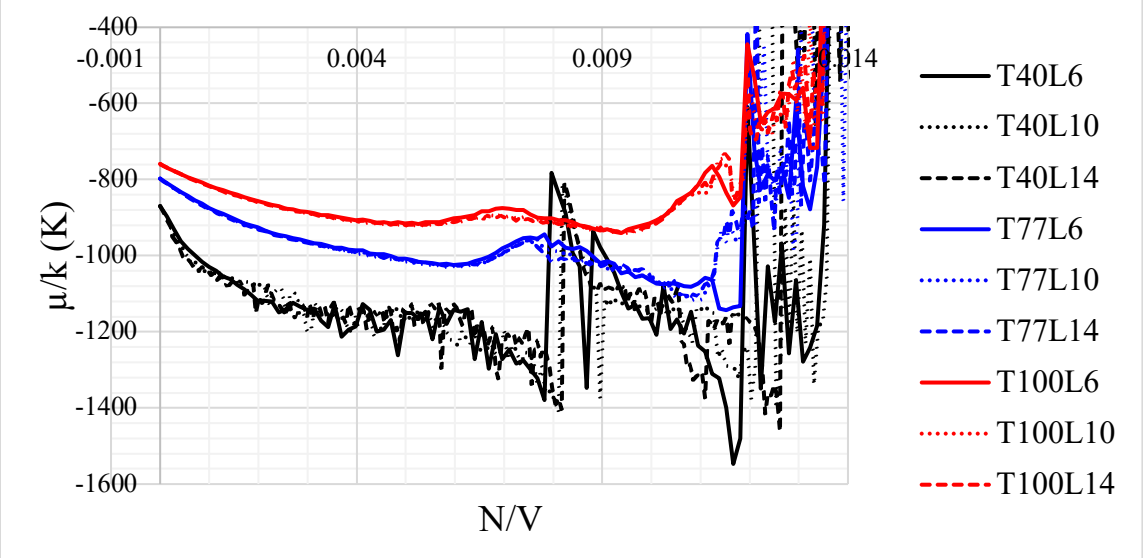

Figure 3.8B. Same data as in Figure 3.8A, but multiplied by kT to remove explicit temperature dependence.

C. Investigation of Discontinuities

As seen in the 40 K plots in Figures 3.8A and 3.8B above, during the filling process at low temperature, there is a discontinuity at N=95. Given that no other runs had previously produced such results, we attempted to probe the conditions of their origination, thereby hoping to find out more about their nature. We attempted using two techniques. First, we decreased the maximum step size from 0.2 Å to 0.1 Å and increased the percent of additional equilibration steps done at the beginning of each run from 5% to 30%. Our goal in this was to make a larger number of gradual changes in the system, reducing the probability of it jumping from one state to a largely different one. Our second technique was to start the simulator at 40 K using the xyz coordinates from a higher temperature, 77 K. The run still proceeded at its given temperature (40 K in practice) but used the extra energy from the hotter simulation to force it into a more stable configuration. Figure 3.9 shows the overlapping plots of runs at 40 K in a pore of radius 10 Å, as well as images of the pore filling at notable points. Neither reducing the step size, increasing equilibration, nor starting from a 77 K configuration smoothed out the discontinuity at N=95. On the contrary, both of the runs that used the 77 K configuration introduced discontinuous points near N=125. These points appear to lay along a continuation of the pre-discontinuity curve. This indicates that there may be a true, lower energy state below the discontinuous part of the plot, and the discontinuity brings the system to a higher metastable state. Notably, there is no visual evidence of the drastic increase in chemical potential: pictures B and C correspond to the points at the low and high points of the discontinuity, and the only discernible difference between them is the addition of one

methane. On the other hand, pictures D, E and F do show an overall change in the filling scheme that could be responsible for the discontinuity: in picture D, the molecules appear to be inserting in a cylindrical fashion inside the pseudopore formed by the molecules lining the wall. In picture F, this filling appears to be continued. In picture E, which corresponds to a downward spike in the chemical potential plot, the inner molecules appear to fill the pore in a more disorganized way. This temporary change in filling scheme may be responsible for the discontinuity, but more research needs to be done on this question and on the origin of the discontinuities overall.

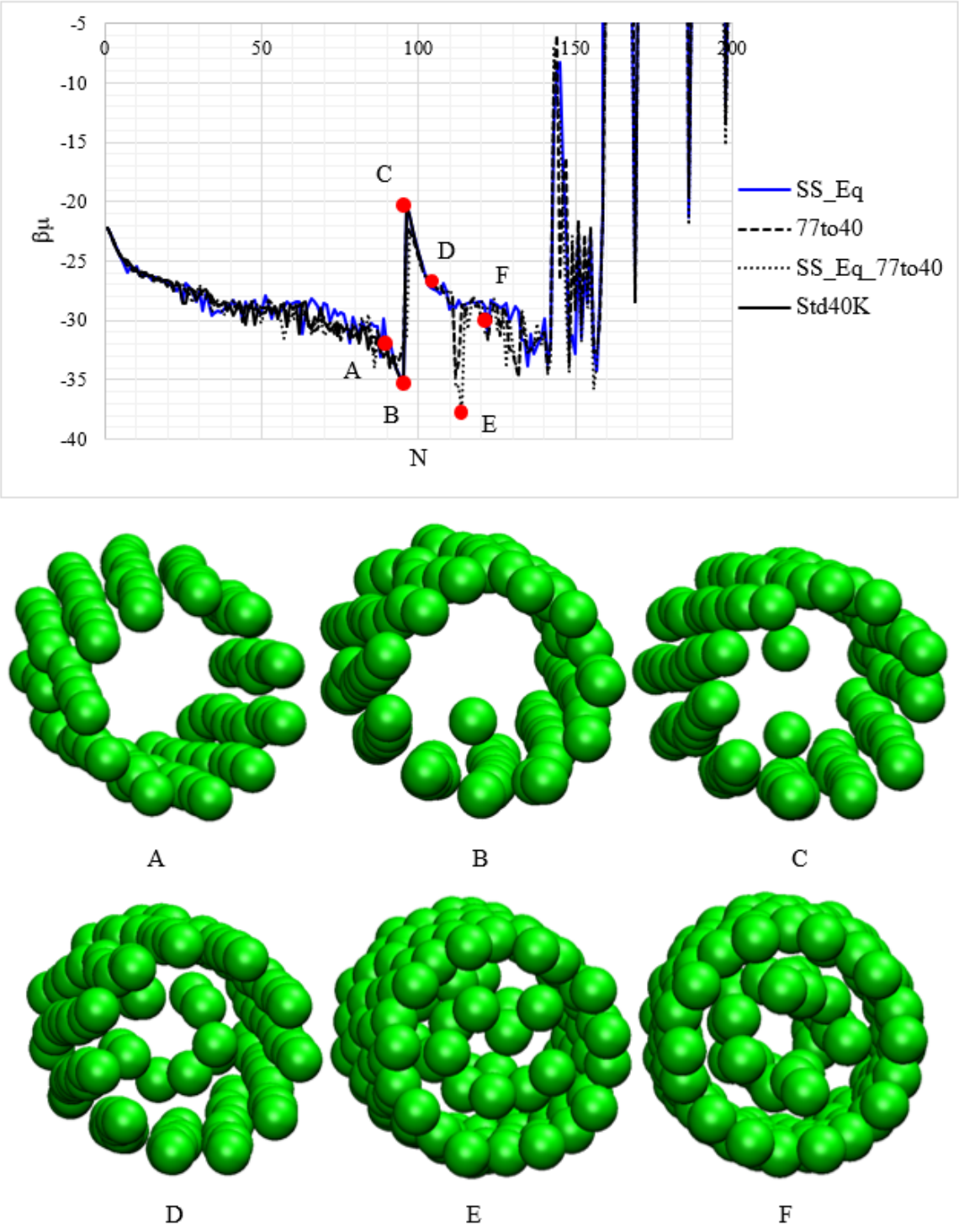

Figure 3.9. Plot of beta times excess chemical potential vs. N at 40 K. SS_Eq *signifies the run changing the stepsize and equilibrations,* 77to40 *signifies the run inputting* meth *files from a 77 K simulation,* SS_Eq_77to40 *signifies a run where both were done, and* Std40K *signifies a standard 40 K run. Pictures A-F show fillings at corresponding points on the* SS_Eq_77to40 *plot.*[39]

IV. CONCLUSION

Summary

It was determined that the reproducibility of data from identical simulations remains very good until the pore is fully filled and βμ values become erratic due to low numbers of insertion locations with energies less than the upper limit. For a run with radius 12.5 Å, temperature 77 K, and pore length 10σ, the average relative standard deviation remained less than 1% until that point. The Widom insertion method and the Vaitheeswaran-Rasaiah insertion/removal method were shown to produce data at multiple pore sizes in high agreement.

It was shown that as a pore fills up, the increased availability of insertion locations where a molecule can feel attractive effects from both the pore wall and other molecules drives down the average insertion energy, causing βμ to become more negative until the molecules form a monolayer along the wall of the pore, or "pseudopore." At this point there is a local maximum due to the separation of insertion locations from the pore wall and limited simultaneous attractive interactions, then βμ decreases again until the pore is saturated, at which point it becomes erratic.

Data showed that the local maximum due to the pseudopore occurs later for larger pore sizes in plots of βμ vs. N, as more molecules are required to complete a layer. The local maximum occurs earlier in plots of βμ vs. ρ, as a single layer of molecules against the wall accounts for less of the pore volume in larger pores. However, for extra small pores

such as 4 Å, packing inefficiency limits the density of the pore, so it is saturated much sooner than most large pores.

When investigating the effect of temperature and pore length on pore filling, it was found that βμ increases with temperature. At higher temperatures, pore cutoff length had no effect, but at 40 K, a smaller pore length caused the data to be scaled down along the y axis. It was theorized that this was because poor methane packing caused the pore to be slightly less favorable for the insertion of a molecule than the longer pores at the same N/V.

The presence of an unexpected discontinuity at N=95 in the 40 K data was investigated. The attempts made to equilibrate the data further by decreasing the maximum step size, adding additional equilibration steps, and running the simulator using coordinates from 77K did not remove the discontinuity, but did indicate more discontinuities near N=125. Seeing as these discontinuities appeared to lie on a continuation of the pre-discontinuity curve, it is suspected that the 40 K data was stuck in a metastable state.

Future Work

Understanding the origin of that discontinuity, together with decreasing the computational time required for large pores (20 Å and above) represent the two most immediate areas of research for this project. For the former, it might be wise to continue investigating conditions that lead to better equilibration. That the “77to40” approach did not diminish the effect of the metastable state in any identifiable way indicates that the overall approach of starting with a pre-equilibrated configuration may not be effective.

Instead, possibilities include “shaking the box,” or periodically making large steps to break out of a metastable state, or forcing molecules into a particular configuration, among others.

Research has already started on ways to reduce computational time: the leading idea is finding a way to skip values of N, only calculating the data that is needed for the isotherm to remain accurate. The idea of testing how tightly packed a pore should be before ending the simulation should also be pursued in its own right. This would help investigate how high insertion energies must be before their contributions to chemical potential are negligible and omitting them does not affect the accuracy of the calculated isotherm.

Once these problems are solved, provided there are no other unforeseen bugs in the code, there should be no obstacle preventing the rest of the simulations from being run. Once all of the simulations are run, a full isotherm can be constructed by averaging isotherms over the pore size distribution of SBA-15 and compared to the experimental data. If the data agree satisfactorily, molecular dynamics simulations can begin with the goal of describing diffusion within SBA-15’s micro- and mesoporous structure.

LIST OF REFERENCES

1. *October 2020 Monthly Energy Review;* U.S. Department of Energy, U.S. Government Printing Office: Washington, DC, 2020.

2. Ahlbrandt, T. S.; McCabe, P. J. Global Petroleum Resources: A View to the Future. *Geotimes* [Online] **2002**, *47*, 14-18.

3. Wang, S.; Li, Z.; Yi, W.; Fu, P.; Zhang, A.; Bai, X. Renewable aromatic hydrocarbons production from catalytic pyrolysis of lignin with Al-SBA-15 and HZSM-5: Synergistic effect and coke behavior. *Renewable Energy* **2021**, *163*, 1673-1681.

4. Kumar, L.; Bharadvaja, N. A review on microalgae biofuel and biorefinery: challenges and way forward. In *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* [Online], Taylor and Francis Group. https://doi.org/10.1080/15567036.2020.1836084

5. Fu, C.; Mielenz, J. R.; Xiao, X.; Ge, Y.; Hamilton, C. Y.; Rodriguez, Jr., M.; Chen, F.; Foston, M.; Ragauskas, A.; Bouton, J.; Dixon, R. A.; Wang, Z. Genetic manipulation of lignin reduces recalcitrance and improves ethanol production from switchgrass. *PNAS* **108**, *9*, 3803-3808.

6. Chen, B.; Xu, G.; Chang, C.; Zheng, Z.; Wang, D.; Zhang, S.; Li, K.; Zou, C. Efficient One-Pot Production of Biofuel 5-Ethoxymethylfurfural from Corn Stover: Optimization and Kinetics. *Energy Fuels* **2019**, *33*, 4310-4321

7. DeSisto, W. J.; Hill II, N.; Beis, S. H.; Mukkamala, S.; Joseph, J.; Baker, C.; Ong, T.-H.; Stemmler, E. A.; Wheeler, M. C.; Frederick, B. G.; van Heiningen, A. Fast Pyrolysis of Pine Sawdust in a Fluidized-Bed Reactor. *Energy Fuels* **2010**, *24*, 2642-2651.

8. Ro, D.; Kim, Y-M.; Lee, I-G.; Jae, J.; Jung, S-C.; Kim, S. C.; Park, Y-K. Bench scale catalytic fast pyrolysis of empty fruit bunches over low cost catalysts and HZSM-5 using a fixed bed reactor. *J. Clean. Prod.* **2018**, *176*, 298-303.

9. Erdiwansyah; Mamat, R.; Sani, M. S. M.; Sudhakar, K.; Kadarohman, A.; Sardjono, R. E. An overview of Higher alcohol and biodiesel as alternative fuels in engines. *Energy Reports* **2019**, *5*, 467-479.

10. Liu, S-N.; Cao, J-P.; Zhao, X-Y.; Wang, J-X.; Ren, X-Y.; Yuan, J-S.; Guo, Z-X.; Shen, W-Z.; Bai, J.; Wei, X-Y. Effect of zeolite structure on light aromatics formation during upgrading of cellulose fast pyrolysis vapor. *J. Energy Inst.* **92**, 1567-1576.

11. Pattiya, A. Catalytic pyrolysis. In: *Direct Thermochemical Liquefaction for Energy Applications*. Woodhead Publishing: Cambridge, England, 2018; pp 29-64.

12. Joseph, J.; Rasmussen, M. J.; Fecteau, J. P.; Kim, S.; Lee, H.; Tracy, K. A.; Jensen, B. L.; Frederick, B. G.; Stemmler, E. A. Effect of zeolite structure on light aromatics formation during upgrading of cellulose fast pyrolysis vapor. *Energy Fuels* **30**, 4825-4840.

13. Verma, P.; Kuwahara, Y.; Mori, K.; Raja, R.; Yamashita, H. Functionalized mesoporous SBA-15 silica: recent trends and catalytic applications. *Nanoscale* **2020**, *12*, 11333-11363.

14. Ghampson, I.T.; Newman, C.; Kong, L.; Pier, E.; Hurley, K.D.; Pollock, R.A.; Walsh, B.R.; Goundie, B.; Wright, J.; Wheeler, M.C.; Meulenberg, R.W.; DeSisto, W.J.; Frederick, B.G.; Austin, R.N. Effects of pore diameter on particle size, phase, and turnover frequency in mesoporous silica supported cobalt Fischer–Tropsch catalysts. *App. Catal. A: General* **2010**, *388*, 57-67.

15. Dou, B.; Zhao, L.; Zhang, H.; Wu, K.; Zhang, H. Renewable hydrogen production from chemical looping steam reforming of biodiesel byproduct glycerol by mesoporous oxygen carriers. *Chem. Eng. J.* **2020**, https://doi.org/10.1016/j.cej.2020.127612.

16. Gök, Y.; Gök, H.Z. Synthesis, characterization and catalytic performance in enantioselective reactions by mesoporous silica materials functionalized with chiral thiourea-amine ligand. *Res. Chem. Intermed.* **2020**, https://doi-org.wv-o-ursus-proxy02.ursus.maine.edu/10.1007/s11164-020-04301-w.

17. Kruk, M.; Jaroniec, M; Ko, C. H.; Ryoo, R. Characterization of the Porous Structure of SBA-15. *Chem. Mater.* **2000**, *12*, 1961-1968.

18. Karger, J.; Ruthven, D. M.; *Diffusion in Zeolites and Other Microporous Solids*. John Wiley: New York, USA, 1992.

19. Ruthven D.M.; Brandani S.; Eic M. Measurement of Diffusion in Microporous Solids by Macroscopic Methods. In: Karge H.G., Weitkamp J. (eds) *Adsorption and Diffusion. Molecular Sieves*. Springer: Heidelberg, Berlin, 2005; Vol 7; pp 45-84.

20. Martin, C.; Coulomb, J. P.; Ferrand, M. Direct measurement of the translational mobility of deuterium hydride molecules confined in a model microporous material: $AlPO_4$-5 zeolite. *Europhys. Lett.* **1996**, *36* (7), 503-508.

21. Jobic, H.; Theodorou, D. N. Quasi-elastic neutron scattering and molecular dynamics simulation as complementary techniques for studying diffusion in zeolites. *Microporous Mesoporous Mat.* **2007**, *102*, 21-50.

22. Pollock, R. A. Neutron Scattering Studies of Structure, Hydrothermal Stability, and Transport in Porous Silica Catalyst Supports. Physics, PhD Thesis, The University of Maine, Orono, ME, 2012.

23. Hoang, V.-T.; Huang, Q,; Ei, M.; Do, T.-O.; Kaliaguine, S. Structure and Diffusion Characterization of SBA-15 Materials. *Langmuir* **2005**, *21*, 2051-2057.

24. Pollock, R. A.; Walsh, B.R.; Fry, J.; Ghampson, I.T.; Melnichenko, Y.B.; Kaiser, H.; Pynn, R.; DeSisto, W.J.; Wheeler, M.C.; Frederick, B.G. Size and Spatial Distribution of Micropores in SBA-15 using CM-SANS. *Chem. Mater.* **2011**, *23*, 3828-3840.

25. Saito, A.; Foley, H.C. Curvature and parametric sensitivity in models for adsorption in micropores. *AIChE J.* **1991**, *37* (3), 429-436.

26. Horvath, G.; Kawazoe, K. Method for the calculation of effective pore size distribution in molecular sieve carbon. *J. Chem. Eng. Jpn.* **1983**, *16* (6), 470-475.

27. Imperor-Clerc, M.; Davidson, A. Existence of a Microporous Corona around the Mesopores of Silica-Based SBA-15 Materials Templated by Triblock Copolymers. *J. Am. Chem. Soc.* **2000**, *122*, 11925-11933.

28. Bhattacharya, S.; Coasne, B.; Hung, F. R.; Gubbins, K. E. Modeling Micelle-Templated Mesoporous Material SBA-15: Atomistic Model and Gas Adsorption Studies. *Langmuir* **2009**, *25(10)*, 5802-5813.

29. Neimark, A. V.; Ravikovitch, P. I. Capillary condensation in MMS and pore structure characterization. *Microporous and Mesoporous Mat.* **2009**, *44-45*, 697-707.

30. Levine, I. Statistical Mechanics. In *Physical Chemistry*, McGraw-Hill, 2017.

31. Goodbody, S. J.; Watanabe, K.; MacGowan, D.; Walton, J. P. R. B.; Quirke, N. J. Molecular Simulation of Methane and Butane in Silicalite. *Chem. Soc. Faraday Trans*., **1991**, *87 (13)*, 1951-1958.

32. Tjatjopoulos, G. J.; Feke, D. L.; Mann, Jr, J. A. Molecule-Micropore Interaction Potentials. *J. Phys. Chem*. **1988**, *92*, 4006-4007.

33. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys*. **1953**, *21 (6)*, 1087–1092.

34. McQuarrie, D. A. *Statistical Mechanics*, 55, University Science Books, 2000.

35. Widom, B. Some Topics in the Theory of Fluids. *J. Chem. Phys.* **1963**, *39(11)*, 2808-2812.

36. Frenkel, D.; Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, 2002.

37. Vaitheeswaran, S.; Rasaiah, J. C.; Hummer, G. Electric field and temperature effects on water in the narrow nonpolar pores of carbon nanotubes. *J. Chem. Phys.* **2004**, *121 (16)*, 7955-7965.

38. Amar, F. G.; Frederick, B. G.; York, K.; Pollock, R. Modeling the filling of methane in heterogeneous pore networks, 248th National Meeting of the American Chemical Society, San Francisco, CA, **2014**.

39. These images were made with VMD. VMD is developed with NIH support by the Theoretical and Computational Biophysics group at the Beckman Institute, University of Illinois at Urbana-Champaign. Homepage: http://www.ks.uiuc.edu/Research/vmd/.

APPENDICES

APPENDIX A: EXAMPLES OF DATA AND LOG FILES

Energy File

0 -16904.014924861374

1 -16743.455375514255

2 -16708.814138922076

3 -16622.332052873830

4 -16580.804670566584

5 -16536.159564734655

.

.

.

1049996 -15923.084331101307

1049997 -15983.587634435244

1049998 -15830.107914543976

1049999 -15732.116927332801

1050000 -15659.132595792897

Histo File

```
#        10   5.0000000000000000       37.299999999999997       77.000000000000000
  -4696.0000000000000       0.0000000000000000       0.0000000000000000
  -4695.0000000000000       0.0000000000000000       0.0000000000000000
  -4694.0000000000000       0.0000000000000000       0.0000000000000000
  -4693.0000000000000       0.0000000000000000       0.0000000000000000
.
.
.
  -1494.0000000000000       2221.0000000000000       2164.0000000000000
  -1493.0000000000000       2210.0000000000000       2121.0000000000000
  -1492.0000000000000       2251.0000000000000       2105.0000000000000
  -1491.0000000000000       2202.0000000000000       2147.0000000000000
  -1490.0000000000000       2105.0000000000000       2128.0000000000000
.
.
.
   4691.0000000000000       0.0000000000000000       189.00000000000000
   4692.0000000000000       0.0000000000000000       163.00000000000000
   4693.0000000000000       0.0000000000000000       172.00000000000000
   4694.0000000000000       0.0000000000000000       179.00000000000000
   4695.0000000000000       0.0000000000000000       9910624.0000000000
```

m File

```
        10
 5.0000000000000000       77.000000000000000         1000000
Me   1.3222774459562574      -1.3165452317492694       17.493157650234991
Me  -1.2628104597645180      -1.4196769986095359       11.511658755669298
Me  -1.3631914512007739       0.99473208135338043     -17.376584145410689
Me   0.96092485159687036      1.5987845844127777      -10.994712527419955
Me  -1.3747842753646409       0.91322513217827384     -2.5306620970833554E-002
Me   0.26751224354140934     -1.8353619372117111      -13.260013409564232
Me  -1.4519078517389228       0.95083925515740786      15.529285372652300
Me  -1.5876894513029944      -0.24535050880586026     -4.0633198192195081
Me   1.6469576531365240      -0.62885952256208921     -7.1907631413305602
Me   0.87905881857797297      1.7685901218436204       8.4619768157123865
```

.dat File

```
meth00010rad5t77l10.xyz     ! input #methanes; radius of pore in Ang;length of pore in sigmeth units label (10)
m00010r5t77l10.xyz          ! restart file write over every pass (20)
meth00011rad5t77l10.xyz     ! start file n+1 meth from minimum insert en.(25)
energym00010r5t77l10.txt    ! output final data energy record filename (30)
histo00010r5t77l10.txt      ! output final histogram data (35)
snap00010-5-77-10.xyz       ! snap   movie file name (40)wr
16.042                      ! mass
147.9                       ! epsmeth
3.73                        ! sigmeth
133.3                       ! epspore
3.21                        ! sigpore
0.153                       ! rhopore
1                           ! iranstate 0=default seed, 1=random time-based seed
5.00                        ! radpore in angstroms
2.5                         ! rporecut  in Angtrom radpore-rporecut defines free Vol
10.00                       ! porelength in units of sigmeth
77.00                       ! temp    in degrees K
5                           ! cutoff    in sigmeth units   error<.1%
1000000                     ! npass    #of passes
0.05                        ! equil   # of equilibration passes as fract of npass
5                           ! insertfactor
100                         ! ncalib  parameter to recalculate energytot
20                          ! deltest parameter to select move for binning
10000                       ! nsnap   print movie step parameterc
0.2                         ! stepmax        in angstroms
```

Meth File

```
        10
 5.0000000000000000      77.000000000000000      -16904.015625000000
Me   1.6100030126020899     -0.70058397817597839     -15.999790995534877
Me  -1.7212594608161844     -0.39765920857091092      13.973561429122205
Me   1.5173461109371786      0.52766168447801987      7.2449407600727840
Me  0.79525552482010220      -1.5344502477466491     -11.351110487356912
Me  -1.3128132879991226      -1.1896360240579376      9.8730600415194694
Me -0.92684119186698433       1.3092433223385527     -9.0143520720857691
Me  -1.4897446389779525      0.82836159249945851     -18.513888700574128
Me  -1.6458252164566483      0.53582121582674991     -13.819942250037546
Me   1.5063810443782759      0.89585669881760344      12.024722835869323
Me   1.6310370259180202      0.13152313650518777      16.524874938781164
```

.log File

.

.

.

nmol= 00009 done

iranstate 1 initial rand no= 0.429346383

-2190.6062216112582 -14713.408703250116 -16904.014924861374

-1460.4726198143226 -4695.74512 4695.74512 -4696

-4696 4695 ntestbin= 9391

9391 -4696.0000000000000 4694.0000000000000 4695.0000000000000

npass= 1000000 nequil= 50000 deltest= 20

unnormalized sum of chempotreg+chempothigh 1439368901689117.8

unnormalized sum of chempotreg 1439368901689117.8

unnormalized sum of chempothigh 7.1172908612726737E-032

Temp= 77.000000000000000 beta= 1.2987012987012988E-002

inssample= 3709285

chempotreg= 388044839.28550053

testbin(3,ntestbin+1) = 9910624.0000000000

chempothigh= 0.00000000

totsample = 13619909

chempot= 105681242.19399101

N= 10 betamu1= -18.475937972405166 volpore= 2929.5352309942241

numdens= 3.4135107488044119E-003 density= 9.0807777024156114E-002

mubye= -9.6189805535848389

nlowbin= 0

mubye1 = -9.6189805535848389

lambda= 0.49684132074025478 volpore= 2929.5352309942241 muidbye= -4.0000211598322757

1050000 -15659.132719200879 10500000 7279824

nmol= 00010 done …

Snap File

```
        10
 Initial energy=   -16904.014924861374
 Me   1.6100030126020899      -0.70058397817597839       2.6502090044651219
 Me  -1.7212594608161844      -0.39765920857091092       32.623561429122205
 Me   1.5173461109371786       0.52766168447801987       25.894940760072782
 Me  0.79525552482010220       -1.5344502477466491       7.2988895126430862
 Me  -1.3128132879991226       -1.1896360240579376       28.523060041519468
 Me -0.92684119186698433        1.3092433223385527       9.6356479279142295
 Me  -1.4897446389779525       0.82836159249945851       0.13611129942587041
 Me  -1.6458252164566483       0.53582121582674991       4.8300577499624531
 Me   1.5063810443782759       0.89585669881760344       30.674722835869321
 Me   1.6310370259180202       0.13152313650518777       35.174874938781159
.
.
.
        10
    1050000  -15659.132595792897
 Me   1.3222774459562574      -1.3165452317492694       17.493157650234991
 Me  -1.2628104597645180      -1.4196769986095359       11.511658755669298
 Me  -1.3631914512007739       0.99473208135338043      -17.376584145410689
 Me  0.96092485159687036        1.5987845844127777      -10.994712527419955
 Me  -1.3747842753646409       0.91322513217827384      -2.5306620970833554E-002
 Me  0.26751224354140934       -1.8353619372117111      -13.260013409564232
 Me  -1.4519078517389228       0.95083925515740786       15.529285372652300
 Me  -1.5876894513029944      -0.24535050880586026      -4.0633198192195081
 Me   1.6469576531365240      -0.62885952256208921      -7.1907631413305602
 Me  0.87905881857797297        1.7685901218436204       8.4619768157123865
```

.pbs Script

```
#PBS -q batch

#PBS -N single-core-program

#PBS -l procs=1

#PBS -l walltime=144:00:00

cd $PBS_O_WORKDIR

module load gcc/7.3.0

module load netcdf/gnu-5.4.0-4.4.3-fortran

time ./runmcporeV10.sh >> runmcpore.log
```

Python Code

```
        #to run the script, must have the .py file and .dat file in same folder
#(the working folder for your simulations) and type in the terminal----
#  python WriteDatFileV10_R15T77L10.py mcpore_trash.dat



#from sys import argv #import test.dat
#script, filename = argv

import subprocess

i = int(raw_input("input start number of molecules: "))
x = i


nmolfinal= int(raw_input("input final number of molecules: "))

radpore= raw_input("input pore radius in angstroms: ")
radporefloat= float(radpore)

temp= raw_input("input temperature in Kelvins: ")
tempfloat= float(temp)

porelen= raw_input( "input pore length in sigmeth units: ")
porelenfloat= float(porelen)



# float will round the number when %.0f is used, if number decimal >= .5 rounds up if <.5 rounds down
while i < nmolfinal +1:
  k=10+i
```

```
  j=i+1



  filename = "mcporeV10_r%.0f_%.0f_%.0f-%05d.dat" % (radporefloat, tempfloat, porelenfloat,
i)#change file name


  target = open(filename, 'w')



  target.truncate()

  #Change the filenames for each different set of radius/temperature simulation.

  line1 = "meth%05drad%.0ft%.0fl%.0f.xyz     ! input #methanes; radius of pore in Ang;length of pore in
sigmeth units label (10)" % (i, radporefloat, tempfloat, porelenfloat) #change file name

  line2 = "m%05dr%.0ft%.0fl%.0f.xyz                    ! restart file write over every pass (20)" % (i,
radporefloat, tempfloat, porelenfloat) #change file name

  line3 = "meth%05drad%.0ft%.0fl%.0f.xyz     ! start file n+1 meth from minimum insert en.(25)" % (j,
radporefloat, tempfloat, porelenfloat)#change file name

  line4 = "energym%05dr%.0ft%.0fl%.0f.txt! output final data energy record filename (30)" % (i,
radporefloat, tempfloat, porelenfloat)#change file name

  line5 = "histo%05dr%.0ft%.0fl%.0f.txt      ! output final histogram data (35)" % (i, radporefloat,
tempfloat, porelenfloat)#change file name

  line6 = "snap%05d-%.0f-%.0f-%.0f.xyz       ! snap   movie file name (40)wr" % (i, radporefloat,
tempfloat, porelenfloat) #change file name

  line7 = "16.042                    ! mass"

  line8 = "147.9                     ! epsmeth"

  line9 = "3.73                      ! sigmeth"

  line10 = "133.3                    ! epspore"

  line11 = "3.21                     ! sigpore"

  line12 = "0.153                    ! rhopore"

  line13 = "1                              ! iranstate 0=default seed, 1=random time-based seed"

  line14 = "%.02f                ! radpore in angstroms" % radporefloat #change this

  line15 = "2.5                     ! rporecut  in Angtrom radpore-rporecut defines free Vol"

  line16 = "%.02f                ! porelength in units of sigmeth" % porelenfloat #change this

  line17 = "%.02f                ! temp    in degrees K" % tempfloat #change this

  line18 = "5                              ! cutoff    in sigmeth units   error<.1%" #Hardcoding cut off =3
only for l=6 if l > 10 cutoff stays at 5
```

```
  line19 = "1000000                    ! npass    #of passes"
  line20 = "0.05                  ! equil   # of equilibration passes as fract of npass"
  line21 = "5                          ! insertfactor"
  line22 = "100                  ! ncalib  parameter to recalculate energytot"
  # for less than 90 molecules:
  line23 = "%d                     ! deltest parameter to select move for binning" % k
          # for >90 molecules
  #line23 = "100                 ! deltest parameter to select move for binning"
  line24 = "10000                    ! nsnap   print movie step parameterc"
  line25 = "0.2                    ! stepmax          in angstroms"

 #  write the file

  target.write(line1)
  target.write("\n")
  target.write(line2)
  target.write("\n")
.
.
.
  target.write(line25)
  target.write("\n")

  target.close()

  i=i+1


f= open('runmcporeV10.sh', 'w')
string0 = "#! /bin/bash"
f.write(string0 + "\n")

#print (string0)
```

```
string3 = "echo file done"



while x < nmolfinal +1:


   filename = "mcporeV10_r%.0f_%.0f_%.0f-%05d.dat" % (radporefloat, tempfloat, porelenfloat,
x)#change file name

# edit next line to change mcpore version

   string = "mcporeV12b < " + filename

   #subprocess.call(string, shell=True)

   nx=str(x)

   string2 = "echo nmol= %05d done" % x

   #subprocess.call(string2, shell=True)


   #print(string)

   #print(string2)


   f.write(string + "\n"+ string2 + "\n")


   x = x+1


f.write(string3)

f.close()

print(string3)
```

APPENDIX B: MCPORE CODE (V12b)

```
      program mcpore
c This program calculates the potential energy histograms of a model of methane
c in a zeolite pore. The model of  XXX is used.
c
c mcporeV2: Added May 9, 2012:  code to calculate removal and insertion histograms
c following the Vaitheeswaran/Rasaiah/etal procedure
c
c mcporeV4: Added June 2, 2012:
c 1 code to do uniform sampling of insertion in pore
c 2 uniform lower bound on removal and insertion bins
c 3 storage of high energy insertion counts in the last bin of testbin(3,max)
c 4 a restart file for n+1 methanes using the lowest stored insertion energy
c   from the current run
c
c mcporeV6: added June 19, 2012: compute chemical potential from simple
c Widom method before printing histogram
c
c mcporeV7: added June 22-25, 2012:  print vpore vs r data and calculate
c chemical potential with two versions of free volume.
c
c mcporeV8: added July 9, 2012:  final choice of free volume and some changes in
c units of input parameters
c
c mcporeV9: added July 17, 22012:  switch to calculate insertion energy
c for the empty cavity.
c
c mcporeV10:  Feb. 26, 2013 removed a line that was commented out that
```

```
c  multiplied radpore by sigmeth. This line had the comment removed
c  inadvertently to give an error during Jan. 2013.This version based
c  mcporeV9_bgf.f that R Pollock gave to B. Frederick around Dec 20, 2012.
c  also added some  !comments specifying reads and conversions of porelength
c  and cutoff variables etc.
c
c mcporeV11: Jan 21, 2021 :  fixed inssample and remsample variables
c so they are properly declared. Added some error checking that cutoff is
c not greater than porelength/2. Rewrote the in-code calculation of chempot
c to calculate chempotreg by summing the exp(-betaU) term in the regular bins
c and summing chempothigh for the energies beyond the last regular bin and before
c collapsing them into the ntestbin+1 location.
c Thus chempot = (chempotreg+chempothigh)/totsample where
c   totsample = inssample + tesbin(3,ntestbin+1)
c Eliminated calculation of chempot2, etc
c
c mcporeV12: Feb 5, 2021 : printing histo values for ntestbin+1 rather than just ntestbin
at end of run.
c Define nlowbin as int*16 to count low energy removals instead of printing each one,
then print at end of run.
c Previously (V11a) printed unnormalized chempot, chempothigh, and chempotreg.
c
c mcporeV12b: April 7, 2021: low initial value of energypn1min=0 means that when
c insertions are higher than zero, the pos variable will stay at initialized or
c default values of 0 and so next xyz file will have many particles at origin.
c will try setting energynp1min to a high value near largest double precision real
c like 1.0e300; may need to do check on these overlaps before printing meth file


c
c Input needed includes temperature, T and associated bulk methane chemical
```

```
c potential, mu; radius of pore.
c Multiple runs are needed with different methane number densities
c Post-analysis of the histogram data will yield distribution of filling of
c a heterogeneous set of pores using the grand-canonical frameworks of Rasaiah
c et al, YYY
c
      real*8 a,b,c,x, posold(3),rnew,rhopore,mubye,mass
      real*8 lambda, volpore, allsample, muidbye,betamu
      real*8 energynp1min,equil,temp,beta,chempot,chempotreg
     real*8 chempothigh
      integer insertfactor
     integer*16 inssample, remsample, totsample, nlowbin
      real*8, allocatable :: pos(:,:),r(:),testbin(:,:)
      real*8, allocatable :: energypass(:),energybin(:,:)
      real*8, allocatable :: posnp1min(:,:)
      real*8 hgx(101),hg15y(101),hg15,hg15y2(101)
      real*8 hg45y(101),hg45,hg45y2(101)
      character*30 input, output,restart,startnp1,histogram,snap
      character*80 comment
      character*2, allocatable :: id(:)
      integer nmol,i,j,k,n,iter,ipass,imove,iranstate
      integer npass,nequil,calib,ntest, nsnap,ibin,ioffset
      integer movaccept, movattempt,deltest,dioffset
      real*8 epsmeth,sigmeth,epspore,sigpore,porelength,radpore
      real*8 pi,rgas,estar,kbstar,vtot,cutoff2,energytot,vp
      real*8 vpore,vmol,hyperg, ran,random(3),energytotold
      real*8 fluidenergy,fluidenergytot,poreenergy,poreenergytot
      real*8 vmold,vmnew,vpold,vpnew,energynew,energyold,energymol
      real*8 flen,flentot,poren,porentot,entot,enmolalt
      real*8 xtest,ytest,rtest,numdens,density, ans
```

```
C
c spline data for hyperg15 2F1(-1.5,-1.5,1,x) and hyperg45 2F1(-4.5,-4.5,1,x)
c at 101 points 0,1,step 0.01
c       nhg=101
c       do i=1,nhg
c       hgx(i)=(i-1)*0.01d0
c       hg15y(i)= hyperg(-1.5d0,-1.5d0,1.0d0,hgx(i),iter,1.0d-10)
c       hg45y(i)= hyperg(-4.5d0,-4.5d0,1.0d0,hgx(i),iter,1.0d-10)
c       print*,hgx(i),hg15y(i),iter
c       enddo

c initialize hyperg15 and hyperg45
c use natural cubic spline
c       call spline(hgx,hg15y,nhg,1.0d30,1.0d30,hg15y2) ! natural y"=0 at ends
c       call spline(hgx,hg15y,nhg,0.094,.05157,hg15y2) ! specified y'
c       do i=1,nhg
c       print*,hgx(i),hg15y2(i)
c       enddo
c check accuracy between fitted points
c       do i=1,201
c       x=rand()
c       x=0.005*(i-1)
c       call splint(hgx,hg15y,hg15y2,nhg,x,hg15)
c       exact=hyperg(-1.5d0,-1.5d0,1.0d0,x,iter,1.0d-10)
c       print*,x,hg15,exact,exact-hg15
c       enddo
c
c       call spline(hgx,hg45y,nhg,1.0d30,1.0d30,hg45y2)
c       do i=1,nhg
c       print*,hgx(i),hg45y2(i)
```

```
c       enddo
c       do i=1,201
c       x=rand()
c       x=0.005*(i-1)
c       call splint(hgx,hg45y,hg45y2,nhg,x,hg45)
c       exact=hyperg(-4.5d0,-4.5d0,1.0d0,x,iter,1.0d-10)
c       print*,x,hg45,exact,exact-hg45
c       enddo
c
c spline seems to be about 3 times faster than raw hyperg
c units for calculation
c some units and dimensions
c
c we will read and write
c     distances in Angstroms
c     energies in K
c     mass in g/mol or amu
c
c to handle periodic boundary conditions in the axial or z direction of pore
c after a move apply  RZ(I)=RZ(I)-porelength*anint(Rz(i)/porelength)
c after calculating pair separation vector apply RZIJ=RZIJ-
porelength*anint(rzij/porelength)
c
c some parameters like masses, potential energy parameters
c
c


c       mass=16.042             !mass of methane in amu
c       epsmeth=147.9                   !epsilon for methane (Goodbody FT1991,82,1951)
c       sigmeth=3.73            !req for methane (Goodbody FT1991,82,1951)
```

```
c       epspore=133.3          !really rho_s * epspore
c       sigpore=3.21           !need better values
c       rhopore=0.153          !density of O atoms per angstrom^2
        pi=acos(-1.0)
        ans=42
c
c some lines of code for testing
c test pore potential
c       print*,'test methane pore potential'
c       do i=1,101
c       radpore=10.
c       rtest=(i-1)*0.1
c       poreenergy=vpore(rtest,radpore,epspore,sigpore,rhopore)
c       print*,rtest,radpore-rtest,poreenergy
c       enddo
c
c mcpore.input will be the file that contains all the control parameters and will be directed
c in to the program using  <  or redirect character (unit 5)
c
c <datafile.xyz>        / name of file containing starting config unit 10
c <restart.xyz>                 / name of file for restart config (n methaanes) unit 20
c <startnp1.xyz>        / name of file for start config (n+1 methanes) unit 25
c <energy.out>                  / name of final total energy data (unit 30)
c <histo.out.           / name of removal and insertion energy histogram file (35)
c <snap.xyz>            / name of movie file (40)
c ranstate              / ranstate = 0,1= default seed ,time-based seed
c radpore               / radius of confining pore in Angstrom
c porelength            / length of pore in sigmeth units
c temperature           / temperature in K
c cutoff                / potential cutoff in sigmeth units
```

```
c npass                 / number of MC passes (will be multiplied by 1.1 and first 10% not
stored
c insertfactor        / ? this line added 9/16/20
c ncalib                / number of passes between total MC energy recalculations
c deltest               / number of accepted moves between binned moves
c nsnap                         / number of xyz files to print out in snap
c iconfig               / 0, 1, 2    thermal, quenched, both
c stepmax               / stepmax for MC moves in Angstroms
c
        read*,input
        read*,restart
        read*,startnp1
        read*,output
        read*,histogram
        read*,snap
c
        read*,mass
        read*,epsmeth
        read*,sigmeth
        read*,epspore
        read*,sigpore
        read*,rhopore
c
        read*,iranstate
        read*,radpore   ! in Angstroms
c removed a commented line here that multiplied radpore by sigmeth (2/26/13)
        read*,rporecut ! in Angstrom
        read*,porelength ! in sigmeth units so next line converts to Angstroms
        porelength=porelength*sigmeth        !
c simulation assumes z coordinates of central box are from -.5 to +.5 times porelength
```

```
      read*,temp
      beta=1./temp
      read*,cutoff  ! in sigmeth so next line converts to Angstroms
      cutoff=cutoff*sigmeth
      cutoff2=cutoff**2
      read*,npass
      read*,equil
      nequil=npass*equil
      read*,insertfactor
      read*,ncalib
      read*,deltest
      read*,nsnap
      read*,stepmax
    if (cutoff.gt. porelength*0.5) then
    print *, 'Porelength and cutoff values inconsistent: ', porelength, cutoff
    stop
    endif
c
c some lines of code for testing
c test pore potential
      open (unit=50,file='vporetest.out')
      write(50,*)'#test molecule pore potential'
      write(50,*) '#', radpore,epsmeth,sigmeth,epspore,sigpore,rhopore
      do i=1,101
      rtest=(i-1)*(radpore/100)
      poreenergy=vpore(rtest,radpore,epspore,sigpore,rhopore)
      write(50,*) rtest,radpore-rtest,poreenergy,poreenergy/epsmeth
      enddo
      close(50)
c
```

```
      open (unit=10,file=input)
      open (unit=20,file=restart)
      open (unit=25,file=startnp1)
      open (unit=30,file=output)
      open (unit=35,file=histogram)
      open (unit=40,file=snap)

c initialize random number generator
      call randinit(iranstate)
c read in initial configuration
      read(10,*) nmol
      read(10,*) comment
      allocate(pos(3,nmol+1))
      allocate(r(nmol+1))
      allocate(energypass(nmol+1))
      allocate(id(nmol+1))
c
      do i=1,nmol
      read(10,*) id(i),(pos(k,i),k=1,3)
      r(i)=sqrt(pos(1,i)**2+pos(2,i)**2)
      if (r(i).ge.radpore-1.5) then
      print*,'Warning: atom ',i,' is too close to or beyond pore
     .          radius: STOPPING'
      stop
      endif
      enddo
c calculate potential energy of each atom and of the whole configuration
      fluidenergytot=0.0d0
      poreenergytot=0.0d0
      do i=1,nmol
```

```
c       BGF: Sum over all gas phase molecule interactions of molecule i with j for i≠j:
        fluidenergy=vmol(pos,nmol,i,porelength,cutoff2,sigmeth,epsmeth)
        fluidenergytot=fluidenergytot+fluidenergy
c       BGF: calculate molecule-wall potential energy of molecule i:
        poreenergy= vpore(r(i),radpore,epspore,sigpore,rhopore)
        poreenergytot=poreenergytot+poreenergy
        energymol=fluidenergy+poreenergy   !"energy" of molecule i
c       print*,i,fluidenergy,poreenergy,energymol
        enddo
        fluidenergytot=0.5*fluidenergytot
c       BGF: division by 2 to account for double counting molecule-molecule
interactions.
        energytot=fluidenergytot+poreenergytot
        print*, fluidenergytot,poreenergytot,energytot
c print initial config for movie
        write(40,*) nmol
        write(40,*) 'Initial energy= ', energytot
        do i=1,nmol
        write(40,*) id(i),pos(1,i),pos(2,i),pos(3,i)+porelength/2 !shift to (0,porelength) for
vis
        enddo
        i=0
        write(30,*) i,energytot
c setup the array to bin total energies
c
        ebinwidth=1  ! units are K
        vp=vpore(radpore-sigpore,radpore,epspore,sigpore,rhopore)
c       print*,nmol,radpore, radpore-sigpore,vp
        esmall = -nmol*3*epsmeth +nmol*vp  ! lowest total energy for n molecules
c
```

```
c       BGF: !!Question!! Does the above mean that esmall depends on the number of
molecules?
c      FGA (2/5/21): esmall is used for total energy in pore and depends on number of
molecules (desmall does not)
c
c       print*, esmall,floor(esmall)
c       ebig = esmall*0.85        !problem for n=1
        ebig =0.0
        nbin=(floor(ebig)-floor(esmall))/ebinwidth
c       print*,esmall, ebig
c       print*, floor(esmall),floor(ebig),'nbin= ', nbin
        allocate (energybin(2,nbin))
        do i=nbin,1,-1
        energybin(1,i)= int(esmall+i)-2
        energybin(2,i)=0.0
        enddo
c       print*,energybin(1,1),energybin(1,nbin)
        vp=esmall*.92
        ibin =int(vp)-int(esmall)+1
        ioffset=int(esmall)-1
c       print*,vp, esmall, vp-esmall, ibin,energybin(1,ibin)
c
c setup the arrays to bin insertion and removal energies
c added May 9. 2012 modified June 2, 2012
c Feb 4, 2021: note: ntestbin values do not depend on number of mols
        vp=vpore(radpore-sigpore,radpore,epspore,sigpore,rhopore)
        radmax=radpore-rporecut
        radmax2=radmax**2
        desmall = 2*vp-12*epsmeth
        debig =-desmall
```

```
          dioffset=floor(desmall)/ebinwidth
          ntestbin=(floor(debig)-floor(desmall))/ebinwidth
          print*, vp,desmall, debig, dioffset
          print*, floor(desmall),floor(debig),'ntestbin= ', ntestbin
          allocate(testbin(3,ntestbin+1))
          do i=1,ntestbin+1
          testbin(1,i)=floor(desmall)/ebinwidth+(i-1)*ebinwidth
          testbin(2,i)=0.0
          testbin(3,i)=0.0
          enddo
          print*,ntestbin,testbin(1,1),testbin(1,ntestbin),
     .    testbin(1,ntestbin+1)
c
       chempothigh=0.0
       nlowbin=0              !Added 2/4/21
c switch for calculating insertion energy of empty cavity
          if(nmol.eq.0) then
c insertion histogram
c         BGF: Choose points for insertion of a molecule within a box of length porelength
c         and x,y in ±r, then see if the x,y coordinates are within a radius (r - rcutoff)
c         Sampling is therefore uniform in Cartesian space.
c
       energynp1min=1.0d300
          allocate(posnp1min(3,nmol+1))
                   do j=1,10000000     !(how many times should we try this?)
                   call random_number(random)
                   pos(3,nmol+1)= (random(3)-0.5)*porelength   ! already in A
c these lines give uniform disk and therefore volume sampling
                   xtest=2*(radmax)*(random(1)-0.5)
                   ytest=2*(radmax)*(random(2)-0.5)
```

```
c                 print*,xtest,ytest
                  if((xtest**2+ytest**2).le.radmax2) then  ! no sqrt needed for test
                  rtest=sqrt(xtest**2+ytest**2)
                  pos(1,nmol+1) = xtest
                  pos(2,nmol+1)= ytest
                  etest=vpore(rtest,radpore,epspore,sigpore,rhopore)+
     .   vmol(pos,nmol+1,nmol+1,porelength,cutoff2,sigmeth,epsmeth)
c                 ibin=floor(etest)-dioffset

c test if this insert config is lower than all others so far
c                 print*, ipass, i, energypass(i),etest, etesttotal
c
c         BGF: if the insertion energy is a new minimum, then update array posnp1min
c         which will be output as the starting configuration for the n+1 molecule simulation.
c
                  if(etest.le.energynp1min) then
                  energynp1min=etest
                    do jj=1,nmol+1
                    do k=1,3
                    posnp1min(k,jj)=pos(k,jj)
                    enddo
                    enddo
                   endif
c
                   if(etest.lt.floor(desmall)) then
                   print*,j,ibin,etest,"insertion energy LOW"
          stop   ! added stop on 1-21-21
                   else
                   if(etest.gt.floor(debig)+1) then
```

```
                    ibin=floor(etest)-dioffset
c                   print*,ipass,j,ibin,etest, "insertion energy HIGH"
                    testbin(3,ntestbin+1)=testbin(3,ntestbin+1) +1
                chempothigh= chempothigh+exp(-beta*(etest))
                    else
                    ibin=floor(etest)-dioffset
c                   print*, ipass, j,ibin,etest,"insertion energy normal"
                    testbin(3,ibin)=testbin(3,ibin)+1
                    endif
                   endif
                   endif
                   enddo
c print histogram
          write(35,*) '#',nmol, radpore, porelength, temp
          inssample=0
          chempotreg=0.0
          do i=1,ntestbin ! 1-21-21 chempot now excludes extra overflow bin
c compute chemical potential using standard insertion method
          chempotreg = chempotreg+ testbin(3,i)*exp(-beta*(testbin(1,i)
     .                                          +0.5*ebinwidth))
        write(35,*) testbin(1,i),0,testbin(3,i)
          inssample=inssample+int(testbin(3,i))  !  does int need version for precision
          enddo
c compute mu based on sampling performed with no annulus correction
c error checking
       if (inssample < 0) print *, 'Inssample is negative! Check whether precision of
inssample
     .   has been exceeded'


       chempot = chempotreg +chempothigh  !unnormalized
```

```
      totsample = (inssample+testbin(3,ntestbin+1))
      chempot = chempot/totsample
      chempothigh=chempothigh/testbin(3,ntestbin+1)
      chempotreg=chempotreg/inssample
         betamu= -log(chempot)
         mubye=betamu/beta/epsmeth
      print*, 'N=',nmol,'Temp=',temp,'beta=',beta
         print*,'inssample= ',inssample
      print*,'chempotreg= ', chempotreg
      print*,'testbin(3,ntestbin+1) = ', testbin(3,ntestbin+1)
      print*,'chempothigh= ', chemphothigh
      print*,'totsample = ', totsample
      print*,'chempot= ',chempot
         print*,'betamu1= ',betamu
         print*,'mubye= ',mubye
c
         write(25,*) nmol+1 ! start config for n+1 methanes
         write(25,*) radpore, temp, energynp1min
         do i=1,nmol+1
         write(25,*) 'Me',(posnp1min(k,i),k=1,3)
         enddo
c now stop calculation
         stop
         endif  ! ends nmol = 0 switch
c set up initial energy of nmol+1 methanes
         energynp1min=1.0d300
         allocate(posnp1min(3,nmol+1))

c
c
```

```
c begin MC
c
        movaccept=0.0
        movattempt=0.0
        ntest=deltest
        print*, 'npass=',npass,'nequil=',nequil,'deltest=',deltest
c
c       BGF: Beginning of outer loop over #of passes (+ equilibration).
c       ipass counts from 1 to a million + 50k passes for npass = 10^6 and equil = .05
c       In each "pass", attempt to move each molecule once.
c       Every "deltest" accepted moves, sample the insertion and removal histograms.
c       If deltest = nmol, then the sampling is done once per pass.
c       If deltest < nmol, then the sampling is done more times than there are passes.
c
        do ipass=1,npass+nequil
        if(ipass.eq.nequil+1) next=movaccept+deltest
                do i=1,nmol  ! moving molecule i
c compute vmold before move
                vmold=vmol(pos,nmol,i,porelength,cutoff2,sigmeth,
     .          epsmeth)
                vpold=vpore(r(i),radpore,epspore,sigpore,rhopore)
                energyold=vmold+vpold
                        call random_number(random)
                        do k=1,3
                        posold(k)=pos(k,i)
                        pos(k,i)= pos(k,i)+2*stepmax*(random(k)-0.5)
                        enddo
c               print*, 'pass= ',ipass,'i= ',i,'stepmax= ',stepmax
c               print*, (posold(k),k=1,3)
c               print*, (random(k),k=1,3)
```

```
c                 print*, (pos(k,i),k=1,3)
                  rold=r(i)
                  r(i)=sqrt(pos(1,i)**2+pos(2,i)**2)
c                 print*, rnew, radpore
                  if(r(i).lt.radpore-1.5d0) then
c
c       BGF: if move goes beyond radial cutoff, don't bother to test move.
c
c after a move apply  RZ(I)=RZ(I)-porelength*anint(Rz(i)/porelength)
                  pos(3,i)=pos(3,i)-porelength*anint(pos(3,i)/porelength)
c get new energy of atom i
        vmnew=vmol(pos,nmol,i,porelength,cutoff2,sigmeth,epsmeth)
                  vpnew=vpore(r(i),radpore,epspore,sigpore,rhopore)
                  energynew=vmnew+vpnew
c                 print*, energynew, energyold, 'deltaE= ',energynew-
c    .                              energyold
                  dele=energynew-energyold
c       print*,'mc:',ipass,i,vmnew,vpnew,energynew,energyold,dele
c call mccompare to accept or reject move
                  call mccompare(energynew,energyold,temp,iacc)
c                 print*,'iacc= ',iacc
c
c       BGF: if move is accepted, then change configuration, increment both movattempt
and
c       movaccept.  Otherwise, retain positions, but increment movattempt.
c
                        if(iacc.eq.1) then
c move accepted, update energytot, energymol(i), keep new positions, update r(i)
                  energytot=energytot-energyold+energynew
c                 print*,ipass,i,iacc,energynew,energyold,energytot
```

```
                movattempt=movattempt+1
                movaccept=movaccept+1
                energypass(i)=energytot

                    else
c move rejected, restore old positions, leave energy unchanged
c               print*,ipass,i,iacc,energynew,energyold,energytot
                do k=1,3
                pos(k,i)=posold(k)
                enddo
                r(i)=rold
                movattempt=movattempt+1
                energypass(i)=energytot
                    endif
                else
c move rejected because rnew too large: restore old positions
                iacc=2
c               print*,ipass,i,iacc,energynew,energyold,energytot
                do k=1,3
                pos(k,i)=posold(k)
                enddo
                r(i)=rold
                movattempt=movattempt+1
                energypass(i)=energytot
                endif
c
c       BGF: this is the end of the IF to see if the move was less than radial cutoff.
c
c               ibin=int(energytot)-ioffset
c               if(ibin.lt.1.or.ibin.gt.nbin) then
```

```
c                 print*,ipass,i,"energy out of range,equilibration issue"
c                 else
c                 energybin(2,ibin)=energybin(2,ibin)+1
c                 print*, ipass,i,ibin,energytot,energybin(1,ibin),
c    .  energybin(2,ibin)
c                 endif


c ntest loop to check do test insertion and removal
c                 print*,'before test',movaccept, next
c                 if(ipass.ge.nequil) then
c
c         BGF: The calculation of insertion and removal histograms is done based on the
number
c         of accepted moves, not attempted moves.  It is done every "deltest" accepted
moves.
c
                  if(movaccept.eq.next) then
c                 print*,'after test',movaccept,next
                  last=movaccept
                  next=last+deltest
c removal histogram
c                 print*, "about to check removals"
                   do j=1,nmol
                  etest=vpore(r(j),radpore,epspore,sigpore,rhopore)+
    .   vmol(pos,nmol,j,porelength,cutoff2,sigmeth,epsmeth)
                  ibin=floor(etest)-dioffset
                    if(ibin.lt.1)then
c                 print*,ipass,j,'removal test energy less than',testbin(1,1)
             nlowbin=nlowbin+1          !binning this test situation rather than printing,
2/4/21
                    elseif (ibin.gt.ntestbin) then
```

```
                testbin(2,ntestbin+1)=testbin(2,ntestbin+1)+1
                else
                testbin(2,ibin)=testbin(2,ibin)+1
                 endif
                 enddo   ! end of j loop over nmol
c insertion histogram
          chempothigh=0.0
                do j=1,nmol*insertfactor      !(how many times should we try this?)
                call random_number(random)
                pos(3,nmol+1)= (random(3)-0.5)*porelength   ! already in A
c               print*,ipass,j, pos(3,nmol+1)
c these four lines gave volume sampling biased towards center
c               rtest=random(1)*(radpore-0.5*sigpore)
c               phitest=random(2)*2*pi
c               pos(1,nmol+1)=rtest*cos(phi)
c               pos(2,nmol+1)=rtest*sin(phi)
c these lines give uniform disk and therefore volume sampling
                xtest=2*(radmax)*(random(1)-0.5)
                ytest=2*(radmax)*(random(2)-0.5)
c               print*,xtest,ytest
                if((xtest**2+ytest**2).le.radmax2) then  ! no sqrt needed for test
                rtest=sqrt(xtest**2+ytest**2)
                pos(1,nmol+1) = xtest
                pos(2,nmol+1)= ytest
                etest=vpore(rtest,radpore,epspore,sigpore,rhopore)+
    .   vmol(pos,nmol+1,nmol+1,porelength,cutoff2,sigmeth,epsmeth)
c               ibin=floor(etest)-dioffset

c test if this insert config is lower than all others so far
                etesttotal=energypass(i)+etest
```

```
c                    print*, ipass, i, energypass(i),etest, etesttotal
                     if(etesttotal.le.energynp1min) then
                     energynp1min=etesttotal
                       do jj=1,nmol+1
                       do k=1,3
                       posnp1min(k,jj)=pos(k,jj)
                       enddo
                       enddo
                     endif
c
                     if(etest.lt.floor(desmall)) then
                     print*,ipass,j,ibin,etest,"insertion energy LOW"
c  Printing problematic configuration into log file
               print*,nmol+1
               print*,
                 do jj=1,nmol+1
                       print*,'Me',(pos(k,jj),k=1,3)
                       enddo
c  Writing lowest energy config to meth file to continue calculations
               write(25,*) nmol+1 ! start config for n+1 methanes
                  write(25,*) radpore, temp, energynp1min
                  do jj=1,nmol+1
                  write(25,*) 'Me',(posnp1min(k,jj),k=1,3)
                  enddo
c  Stopping this run because of low energy error, but running next nmol
               stop  ! added 1-21-21
                       else
                       if(etest.gt.floor(debig)+1) then
                       ibin=floor(etest)-dioffset
c                      print*,ipass,j,ibin,etest, "insertion energy HIGH"
```

```
                    testbin(3,ntestbin+1)=testbin(3,ntestbin+1) +1
               chempothigh= chempothigh+exp(-beta*(etest))
                    else
                    ibin=floor(etest)-dioffset
c                   print*, ipass, j,ibin,etest,"insertion energy normal"
                    testbin(3,ibin)=testbin(3,ibin)+1
                    endif
                   endif
                   endif
                   enddo
c
                   endif  !end of test insertion/removal loop
c                  print*,'new ntest=', ntest
 15                enddo  ! end i loop to complete one pass
          accratio=movaccept/movattempt
c
c write energy at end of each pass to a file
          write(30,*) ipass, energytot
c dump energies accumulated in this pass into bins
c                  do i=1,nmol
c                  ibin = function of energypass(i)
c                  endo
c dump snapshots into movie file (end of each nsnap passes)
c note we're adding porelength/2 to the z direction to center
c the box on porelength/2 for display purposes (VMD's nanotubes are built
c at that position.
          if(mod(ipass,nsnap).eq.0) then
          write(40,*) nmol
          write(40,*) ipass, energytot
          do i=1,nmol
```

```
c         write(40,*) id(i),pos(1,i),pos(2,i),pos(3,i)+porelength/2
          write(40,*) id(i),pos(1,i),pos(2,i),pos(3,i)
          enddo
          endif
c
c within calib loop recalculate energytot from scratch to recalibrate
c energytot and reduce roundoff error
          if(mod(ipass,ncalib).eq.0) then
c
c calculate potential energy of each atom and of the whole configuration
          flentot=0.0d0
          porentot=0.0d0
          do i=1,nmol
          flen=vmol(pos,nmol,i,porelength,cutoff2,sigmeth,epsmeth)
          flentot=flentot+flen
          poren= vpore(r(i),radpore,epspore,sigpore,rhopore)
          porentot=porentot+poren
          enmolalt=flen+poren
c         print*,'alt energy loop: ',i,flen,poren,enmolalt
          enddo
          flentot=0.5*flentot
          entot=flentot+porentot
c         print*,'alt energy results:',ipass,entot,energytot,
c    .                        entot-energytot
          energytot=entot
c
          endif ! end of ncalib  recalibrate if loop
c
          enddo  !  end ipass loop to complete simulation
c
```

```
c print out results
c
          write(35,*) '#',nmol, radpore, porelength, temp
          remsample=0
          inssample=0
          chempotreg=0.0
          do i=1,ntestbin
c compute chemical potential using standard insertion method
          chempotreg = chempotreg+ testbin(3,i)*exp(-beta*(testbin(1,i)
     .                                        +0.5*ebinwidth))
        write(35,*) (testbin(k,i),k=1,3)
          remsample=remsample+int(testbin(2,i))
          inssample=inssample+int(testbin(3,i))
          enddo
c add last bin to histogram file (added 2/5/21)
        write(35,*) (testbin(k,ntestbin+1),k=1,3)
c compute mu based on sampling performed with no annulus correction
          if (inssample < 0) then
             print *, 'inssample is negative, check on precision', inssample
                    endif
       chempot = chempotreg +chempothigh  !unnormalized
       print*, 'unnormalized sum of chempotreg+chempothigh ', chempot
       print*, 'unnormalized sum of chempotreg ', chempotreg
       print*, 'unnormalized sum of chempothigh ', chempothigh
       totsample = (inssample+testbin(3,ntestbin+1))
       chempot = chempot/totsample
       chempothigh=chempothigh/testbin(3,ntestbin+1)
       chempotreg=chempotreg/inssample
       betamu= -log(chempot)
       mubye=betamu/beta/epsmeth
```

```
            volpore=pi*radpore**2*porelength  !Angstroms cubed
            numdens=nmol/volpore
            density=numdens*16.02/6.022E23*1.0E24
      print*,'Temp=',temp,'beta=',beta
      print*,'inssample= ',inssample
      print*,'chempotreg= ',chempotreg
      print*,'testbin(3,ntestbin+1) = ', testbin(3,ntestbin+1)
      print*,'chempothigh= ', chemphothigh
      print*,'totsample = ', totsample
      print*,'chempot= ',chempot
      print*,'N= ',nmol,' betamu1= ',betamu,' volpore= ',volpore
      print*,' numdens= ',numdens,' density= ',density !numdens is N/V, density is gm/cc
      print*,'mubye= ',mubye
      print*,'nlowbin= ',nlowbin
c
c commented out all the chempot2 scaling on 1-21-21
c add missing samples from outer annulus
c        allsample=inssample*(radpore**2)/radmax2
c        print*,'radpore=',radpore,radmax,radpore**2/radmax2,allsample
c        chempot2=chempot/(allsample)
c        betamu2=-log(chempot2)
c        mubye2=betamu2/beta/epsmeth
c
         lambda=6.626e-34/sqrt(2*pi*mass/6.022e26*1.38e-23*temp)*1.e10
c        volpore=pi*radpore**2*porelength  !Angstroms cubed
         muidbye=-temp*log(volpore/lambda**3/(nmol+1))/epsmeth
c        print*, 'remsample=',remsample, "insertsample=",inssample
c        print*, 'allsample=',allsample

c        print*, 'exp(-bmu)=',chempot
```

```
c       print*, 'betamu1=',betamu1
c'betamu2=',betamu2 ! commmented out 1-21-21
        print*, 'mubye1 = ',mubye
c'mubye2=',mubye2 ! commmented out 1-21-21
        print*,'lambda= ',lambda,'muidbye=',muidbye
        write(20,*)nmol    ! restart config
        write(20,*) radpore,temp,npass
        do i=1,nmol
        write(20,*) 'Me',(pos(k,i),k=1,3)
        enddo
        write(25,*) nmol+1 ! start config for n+1 methanes
        write(25,*) radpore, temp, energynp1min
        do i=1,nmol+1
        write(25,*) 'Me',(posnp1min(k,i),k=1,3)
        enddo
c
        print*,ipass-1,energytot,movattempt,movaccept
        end
c     END OF MAIN PROGRAM
c
        function vmol(pos,nmol,j,porelength,cutoff2,
    .             sigmeth,epsmeth)
        real*8::  pos(3,nmol+1),xij,yij,zij,porelength,cutoff2
        real*8:: sigmeth, epsmeth,vmol,vpore,hyperg,sigmeth2
        integer::i,j,nmol
c
c       BGF: This subroutine sums the intermolecular (gas phase) interaction potential
c       over all molecules i≠j, after applying periodic boundary conditions to determine
c       if a molecule at the other end of the pore is within the cutoff distance.
c
```

```
      vmol=0.0d0
      sigmeth2=sigmeth**2
      do i=1,nmol
      if (i.ne.j)then
            xij=pos(1,j)-pos(1,i)
            yij=pos(2,j)-pos(2,i)
            zij=pos(3,j)-pos(3,i)
c after calculating pair separation vector apply ZIJ=ZIJ-porelength*anint(zij/porelength)
            zij=zij-porelength*anint(zij/porelength)
            r2ij=xij**2+yij**2+zij**2
            if(r2ij.lt.cutoff2) then
            vmol=vmol+((sigmeth2/r2ij)**6-(sigmeth2/r2ij)**3)
            endif
      endif
      enddo
      vmol=vmol*4*epsmeth
c
      return
      end
c
      function vpore(r,radpore,epspore,sigpore,rhopore)
      real*8 :: r,radpore,epspore,sigpore,rhopore
      real*8 :: tol,factor,x,vpore,arg
c
c     BGF: calculate the molecule-wall potential for one molecule at radius r.
c
      tol=1.0d-10
      pi=dacos(-1.d0)
      pisq=pi**2
      x=radpore-r
```

```
      ratio=x/radpore
      arg=(1.0d0-ratio)**2
      factor=1./(x*(2.d0-ratio)/sigpore)**2
      vpore=pisq*rhopore*epspore*sigpore**2*(
     .  (63./32.)*factor**5 * hyperg(-4.5d0,-4.5d0,1.0d0,arg,iter,tol)
     .       -3*factor**2*hyperg(-1.5d0,-1.5d0,1.0d0,arg,iter,tol) )
      return
      end
c
      subroutine mccompare(energynew,energyold,temp,iacc)
      real*8 energynew, energyold
      real*8 temp
      delv=energynew-energyold
      if(delv.le.0.0) then
c move accepted--downhill
      iacc=1
c     print*,'delv= ',delv,' downhill so iacc= ',iacc
      else
      w=exp(-delv/temp)
      call random_number(ranno)
      if(w.ge.ranno) then
c move accepted--beats the odds
      iacc=1
c     print*,'delv= ',delv,'w= ',w,'ranno=',ranno,'iacc= ',iacc
      else
c move rejected
      iacc=0
c     print*,'delv= ',delv,'w= ',w,'ranno=',ranno,'iacc= ',iacc
      endif
      endif
```

```
      return
      end


      SUBROUTINE SPLINE(X,Y,N,YP1,YPN,Y2)
      PARAMETER (NMAX=150)
      real*8 X(N),Y(N),Y2(N),U(NMAX),yp1,ypn,p,sig
      real*8 un,qn
      integer i,n
      IF (YP1.GT..99d30) THEN
      Y2(1)=0.d0
      U(1)=0.d0
      ELSE
      Y2(1)=-0.5d0
      U(1)=(3.d0/(X(2)-X(1)))*((Y(2)-Y(1))/(X(2)-X(1))-YP1)
      ENDIF
      DO 11 I=2,N-1
      SIG=(X(I)-X(I-1))/(X(I+1)-X(I-1))
          P=SIG*Y2(I-1)+2.
          Y2(I)=(SIG-1.)/P
          U(I)=(6.*((Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))
     .     /(X(I)-X(I-1)))/(X(I+1)-X(I-1))-SIG*U(I-1))/P
11    CONTINUE
          IF (YPN.GT..99d30) THEN
          QN=0.d0
          UN=0.d0
          ELSE
          QN=0.5d0
          UN=(3.d0/(X(N)-X(N-1)))*(YPN-(Y(N)-Y(N-1))/(X(N)-X(N-1)))
          ENDIF
```

```
            Y2(N)=(UN-QN*U(N-1))/(QN*Y2(N-1)+1.)
      DO 12 K=N-1,1,-1
          Y2(K)=Y2(K)*Y2(K+1)+U(K)
12    CONTINUE
          RETURN
      END
c
          SUBROUTINE SPLINT(XA,YA,Y2A,N,X,Y)
          Real*8 :: XA(N),YA(N),Y2A(N),x,y,a,b,h
          integer :: klo,khi,k,n
          KLO=1
          KHI=N
1     IF (KHI-KLO.GT.1) THEN
          K=(KHI+KLO)/2
      IF(XA(K).GT.X)THEN
            KHI=K
               ELSE
               KLO=K
               ENDIF
               GOTO 1
                ENDIF
                   H=XA(KHI)-XA(KLO)
                   IF (H.EQ.0.) STOP 'Bad XA input.'
                      A=(XA(KHI)-X)/H
                      B=(X-XA(KLO))/H
          Y=A*YA(KLO)+B*YA(KHI)+
     .    ((A**3-A)*Y2A(KLO)+(B**3-B)*Y2A(KHI))*(H**2)/6.d0
          RETURN
            END
c
```

```
      function hyperg(a,b,c,x,iter,tol)
      real*8 :: x
      real*8 :: alpha(0:150), beta(0:150),gamma(0:150),eta(0:150)
      real*8 :: a,b,c, tol,s1,s2,s3
      integer :: iter, iterm1,itermax
c
      itermax=150
      iter=0
      alpha(iter)=0.0d0
      beta(iter)=1.0d0
      gamma(iter)=1.0d0
      eta(iter)=1.0d0
c     print*,a,b,c,x
c     print*
c     print*,alpha(iter),beta(iter),gamma(iter),eta(iter)
c     print*
      do iter=1,itermax
      iterm1=iter-1
      alpha(iter) =(alpha(iterm1)+beta(iterm1))*iter*(c+iterm1)
      beta(iter)=beta(iterm1)*(a+iterm1)*(b+iterm1)*x
      gamma(iter)=gamma(iterm1)*iter*(c+iterm1)
      eta(iter)=(alpha(iter)+beta(iter))/gamma(iter)
      s1=abs(eta(iter)-eta(iterm1))/abs(eta(iterm1))
c     print*,iter,alpha(iter),beta(iter),gamma(iter),eta(iter),s1,tol
      if(s1.le.tol) goto 20
      enddo
c
20    hyperg=eta(iter)
      return
      end
```

```
c
      subroutine randinit(iranstate)
      real*4 r
      integer ::i,n,clock,inranstate
      integer,dimension(:), allocatable ::seed
      if (iranstate.eq.0) then
      call random_number(r)
      else
      call random_seed(size=n)
      allocate(seed(n))
      call system_clock(count=clock)
      seed=clock+37+(/(i-1,i=1,n)/)
      call random_seed(put=seed)
      deallocate (seed)
      call random_number(r)
      endif
      print*,'iranstate',iranstate, 'initial rand no=',r
      return
      end
c
c some lines of code for testing
c test pore potential
c     print*,'test methane pore potential'
c     do i=1,101
c     radpore=5.
c     rtest=(i-1)*0.05
c     poreenergy=vpore(rtest,radpore,epspore,sigpore,rhopore)
c     print*,rtest,poreenergy,poreenergy/11604.505
c     enddo
c     print*
```

```
c       print*,'test methane-methane potential'
c test fluid potential
c       allocate (pos(3,2))
c       do i=1,2
c       do k=1,3
c       pos(k,i)=0.0
c       enddo
c       enddo
c       do i=1,301
c       pos(3,2) = (i-11)*.05+sigmeth
c       nmol=2
c       porelength=10.*sigmeth
c       cutoff2=(4*sigmeth)**2
c       j=2
c       fluidenergy=vmol(pos,nmol,j,porelength,cutoff2,
c    .              sigmeth,epsmeth)
c       print*,pos(3,2)-pos(3,1),fluidenergy,fluidenergy/11604.505
c       enddo
c       deallocate(pos)
c       stop
c test random number generator
c       call randinit(iranstate)
c       do i=1,10
c       call random_number(ran)
c       print*, ran
c       enddo
c       call random_number(random)
c       print*,(random(k),k=1,3)
c       stop
```

AUTHOR'S BIOGRAPHY

Samuel W. Bonnevie was born in Cumberland, Maine on October 17, 1998. He was raised in Cumberland, where he graduated from Greely High School in 2017. At the University of Maine, Sam attained a double degree in Chemical Engineering and Chemistry. He was the recipient of multiple scholarships and awards, including a Top Scholar award, a National Merit scholarship, the Maine Space Grant Consortium award, the Department of Chemistry Senior award, and the Division of Physical Chemistry 2020-2021 Undergraduate award. Sam was also a member of the Varsity Track and Field team for all four years as a pole vaulter.

Upon graduation, Sam intends on pursuing his dream of moving to Norway and working as a chemical engineer among the fjords.