Communication: Predicting virial coefficients and alchemical transformations by extrapolating Mayer-sampling Monte Carlo simulations

Harold W. Hatch,1,2) Sally Jiao,2) Nathan A. Mahynski,1 Marco A. Blanco,1,3) and Vincent K. Shen1
1) Chemical Informatics Research Group, Chemical Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8380, USA
2) Department of Chemical and Biological Engineering, Princeton University, Princeton, New Jersey 08544, USA
3) Institute for Bioscience and Biotechnology Research, University of Maryland, Rockville, Maryland 20850, USA

(Dated: 22 December 2017)

Virial coefficients are predicted over a large range of both temperatures and model parameter values (i.e., alchemical transformation) from an individual Mayer-sampling Monte Carlo simulation by statistical mechanical extrapolation with minimal increase in computational cost. With this extrapolation method, a Mayer-sampling Monte Carlo simulation of the SPC/E (extended simple point charge) water model quantitatively predicted the second virial coefficient as a continuous function spanning over four orders of magnitude in value and over three orders of magnitude in temperature with less than a 2% deviation. In addition, the same simulation predicted the second virial coefficient if the site charges were scaled by a constant factor, from an increase of 40% down to zero charge. This method is also shown to perform well for the third virial coefficient and the exponential parameter for a Lennard-Jones fluid. https://doi.org/10.1063/1.5016165

Keywords: virial coefficients, Mayer-sampling Monte Carlo, alchemical transformation, temperature

I. INTRODUCTION

Virial coefficients play a critical role in numerous theories and approximations that explicitly connect intermolecular interactions to the macroscopic behavior of gases and dilute macromolecular solutions.1,2) Some uses for virial coefficients include describing and predicting gas-phase molecular clustering, Joule-Thomson inversion curves, phase behavior, critical properties, phase instabilities, supercritical extraction and fitting parameters for approximate equations of state.3) Virial coefficients are used to collapse phase diagrams via extended corresponding states.4,5) The second osmotic virial coefficient also aids in the prediction of protein crystallization and aggregation,6) and is a useful target in the development of simplified or coarse grained models because it provides a basic description of average pairwise intermolecular interactions.7) However, many of these applications require the computation of virial coefficients over a wide range of different thermodynamic states and different molecular models.

Mayer-sampling Monte Carlo (MSMC) is a computationally efficient method for determining virial coefficients based upon free energy perturbation.1) However, when virial coefficients are needed over a large parameter or thermodynamic space, the number of required simulations can be a computational burden. This is the case for applications such as computational prediction of the effects of protein mutations, where the number of second virial coefficient evaluations increases exponentially with the number of target amino acids. A similar issue also holds for engineering applications for gases of industrial interest, where the temperature dependence of the virial coefficients is required in small increments.

An efficient approach for calculating thermodynamic properties over a large range of conditions far from the original simulation conditions relies on statistical mechanical extrapolation.8–13) This approach is based upon a Taylor series expansion where the coefficients can be related to ensemble averages, which can be calculated straightforwardly. Previously, this method has been applied to simulations in the canonical,14) isothermal isobaric,15) grand canonical9,11,13) and Gibbs16) ensembles to predict phase equilibria and free energy landscapes.

In this work, we apply statistical mechanical extrapolation to MSMC, mathematically shown in Section II. We show that virial coefficients can be predicted accurately at conditions that differ significantly from the original simulation conditions. In other words, we show that a single MSMC simulation can be used to calculate virial coefficients at different temperatures (Section III) and model parameter values, as in alchemical transformations (Section IV).

II. METHODS

The MSMC1) formula to compute the $k$th order virial coefficient, $B_k$, via biased sampling is

$$B_k^* = \frac{B_k}{\langle \hat{B}_k \rangle} = \frac{\langle \hat{B}_k \rangle}{\langle \hat{\omega} \rangle},$$

(1)

where the “hat” in $\hat{B}_k$ refers to a reference potential with a known virial coefficient, $\langle M \rangle = \frac{\int d^r \omega (\hat{\pi})}{\int d^r \omega}$, $\pi$ is the chosen

---

1)Electronic mail: harold.hatch@nist.gov
sampling distribution, and \( \int d\mathbf{r} \) is an integration over the pair-wise transversal and orientational coordinates. For the second and third virial coefficients, \( k = 2 \) and \( 3 \), respectively, \( \gamma_2 = f_{12} \) and \( \gamma_3 = f_{12} f_{13} f_{23} \), \( f_{ij} = e^{-\beta U_{ij}} - 1 \), where \( \beta = 1/k_B T \) is the inverse temperature, \( T \), \( k_B \) is the Boltzmann constant and \( U_{ij} \) is the pair-wise inter-molecular potential. Typically \( \pi = |\gamma_k| \) is found to be the most efficient, as used in this work, but the following derivation does not depend upon this choice (see discussion under Equation 4). MSMC is then performed with a trial acceptance probability to transition from the old state to the new state, \( p = \min(1, \pi^{\text{new}}/\pi^{\text{old}}) \). Each state contributes to the ensemble averages of Equation 1 for both the model of interest and the reference model.

In order to predict the virial coefficient at a state point or model parameter value, denoted by \( \eta \), we use an extrapolation method based on a Taylor expansion around a reference point, \( \eta_0 \), where a simulation is conducted and the derivatives are obtained for the series,

\[
B_k(\eta) - B_k(\eta_0) = \sum_{n=1}^{n_{\max}} \frac{1}{n!} \frac{\partial^n B_k}{\partial \eta^n} |_{\eta_0} \ (\eta - \eta_0)^n. \tag{2}
\]

A Padé approximant can be used to correct for the truncation of the Taylor series to finite order, \( n_{\max} \).

In order to evaluate the derivatives of \( B_k^* \), we make use of the following recursion relation, which can be derived using the quotient and the Leibniz integral rule, to evaluate the derivatives of the numerator and denominator in Equation 1,

\[
\frac{\partial}{\partial \eta} \left( \frac{\partial^n \gamma_k}{\partial \eta^n} \right) = \left( \frac{\partial^{n+1} \gamma_k}{\partial \eta^{n+1}} \right) - \left( \frac{\partial^n \gamma_k}{\partial \eta^n} \right) \left( \frac{\partial \pi}{\partial \eta} \right). \tag{3}
\]

Therefore, the first derivative of \( B_k^* \) using Equations 1 and 3 is

\[
\hat{\gamma}_k \left( \frac{\partial B_k^*}{\partial \eta} \right) = \left( \frac{\partial \gamma_k}{\partial \eta} \right) - \left( \frac{\partial \gamma_k}{\partial \eta} \right) \left( \frac{\partial \hat{\gamma}_k}{\partial \eta} \right). \tag{4}
\]

Note that Equation 4 is independent of \( \left( \frac{\partial \gamma_k}{\partial \eta} \right) \) due to cancellation of terms between the reference and non-reference potential, which greatly simplifies the final expression and leads to extrapolation equations independent of the choice of \( \pi \).

The equations may be further simplified by choosing a hard particle reference system. In this case, \( B_2 \) and \( \hat{\gamma}_k \) of the reference system are independent of temperature. Therefore, \( \frac{\partial \hat{\gamma}_k}{\partial \eta} = 0 \) for \( n > 0 \), when a hard particle reference model is chosen. Thus, the contribution of the derivatives of the reference potential serves only to cancel terms involving \( \left( \frac{\partial \gamma_k}{\partial \eta} \right) \) from the non-reference potential. Using Equation 3 and beginning with Equation 4 when \( \frac{\partial^n \gamma_k}{\partial \eta^n} = 0 \),

\[
\frac{\hat{\gamma}_k}{\pi} \frac{\partial B_k^*}{\partial \eta} = \frac{\partial^n \gamma_k}{\partial \eta^n} \frac{1}{\pi}. \tag{5}
\]

for a hard particle reference potential. Thus, in order to predict \( B_k \) over a range of temperatures or model parameter values, \( \eta \), one performs a single simulation at \( \eta_0 \), collects ensemble averages \( \langle \frac{\hat{\gamma}_k}{\pi} \rangle \) and \( \langle \frac{\partial^n \gamma_k}{\partial \eta^n} \frac{1}{\pi} \rangle \) up to a maximum order, \( n_{\max} \), then applies Equations 2 and 5. For reference models which are not hard particles, see supplementary material.

Derivatives of \( \gamma_k \) are given by,

\[
\frac{\partial^n \gamma_k}{\partial \eta^n} f_{12} = \frac{\partial^n f_{12}}{\partial \eta^n} \tag{6}
\]

\[
\frac{\partial \gamma_3}{\partial \eta} f_{12} f_{13} f_{23} + f_{12} \frac{\partial f_{13}}{\partial \eta} f_{23} + f_{12} f_{13} \frac{\partial f_{23}}{\partial \eta}. \tag{7}
\]

See supplementary material for higher order derivatives.

### III. TEMPERATURE EXTRAPOLATION

First we demonstrate extrapolation with respect to inverse temperature, \( \beta = 1/k_B T \) (i.e., \( \eta \to \beta \)),

\[
\frac{\partial^n f_{ij}}{\partial \beta^n} = (-U_{ij})^n e^{-\beta U_{ij}}. \tag{8}
\]

MSMC was performed for Lennard-Jones (LJ) particles and SPC/E water, using a hard sphere reference potential centered on the LJ particle or oxygen site, respectively, with a diameter of \( \sigma \). The LJ potential is given by \( U^{LJ} = 4\epsilon[(\sigma/r)^{12} - (\sigma/r)^6] \). The simulations ran for \( 10^8 \) trials, beginning with an equilibration period of \( n_{eq} \) trials where the maximum displacement or rotation (if appropriate) was tuned by 5% every \( 10^4 \) trials for a target of 25% to 50% acceptance. No virial coefficient averages were computed during the equilibration period. One particle was frozen at the origin, and the other particles were selected randomly for a trial with equal probability of either displacement or rotation (if appropriate). For LJ and SPC/E simulations, \( n_{eq} = 10^6 \) and \( 10^7 \), respectively. Four independent simulations were performed for each condition. All plotted results have error bars that are the standard deviation of these four simulations or extrapolations, but these error bars are often smaller than the symbols. For the reference hard sphere potential, \( B_{HS}^2 = 4V_{HS}, B_{HS}^3 = 10V_{HS}^2, \) and \( V_{HS} = \pi \sigma^3/6 \). In this work, \( [n_{max} - 1]/1 \) Padé approximant was used. See Reference 17 for the definition of the \([L/M]\) notation.

Figure 1 shows extrapolation with respect to temperature for the SPC/E water model. A single MSMC simulation of SPC/E water at \( T = 800K \) is able to quantitatively predict the second virial coefficient spanning over 3 orders of magnitude in both \( B_2 \) and \( T \) (see Table 1). Note that the discontinuity in the light blue curve in the lower right of Figure 1 is due to a singularity in the Padé approximant. Singularities occur in Padé approximants when the denominator goes to zero, and they may be treated with a variety of methods. Figures 2 and 3 show the results of temperature extrapolation for the second and third virial coefficients, respectively.
Predicting Virial Coefficients

FIG. 1: The second virial coefficient of the SPC/E water model with units of cm$^3$/mol obtained via MSMC (black circles) and extrapolations from $T = 800K$ for a Taylor series truncated at $n_{\text{max}}$ and a $[n_{\text{max}} - 1/1]$ Padé approximant.

FIG. 2: The second virial coefficient of LJ particles, $\alpha = 6$ obtained via MSMC (black circles) and extrapolations from $\beta \epsilon = 1$.

FIG. 3: The third virial coefficient of LJ particles, $\alpha = 6$, obtained via MSMC (black circles) and extrapolations from $\beta \epsilon = 1$. The results are shifted by a parameter, $\Delta = 10$, so results can be plotted on a log scale.

FIG. 4: The second virial coefficient of LJ particles at $\beta \epsilon = 1$ obtained via MSMC (black circles) and extrapolations from $\alpha = 6$.

IV. ALCHEMICAL TRANSFORMATION

Extrapolation can also be used to determine the virial coefficients over a range of model parameter values, as is required in alchemical transformation or the determination of effective pair potentials. In this case, $\eta \rightarrow \omega$,

$$\frac{\partial f_{ij}}{\partial \omega} = -\beta \frac{\partial U_{ij}}{\partial \omega} e^{-\beta U_{ij}},$$

and higher order terms are provided in supplementary material. As an example, consider the exponent parameter, $\alpha$, in the LJ potential (i.e., $\omega \rightarrow \alpha$), $\frac{\partial U_{LJ}}{\partial \alpha} = 4 \epsilon \ln(\sigma/r)^n \left[2^n(\sigma/r)^2 - (\sigma/r)^n\right]$. The extrapolation of $B_2$ with respect to $\alpha$ is shown in Figure 4 using the appropriate statistics from the same simulation used for temperature extrapolation.

The equations for extrapolating with respect to potential parameters are simplified when the potential is a linear function of the parameter of interest. For example, to extrapolate the virial coefficients with respect to the value of the point charge of SPC/E water, $U^{SPC/E} = U^{LJ} + \chi U^Q$, where $U^Q = k q_i q_j / r$, $\chi$ is a parameter which scales the charges of the water model, $q_i \rightarrow \sqrt{\chi} q_i$, and $k$ is the Coulomb’s law constant. In this case, $\frac{\partial U^Q}{\partial \chi} = 0$ for $n > 1$ and the derivatives of $f_{ij}$, needed for Equations 9 and those in supplementary material, simplify to

$$\frac{\partial^n f_{ij}}{\partial \chi^n} = (-\beta U^Q_{ij})^n e^{-\beta U_{ij}}.$$

An example of extrapolation with respect to the charge parameter $\chi$ of SPC/E water is shown in Figure 5. In this case, a single MSMC simulation of SPC/E water with the original charges is capable of predicting the second virial coefficient from 40% more charge down to no charge with high precision.
The additional computational cost to compute temperature and the charge parameter coefficient was extrapolated with respect to both temperature. For the SPC/E water model, the second virial coefficient was extrapolated with respect to temperature and the exponential model parameter, \( \chi \). We have shown that the extrapolation method accurately predicts virial coefficients over an unexpectedly large range of parameters based on results from a single simulation. For the LJ model, the second virial coefficient was extrapolated with respect to both temperature and the exponential model parameter, \( \alpha \), and the third virial coefficient was extrapolated with respect to temperature. For the SPC/E water model, the second virial coefficient was extrapolated with respect to both temperature and the charge parameter \( \chi \) using a single simulation. The additional computational cost to compute the terms for extrapolation was negligible compared to that of MSMC by itself. See supplementary material for a more detailed discussion about efficiency. An example code is provided.

Future work includes the extrapolation of MSMC for higher order virial coefficients and extrapolation with the overlap-sampling version of MSMC. Multidimensional extrapolation with respect to potential parameter values may greatly decrease the computational cost of finding the set of model parameters which yield a given value of \( B_2 \). An example application of this method is in using \( B_2 \) as a predictor for protein aggregation, where one needs to identify suitable amino acid mutations that decrease self-association propensity. The extrapolation approach may be used to perform a single MSMC simulation for the wild type protein to predict \( B_2 \) for all possible mutations of the amino acids.

**Supplementary Material.** See supplementary material for discussion about the efficiency of this method, and additional equations to evaluate higher-order derivatives. In addition, an algorithm is also included which attempts to reduce the number of simulations required to determine virial coefficients over an interval with a given error tolerance, and attempts to automate selection of the optimal maximum order.

**Acknowledgments.** S.J. acknowledges support from a NIST Summer Undergraduate Research Fellowship, Grant No. 70NANB17H102.

![Graph](image_url)

**FIG. 5:** The second virial coefficient of the SPC/E water model with units of \( \text{cm}^3/\text{mol} \) at \( T = 800K \) obtained from MSMC (black circles) and extrapolation with respect to site charges, \( q_i \rightarrow \sqrt[n]{q_i} \), from a single simulation at \( \chi = 1 \). The results are shifted by a parameter, \( \Delta = 100 \text{ cm}^3/\text{mol} \), so results can be plotted on a log scale.

**TABLE I:** The second virial coefficients, \( B_2 \) from MSMC compared with \( n^{th} \) order extrapolation with respect to a variable, \( v \), from \( \eta_0 \) to \( \eta \) with Taylor series, \( B_2^{Taylor} \) and the Padé approximant, \( B_2^{Padé} \). The SPC/E results are in units of \( \text{cm}^3/\text{mol} \) and the LJ results are normalized by \( \sigma^3 \). The subscript represents the standard deviation from replicates.

<table>
<thead>
<tr>
<th>model</th>
<th>( v )</th>
<th>( \eta_0 )</th>
<th>( \eta )</th>
<th>( n )</th>
<th>( B_2 )</th>
<th>( B_2^{Taylor} )</th>
<th>( B_2^{Padé} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LJ</td>
<td>( \beta_e )</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>-18.261</td>
<td>-18.292</td>
<td>-18.282</td>
</tr>
<tr>
<td>LJ</td>
<td>( \alpha )</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>-21.753</td>
<td>-19.485</td>
<td>-21.61</td>
</tr>
<tr>
<td>LJ</td>
<td>( \alpha )</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>-2.192</td>
<td>-1.751</td>
<td>-2.184</td>
</tr>
<tr>
<td>SPC/E</td>
<td>( T )</td>
<td>800</td>
<td>250</td>
<td>20</td>
<td>-5.71 \times 10^3</td>
<td>-8.83 \times 10^3</td>
<td>-5.61 \times 10^4</td>
</tr>
<tr>
<td>SPC/E</td>
<td>( T )</td>
<td>800</td>
<td>350</td>
<td>20</td>
<td>-2794</td>
<td>-2790</td>
<td>-2780</td>
</tr>
<tr>
<td>SPC/E</td>
<td>( T )</td>
<td>800</td>
<td>10^4</td>
<td>20</td>
<td>17.053</td>
<td>17.21</td>
<td>17.01</td>
</tr>
<tr>
<td>SPC/E</td>
<td>( \chi )</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td>18.618</td>
<td>18.51</td>
<td>18.51</td>
</tr>
<tr>
<td>SPC/E</td>
<td>( \chi )</td>
<td>1</td>
<td>2</td>
<td>20</td>
<td>-4100_{40}</td>
<td>-4100_{50}</td>
<td>-4100_{50}</td>
</tr>
</tbody>
</table>

**V. DISCUSSION AND CONCLUSIONS**

We have shown that the extrapolation method accurately predicts virial coefficients over an unexpectedly large range of parameters based on results from a single simulation. For the LJ model, the second virial coefficient was extrapolated with respect to both temperature and the exponential model parameter, \( \alpha \), and the third virial coefficient was extrapolated with respect to temperature. For the SPC/E water model, the second virial coefficient was extrapolated with respect to both temperature and the charge parameter \( \chi \) using a single simulation. The additional computational cost to compute the terms for extrapolation was negligible compared to that of MSMC by itself. See supplementary material for a more detailed discussion about efficiency. An example code is available.

Future work includes the extrapolation of MSMC for higher order virial coefficients and extrapolation with the overlap-sampling version of MSMC. Multidimensional extrapolation with respect to potential parameter values may greatly decrease the computational cost of finding the set of model parameters which yield a given value of \( B_2 \). An example application of this method is in using \( B_2 \) as a predictor for protein aggregation, where one needs to identify suitable amino acid mutations that decrease self-association propensity. The extrapolation approach may be used to perform a single MSMC simulation for the wild type protein to predict \( B_2 \) for all possible mutations of the amino acids.

**Supplementary Material.** See supplementary material for discussion about the efficiency of this method, and additional equations to evaluate higher-order derivatives. In addition, an algorithm is also included which attempts to reduce the number of simulations required to determine virial coefficients over an interval with a given error tolerance, and attempts to automate selection of the optimal maximum order.

**Acknowledgments.** S.J. acknowledges support from a NIST Summer Undergraduate Research Fellowship, Grant No. 70NANB17H102.

Supplementary Material

**Supplementary material for “Communication: Predicting virial coefficients and alchemical transformations by extrapolating Mayer-sampling Monte Carlo simulations”**

Harold W. Hatch,¹, b) Sally Jiao,² Nathan A. Mahynski,¹ Marco A. Blanco,¹, ³ and Vincent K. Shen¹

¹) Chemical Informatics Research Group, Chemical Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8380, USA

²) Department of Chemical and Biological Engineering, Princeton University, Princeton, New Jersey 08544, USA

³) Institute for Bioscience and Biotechnology Research, University of Maryland, Rockville, Maryland 20850, USA

(Dated: 22 December 2017)

Keywords: virial coefficients, Mayer sampling, Monte Carlo

---

a) Contribution of the National Institute of Standards and Technology, not subject to U.S. Copyright.

b) Electronic mail: harold.hatch@nist.gov
I. INTRODUCTION

This supplementary material document includes additional details for The Journal of Chemical Physics Communications manuscript titled “Predicting virial coefficients and al-chemical transformations by extrapolating Mayer-sampling Monte Carlo simulations.” An algorithm for minimizing the number of evaluations of an extrapolated quantity within an interval is described in Section II. A discussion of the relative efficiency and limitations of the statistical mechanical extrapolation method for Mayer-sampling Monte Carlo simulations is discussed in Section III. Finally, some additional equations for derivatives that are referenced in the original manuscript may be found in Section IV.

II. ALGORITHM FOR EVALUATING AN EXTRAPOLATED QUANTITY WITHIN AN INTERVAL

The following algorithm seeks to evaluate a function, $f(x)$ within an interval $[a, b]$ with a maximum order of extrapolation, $n_{\text{max}}$, while minimizing the number of function and derivative evaluations required for an error tolerance of $\sigma$. First, evaluate $f(a)$, $f(b)$ and their $n_{\text{max}}$ order derivatives using 3 or more replicas to obtain the average and standard deviation. For each unique pair-wise combination of extrapolation orders, test if the extrapolation of
Supplementary Material

\( f \) from \( a \) is statistically identical to the extrapolation of \( f \) from \( b \) for all values of \( x \) within a discretization of the interval \([a, b]\) within an error tolerance of \( \sigma \). Attempt to find the highest order pair-wise combination of extrapolations where this test passes. If the test passes, then the algorithm is complete, and \( f \) may be obtained as the (weighted) average of the two extrapolations within the interval \([a, b]\). If the test fails, then perform a new evaluation for \( f(c) \) and its \( n_{\text{max}} \) derivatives at \( c = (a + b)/2 \). Recursively apply the test and possible subdivision described above for both subintervals \([a, c]\) and \([c, b]\) until the desired tolerance is reached for all subintervals.

An optional optimization to the algorithm automatically attempts to select a more optimal maximum order of extrapolation, for the case when the initially provided \( n_{\text{max}} \) is set higher than necessary. When a subinterval passes the extrapolation test, record the highest order extrapolation which passed as \( n_{\text{pass}} \). For the next subinterval, evaluate the derivatives only up to an order of \( n_{\text{pass}}^r = n_{\text{pass}} + \lceil n_{\text{max}}/10 \rceil \), for \( n_{\text{max}}^r \epsilon [n_{\text{min}}, n_{\text{max}}] \).

III. A NOTE ON THE EFFICIENCY AND LIMITATIONS OF THIS METHOD

In this section, the efficiency of the extrapolation method is discussed relative to simply performing a series of direct MSMC simulations and interpolating over the interval of interest. The computational efficiency depends on the computational cost of evaluating the model, the parameter range and resolution of interest, and the acceptable error tolerance. The extra computational cost to compute the derivatives in order to perform extrapolations of the second virial with respect to temperature is relatively small and scales with time and memory with the desired order. For model parameter derivatives, and higher order virial coefficients, the derivatives can be more computationally expensive to compute. The efficiency of the extrapolation method is expected to increase for the more expensive computational models, such as a protein model with over 100 interaction sites, when the computational time required to evaluate the virial coefficient far exceeds the computational cost of recording a few derivatives with only \( n_{\text{max}} \) multiplication operations and memory storage. The desired parameter range and resolution is also an important factor in the efficiency. If one only wants to know the virial coefficient at two different parameter values, then one does not need to use the extrapolation method at all. But if one wants to know the virial coefficient
Supplementary Material

accurately over a continuous range of conditions (e.g., for equations of state and integration or differentiation) then the extrapolation method is expected to be more efficient for the same level or accuracy. Another example where the extrapolation method would be more efficient is searching for a particular virial coefficient value (e.g., the finding the Boyle temperature, $T_B$, where $B_2(T_B) = 0$). Without derivative information from extrapolation, more iterations would be required for the same level of precision. Finally, the acceptable error tolerance is also a deciding factor in whether the extrapolation method is more efficient. For example, for obtaining qualitative trends from a high throughput screening of many different conditions (e.g. point mutations on a protein), the extrapolation method could provide this information with fewer simulations. And for more strict error tolerance, the extrapolation method may still be more effective (e.g., extrapolation of SPC/E from 800$K$ to 250$K$ was as accurate as a direct MSMC simulation at 250$K$).

As an example of the efficiency of the method, consider the SPC/E example results, Fig. 1 and Fig. 5, and compare the amount of information obtained from MSMC simulations without extrapolation versus MSMC simulations using the extrapolation method, for an equal amount of computer time required to run. Note that one single simulation was used to extrapolate with respect to temperature and $\chi$. The computational overhead for calculating the derivative terms was negligible compared to the computational cost of evaluating the potential energy and performing the MSMC simulation. Therefore, the computational cost of an MSMC simulation with and without extrapolation was essentially the same for SPC/E. With five MSMC simulations with extrapolation, four simulations may be run at each of the bounds of temperature and charge (e.g., for $T = [250K, 10^5K]$ with $\chi = 1$, and $\chi = [0, 2.5]$ with $T = 800K$). And the fifth simulation is run at the midpoint temperature (i.e., $T = 800K$, $\chi = 1$). For each of the four different extrapolation paths between the middle and the bounds, extrapolate from both the middle or the bound to any point inclusive in this interval, and use statistical analysis to determine if the two independent extrapolations to any of these points are statistically equivalent. Thus, with only five simulations, the entire interval of interest in both temperature and charge is obtained as a continuous function, and with rigorous statistical confidence over this entire interval (not just the end points and middle). The amount of data from five MSMC simulations without extrapolation would provide much less information. If one is only interested in the temperature range, only two or three extrapolation simulations would be required instead of five.
IV. ADDITIONAL DERIVATIVES

This section provides some additional derivative equations used for extrapolations. Section IV A provides some equations for use with non-hard particle reference potentials (i.e., temperature-dependent potentials). Derivatives for the third virial coefficient are provided in Section IV B. Finally, third and fourth order derivatives for extrapolation in potential parameters may be found in Section IV C.

In order to avoid errors in programming the equations manually, one method is to use Mathematica\(^1\) to generate the derivatives and use the “CForm” or “FortranForm” function to convert to text which resembles a programming language. For example, for the third derivative of \(\gamma_3\), one may write

\[
g = f_{12}[a] \ast f_{13}[a] \ast f_{23}[a] \\
CForm[D[g, \{a, 3\}]].
\]

Then one may use a basic regular expression with groups to reformat the derivatives to call from an array (e.g., s/\text{Derivative}(\text{\ldots})(\text{\ldots})(a)/d\{1\}/g).

Note that the following rules may be used to facilitate taking derivatives of arbitrary order. The derivative of a product to arbitrary order is given by the General Leibniz Rule. The derivative of a reciprocal to arbitrary order is given in Reference 2. The chain rule for an arbitrary order derivative is given by the Faà di Bruno formula.\(^3\)

A. Extrapolation without a hard particle reference potential

For non-hard particle reference potentials, Equation 5 in the main text is no longer valid because it is not necessarily true that \(\frac{\partial^n \gamma_k}{\partial \eta^n} = 0\). The number of terms grow quickly as the order, \(n\), increases. Therefore, we introduce the shorthand notation,

\[
g_n = \left\langle \frac{\partial^n \gamma_k}{\partial \eta^n} \right\rangle. \tag{1}
\]

The first, second and third derivatives are

\[
g_2 \frac{\partial B_k^*}{\partial \eta} = g_1 \dot{g} - g \ddot{g}, \tag{2}
\]

\[
g_2 \frac{\partial^2 B_k^*}{\partial \eta^2} = g_2 \ddot{g} - g \dddot{g} - 2 g_1 \dot{g} \dot{g} \frac{\partial B_k^*}{\partial \eta}, \tag{3}
\]
Supplementary Material

\[ \dot{y}^2 \dot{\partial^3 B}_k^* = g_3 \ddot{y} - g \dot{g}_3 + g_2 \dot{g}_1 - g_1 \ddot{g}_2 - 2 \left[ (\dot{g}_1^2 - \ddot{g}_2) \frac{\partial B_k^*}{\partial \eta} + 2 \dot{g}_1 \frac{\partial^2 B_k^*}{\partial \eta^2} \right]. \]  

(4)

B. Derivatives of \( \gamma_3 \)

For compactness, we make use of the derivative shorthand \( \frac{\partial f}{\partial \eta} \rightarrow f' \) and \( \frac{\partial^n f}{\partial \eta^n} \rightarrow f^{(n)} \).

\[ \gamma_3'' = f_{23} (2 f_{12}'' f_{13} + f_{13} f_{12}'' + f_{12} f_{13}'') + 2 (f_{13} f_{12}' + f_{12} f_{13}) f_{23}' + f_{12} f_{13} f_{23}'' \]  

(5)

\[ \gamma_3^{(3)} = 3 f_{23}' (2 f_{12}' f_{13}'' + f_{13}' f_{12}'' + f_{12} f_{13}''') + 3 (f_{13}' f_{12}'' + f_{12} f_{13}''') f_{23}' + f_{23} (3 f_{13}' f_{12}''' + f_{13} f_{12}''' + f_{13}' f_{12}'' + f_{12} f_{13}''') \]  

(6)

\[ \gamma_3^{(4)} = 6 f_{23}'' (2 f_{12}'' f_{13} + f_{13} f_{12}'' + f_{12} f_{13}'') + 4 f_{23} (3 f_{13}' f_{12}'' + f_{13} f_{12}'' + f_{13}' f_{12}'' + f_{12} f_{13}'' + f_{12} f_{13}'') + 4 f_{23} (f_{13}' f_{12}' + f_{13}'' f_{12} + f_{12} f_{13}' + f_{12} f_{13}' + f_{12} f_{13}'') \]  

(7)

\[ \gamma_3^{(5)} = 10 f_{23}''' (3 f_{13}' f_{12}''' + f_{13} f_{12}''' + f_{12} f_{13}''') + 10 f_{23} (3 f_{13}' f_{12}'' + f_{13} f_{12}'' + f_{12} f_{13}'') + 5 f_{23} (4 f_{12}'' f_{13}' + f_{13} f_{12}'' + f_{12} f_{13}'' + f_{12} f_{13}'' + f_{12} f_{13}'') \]  

(8)

C. Extrapolation in potential parameters

Higher order derivatives for extrapolation in potential parameters, where \( U = U(\omega) \) (see Section IV in the main text for model parameter extrapolation), are provided for convenience below.

\[ \frac{\partial^3 f_{ij}}{\partial \omega^3} = \beta e^{-\beta U_{ij}} \left[ -\beta^2 \left( \frac{\partial U_{ij}}{\partial \omega} \right)^3 + 3 \beta \frac{\partial U_{ij}}{\partial \omega} \frac{\partial^2 U_{ij}}{\partial \omega^2} - \frac{\partial^3 U_{ij}}{\partial \omega^3} \right]. \]  

(9)
\[ \frac{\partial^4 f_{ij}}{\partial \omega^4} = \beta e^{-\beta u_{ij}} \left[ \beta^3 \left( \frac{\partial U_{ij}}{\partial \omega} \right)^4 - 6\beta^2 \left( \frac{\partial U_{ij}}{\partial \omega} \right)^2 \frac{\partial^2 U_{ij}}{\partial \omega^2} + 3\beta \left( \frac{\partial^2 U_{ij}}{\partial \omega^2} \right)^2 + 4\beta \frac{\partial U_{ij}}{\partial \omega} \frac{\partial^3 U_{ij}}{\partial \omega^3} - \frac{\partial^4 U_{ij}}{\partial \omega^4} \right]. \]

\text{REFERENCES}

1 Certain commercial firms and trade names are identified in this document in order to specify the installation and usage procedures adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that related products are necessarily the best available for the purpose.
