universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Calculation of solvation free energies using a Drude polarizable force field

verfasst von | submitted by

Marvin Löfler BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 863
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Biologische Chemie
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Stefan Boresch

Acknowledgements

I would like to express my sincerest gratitude to Professor Stefan Boresch for his unwavering support and guidance. He not only made himself consistently available to address my questions and concerns, but also did so with unmatched thoroughness and care. His open-door policy, enthusiasm for teaching and sharing his expertise, and great intellectual honesty are qualities one should not take for granted, and I am truly grateful to have experienced them. Next, I would like to extend my thanks to Johannes Karwounopoulos — co-supervisor in spirit, if not in name — for his invaluable guidance on practical matters. He has been a patient and gracious mentor throughout my work with `transformato` and, more broadly, a great person to be around. The same goes for Anna Picha, Marion Sappl, and Márta Gődény, who, despite having no obligation to help, were always willing to do so. Finally, I would like to thank Professor Christian Schröder and Professor Othmar Steinhauser for the engaging discussions we had and the knowledge they generously shared.

Abstract

Calculating free energy differences is of great utility in various scientific disciplines, including chemistry, biology, and materials science. Free energy simulations (FES) have become an increasingly important computational tool for calculating free energy differences in recent decades. Molecular dynamics (MD)-based FES methods have especially benefitted from recent developments in GPU acceleration.

This thesis validates the 'serial atom insertion' (SAI) approach for the computation of absolute solvation free energies (ASFEs) using a Drude polarizable force field. The correctness and utility of the SAI approach has previously been demonstrated for additive force fields. Earlier findings suggest that explicitly incorporating polarizability improves ASFE calculations for small molecules compared to using additive force fields.

In practice, the calculation of free energy differences requires simulations of multiple intermediate states. The SAI approach generates these states in a way that allows for the use of 'plain' MD simulations — without the need for specialized free energy routines. By itself, avoiding these routines already accelerates the simulations. Additionally, because SAI intermediate states can be used with any MD engine, they can leverage GPU-accelerated platforms. The setup of SAI intermediate states is facilitated by `transformato`, an in-house Python tool originally developed for additive force fields and adapted here for a Drude polarizable force field.

Kurzfassung

Die Berechnung freier Energieunterschiede ist von großer Bedeutung in verschiedenen wissenschaftlichen Disziplinen, einschließlich der Chemie, der Biologie und der Materialwissenschaft. Freie-Energie-Simulationen (FES) haben als computergestütztes Verfahren zur Berechnung freier Energieunterschiede in den letzten Jahrzehnten an Bedeutung gewonnen. Insbesondere haben FES-Methoden, die auf der Molekulardynamik (MD) basieren, von Entwicklungen in der GPU-Beschleunigung profitiert.

Diese Arbeit ist mit der Validierung des ‚serial atom insertion' (SAI)-Ansatzes zur Berechnung absoluter Solvatisierungsenergien (ASFEs), unter Nutzung eines Drude-polarisierbaren Kraftfelds, befasst. Die Korrektheit und die Nützlichkeit der SAI-Methode wurden für additive Kraftfelder bereits gezeigt. Des Weiteren wurden in einer vorangegangenen Studie mit expliziter Behandlung der Polarisierbarkeit bessere Ergebnisse bei der Berechnung von ASFEs erzielt als unter Verwendung eines additiven Kraftfelds.

In der Praxis erfordert die Berechnung freier Energieunterschiede die Simulation mehrerer Zwischenzustände. Generiert man diese nach der SAI-Methode, können sie zum Aufsetzen ‚einfacher' MD-Simulationen verwendet werden — also, ohne dass in den Simulationen selbst spezielle Routinen für die Berechnung freier Energien erforderlich sind. Dies allein beschleunigt die Berechnung freier Energien bereits. Zudem sind die SAI-generierten Zwischenzustände grundsätzlich mit jedem MD-Programm kompatibel, was zusätzlich die Verwendung GPU-beschleunigter Plattformen ermöglicht. Die Erzeugung der SAI-Zwischenzustände erfolgt mit `transformato`, einem hauseigenen Python-Tool, welches ursprünglich für FES mit additiven Kraftfeldern entwickelt, und hier für ein Drude-polarisierbares Kraftfeld angepasst und erweitert wurde.

Contents

Abbreviations

ABNR Adopted Basis Newton-Raphson

ASFE Absolute Solvation Free Energy

BAR Bennett Acceptance Ratio

CC Common Core

FES Free Energy Simulations

FF Force Field

GPU Graphics Processing Unit

LJ Lennard-Jones

MBAR Multistate Bennett Acceptance Ratio

MC Monte Carlo

MD Molecular Dynamics

PSF Protein Structure File

SAI Serial Atom Insertion

SCF Self-Consistent Field

SD Steepest Descent

TI Thermodynamic Integration

1 Introduction

Free energy is a fundamental thermodynamic property, representing the portion of a system's energy available to do work at a constant temperature. It is a quantity of considerable interest because it can enable us to predict how a system evolves. Specifically, the free energy difference between states determines their relative stability and the spontaneity of the associated transition process. Therefore, the accurate calculation of free energy differences and free energy landscapes has become an important tool in the study of a diverse range of systems and processes. Examples in the field of biochemistry include small molecule solvation, ligand-receptor binding affinity, and protein conformational dynamics.[1,2]

Free energy differences can be derived from experimental thermodynamic data. However, these are often not readily available, particularly for novel drug candidates or hazardous chemicals.[3] Furthermore, laboratory experiments have considerable throughput constraints compared to computational methods. With these limitations on scalability, free energy simulations (FES) have garnered significant interest over the past decades, already being employed academically and commercially in areas of drug design, materials science, and chemical and process engineering.[4–6]

The methods employed in FES are varied and subject to considerable fine-tuning. They differ in the force fields used, in the way they model the system under study, in the simulation parameters employed, and in the calculation methods applied to the simulation data.[7,8] For example, FES are either based on Monte Carlo (MC) or molecular dynamics (MD) simulations (the former is not the focus of this thesis and is only mentioned for completeness). MD has become the standard sampling method for FES, benefitting from recent developments such as GPU acceleration.[9] However, both approaches are in use still, and combining them can yield even better results than relying on one or the other.[10]

Incorporating polarizability into free energy simulations has been shown to improve agreement with experimental values and provide new insights.[11–15] Among the various methods available to account for polarizability, the Drude oscillator model[16] stands out, as it combines ease of implementation and computational efficiency.[17]

`transformato`,[18] a Python tool developed in-house, offers rapid calculation of free energy differences based on MD simulations with the so called "common core/serial atom insertion" (CC / SAI) approach. It facilitates an accelerated workflow and cross-platform compatibility, and can currently be used to calculate absolute and relative solvation free energies, as well as relative binding free energies, using an additive force field.[19]

Building on these foundations, this thesis aims to

1. validate the SAI approach for the calculation of absolute solvation free energies (ASFE), while employing a Drude polarizable force field. For this purpose, a test set of small molecules is used, for which well-validated reference and experimental values exist. The systems are set up with the help of `transformato`.

2. examine the agreement of the obtained values across platforms, in this case, CHARMM[20] and OpenMM.[21]

3. explore and document potential issues and sources of error, particularly as they pertain to the accelerated workflow for polarizable FES with `transformato`.

The next chapter outlines the theoretical underpinnings of this thesis, and provides some insight into, and justification for, the particular approaches used. The methods chapter details the setup of the systems, the parameters used for the simulations, and the post-processing of the generated trajectories to obtain free energy estimates. Next, the results are presented and discussed. The calculated free energy differences are compared to reference and experimental values, and the significance and implications of their agreement — or lack thereof — are evaluated. Finally, special considerations regarding the implementation of polarizability when using `transformato` are addressed.

2 Theory

This chapter provides an overview of the theoretical concepts underlying the computational studies carried out in this work. It covers free energy simulations and molecular dynamics in general, the calculation of free energy differences from simulation data, the implementation of polarizability using the Drude force field, and the use of alchemical transformations to create intermediate states.

1 Solvation free energies

The absolute solvation free energy (ASFE) is the free energy change involved in transferring a solute from vacuum into a solvent. The free energy difference between these two endstates, i.e., no solute-solvent interaction and "full" solute-solvent interaction, can be derived from a thermodynamic cycle as shown in Figure 2.1.

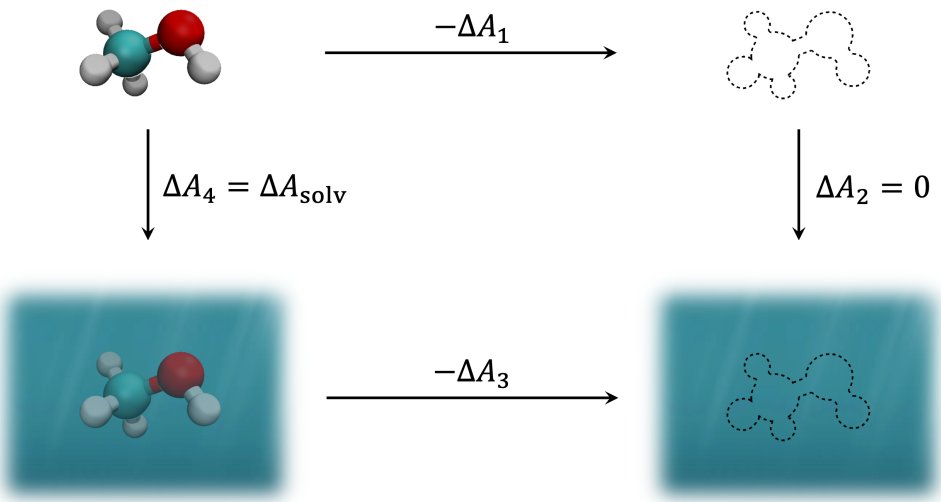

Figure 2.1: The thermodynamic cycle used for ASFE calculations. The horizontal legs represent the annihilation of the solute in vacuum and in solution, associated with free energy changes $-\Delta A_1$ and $-\Delta A_3$, respectively. The vertical legs represent the solvation of the non-interacting (associated with ΔA_2, equaling zero) and native solute (associated with ΔA_4, the quantity of interest).

The horizontal legs of the thermodynamic cycle represent the process of deactivating the non-bonded interactions of the solute in either environment. This process is termed 'annihilation'. Since free energy is a state function, the free energy difference associated with the traversal of the thermodynamic cycle equals zero. By rearranging the resulting sum, the ASFE $\Delta A_{\text{solv}} = \Delta A_4$ can be calculated as:

$$\Delta A_{\text{solv}} = \Delta A_3 - \Delta A_1 \,. \tag{1}$$

The term ΔA_2 is absent from the equation because transferring the annihilated solute from vacuum into the solution does not involve any change in free energy; the solute's remaining self-interactions are unaffected by solvation. Annihilation removes both the intramolecular non-bonded interactions and the intermolecular solvent-solute interactions. Therefore, subtracting (or adding, if the direction is reversed) the contribution of the annihilation *in vacuo* leaves only the free energy change associated with removing the solvent-solute interactions. This is known as the 'gas phase correction.'

2 Molecular dynamics simulations

Molecular dynamics (MD) is a simulation technique used to model the time evolution of molecular systems. In the classical approach, this is achieved by integrating the Newtonian equations of motion. Generally, an MD simulation proceeds as follows:

1. Initialize the system. The positions and velocities of every particle in the system are assigned initial values. The positions may be based on experimental data or lattice arrangements. The velocities are initialized with random values in accordance with a Maxwell-Boltzmann distribution.

2. Compute the forces acting on each particle, typically derived from a potential energy function that describes the interactions between particles.

3. Numerically integrate the Newtonian equations of motion over the time step to determine the new positions and velocities.

4. Repeat the force calculation and integration steps for a specified number of time steps.

The result of such a simulation will be a trajectory that is a sampling from the microcanonical (NVE) ensemble. To sample from the canonical (NVT) ensemble, a 'thermostat' is

employed, an algorithm designed to maintain the system's temperature near a specified target value. Similarly, sampling from the isobaric-isothermal (NPT) ensemble further requires the addition of a 'barostat' to regulate pressure alongside temperature.

The potential energy function describing the force field includes bonded and non-bonded terms. The parametrization and even the inclusion of certain terms differs between force fields. The general form of the additive CHARMM force field, as described by Brooks et al.,[22] is shown in Equation 2:

$$\begin{aligned} U(\vec{R}) = & \sum_{\text{bonds}} K_b(b-b_0)^2 + \sum_{\text{angles}} K_\theta(\theta-\theta_0)^2 \\ & + \sum_{\text{Urey-Bradley}} K_{\text{UB}}(S-S_0)^2 + \sum_{\text{dihedrals}} K_\phi(1+\cos(n\phi-\delta)) \\ & + \sum_{\text{impropers}} K_\omega(\omega-\omega_0)^2 \\ & + \sum_{\text{non-bonded pairs}} \left\{ \epsilon_{ij}^{\min} \left[\left(\frac{R_{ij}^{\min}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{ij}^{\min}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 \epsilon r_{ij}} \right\} \\ & + \sum_{\text{residues}} U_{\text{CMAP}}(\phi,\psi)\,. \end{aligned} \tag{2}$$

Here, $\vec{R}$ is the vector of atomic coordinates.[23] Apart from the dihedrals, the bonded contributions are modeled using a harmonic oscillator, with equilibrium values denoted by the subscript 0. The dihedral term is described by a sinusoidal function, incorporating terms for multiplicity (n) and phase shift (δ). All bonded terms are scaled by their respective force constants (K_x). The Urey-Bradley term, which is not included in all force fields, is only defined for certain combinations of atom types in CHARMM. This term describes the interaction between the first and third atoms that define an angle.

The non-bonded terms consist of a Lennard-Jones (LJ) and a Coulomb potential. The parameters $\epsilon_{ij}^{\min}$ and $R_{ij}^{\min}$ are derived as combinations of the respective atomic parameters for each atom pair i, j. The Coulomb potential is defined between point (partial) charges q_i and q_j. In implicit solvents, the Coulomb interaction may be scaled by a relative dielectric constant ϵ with a value specific to the solvent. In explicit solvent, ϵ is set to one.

For peptides and proteins, CHARMM also includes a correction term (CMAP) for adjacent backbone dihedrals.[24] This term does not apply to the small molecules investigated in this work.

As described in the previous section, the endstates of interest are the solute in its native and annihilated state, either in vacuum or solvated in water. In the annihilated state, contributions to the non-bonded term from pairs involving the solute atoms are equal to zero.

3 Methods to calculate free energy differences

The calculation of free energy differences from simulation data follows from fundamental thermodynamic relations. In the canonical ensemble, the Helmholtz free energy A is related to the partition function by:

$$A = -k_B T \ln Z\,, \tag{3}$$

where k_B is the Boltzmann constant, T is temperature in Kelvin, and Z is the partition function. Consequently, the free energy difference ΔA between state 0 and state 1 is given by:

$$\Delta A = A_1 - A_0 = -k_B T \ln \frac{Z_1}{Z_0}\,. \tag{4}$$

In classical systems, the potential and kinetic contributions to the partition function are separable. The kinetic part can be straightforwardly calculated as a function of system mass and temperature. These quantities do not change upon annihilation of the solute. Therefore, the kinetic parts are equal between states and cancel out, leaving:

$$\Delta A = -k_B T \ln \frac{Q_1}{Q_0}\,, \tag{5}$$

where Q is the configurational partition function. It should be noted that in the systems under investigation, the pressure-volume work is negligibly small, so for all practical purposes, ΔA is equal to the Gibbs free energy ΔG.

Calculating the partition function explicitly is only feasible for very simple cases. Therefore, to obtain reliable estimates of free energy differences from the limited sampling in simulation data, several methods have been developed. Among the most widely used approaches are thermodynamic integration (TI),[25] exponential averaging (also known as the 'Zwanzig relationship'),[26] and Bennett's acceptance ratio (BAR)[27] method. BAR has been extended to a generalization for the simultaneous use of data from multiple states, known as the multistate Bennett acceptance ratio (MBAR)[28] method.

Since TI, BAR, and MBAR were used for the calculations in this work, a brief introduction to these methods follows.

3.1 Thermodynamic integration

Thermodynamic integration employs a coupling parameter, λ, which is used to interpolate between the endstates. Accordingly, the free energy function depends on λ explicitly, i.e.,

$$A(\lambda) = -k_B T \ln Q(\lambda)\,. \tag{6}$$

Taking the derivative of this thermodynamic relation with respect to λ then yields:

$$\frac{\mathrm{d}A}{\mathrm{d}\lambda} = -\beta^{-1}\frac{\mathrm{d}}{\mathrm{d}\lambda}\ln Q = -\beta^{-1}\frac{\frac{\mathrm{d}}{\mathrm{d}\lambda}\int \exp\left[-\beta U(\lambda,\vec{q})\right]\mathrm{d}\vec{q}}{Q}\,, \tag{7}$$

where β is $\frac{1}{k_B T}$ and $\vec{q}$ are the generalized coordinates. The right-hand side of Equation 7 represents an ensemble average, $\langle . \rangle_\lambda$, which can be shown by evaluating the derivative using Leibniz' rule:

$$\frac{\mathrm{d}A}{\mathrm{d}\lambda} = -\beta^{-1}\frac{-\beta\int \frac{\mathrm{d}U(\lambda,\vec{q})}{\mathrm{d}\lambda}\exp\left[-\beta U(\lambda,\vec{q})\right]\mathrm{d}\vec{q}}{Q} = \left\langle \frac{\mathrm{d}U(\lambda,\vec{q})}{\mathrm{d}\lambda}\right\rangle_\lambda\,. \tag{8}$$

By integrating over the whole range of λ, we obtain the free energy difference between the endstates:

$$\Delta A = \int_0^1 \left\langle \frac{\mathrm{d}U(\lambda,\vec{q})}{\mathrm{d}\lambda}\right\rangle_\lambda \mathrm{d}\lambda\,. \tag{9}$$

In practice, the number of simulated λ-states is, of course, finite, requiring a numerical integration, i.e.,

$$\Delta A \approx \sum_{k=1}^{K} w_k \left\langle \frac{\mathrm{d}U(\lambda,\vec{q})}{\mathrm{d}\lambda}\right\rangle_k\,, \tag{10}$$

where w_k is the appropriate weighting factor for the chosen numerical integration method. An error estimate is then obtained by accumulating the variance at each state by Gaussian error propagation:

$$\mathrm{var}(\Delta A) \approx \sum_{k=1}^{K} w_k^2 \mathrm{var}\left(\frac{\mathrm{d}U(\lambda,\vec{q})}{\mathrm{d}\lambda}\right)_k\,. \tag{11}$$

3.2 Bennett acceptance ratio method

The derivations of the BAR method in this section are based on the original 1976 publication.[27] For clarity, *reduced* quantities (i.e., the dimensional potential energies and free energies divided by $k_B T$) are used to simplify the notation.

Bennett demonstrated that the ratio of the configurational integrals of two states is equal to the ratio of their ensemble averages of the Metropolis function, evaluated at the potential energy differences $\Delta U = U_0 - U_1$, and $-\Delta U$:

$$\frac{Q_0}{Q_1} = \frac{\langle M(U_0 - U_1)\rangle_1}{\langle M(U_1 - U_0)\rangle_0}, \tag{12}$$

where Q_0 and Q_1 are the configurational integrals and $\langle . \rangle_0$ and $\langle . \rangle_1$ denote ensemble averages of the respective states. M is the Metropolis function $M(x) = \min(1, \exp(-x))$. The equality holds because the function has the property $\frac{M(x)}{M(-x)} = \exp(-x)$. However, the Metropolis function is neither the only function satisfying this condition nor the most statistically efficient choice. Indeed, Bennett showed that the most optimal estimator (minimizing the expected square error) uses the Fermi function, along with a constant value to shift the origin of the energy function. This results in the (reduced) free energy estimate:

$$\Delta\hat{A} = \ln\left(\frac{\langle f(U_0 - U_1 + C)\rangle_1}{\langle f(U_1 - U_0 - C)\rangle_0}\right) + C, \tag{13}$$

where f is the Fermi function $f(x) = (1 + \exp(x))^{-1}$, and the constant $C = \ln\frac{Q_0 N_1}{Q_1 N_0}$, with N_0 and N_1 corresponding to the finite number of samples from the respective states. This particular expression for C minimizes the statistical error. Because $\frac{Q_0}{Q_1}$ is an unknown quantity, one iteratively searches for the value of C so that the ensemble averages in Equation 13 become equal, i.e.,

$$\langle f(U_0 - U_1 + C)\rangle_1 = \langle f(U_1 - U_0 - C)\rangle_0, \tag{14}$$

balancing the sampling contributions from the forward and reverse transition, and arriving at the self-consistency criterion:

$$\Delta\hat{A} = C - \ln\frac{N_1}{N_0}. \tag{15}$$

The BAR method was initially developed for MC simulation data because MD simulations without temperature control sample from the *microcanonical* ensemble, though significant deviations from the canonical ensemble are not expected for conditions close to standard temperature and pressure. Today, this limitation has been rendered irrelevant; modern thermostat algorithms enable MD simulations to sample effectively from the canonical ensemble.[29]

3.3 Multistate Bennett acceptance ratio method

The following derivations and discussion of the MBAR method are based on the original paper by Shirts and Chodera (2008).[28] MBAR is an estimator for the use of data from multiple states. Free energy estimates are obtained by self-consistently solving a set of K estimating equations for the reduced free energies $\hat{A}_i$:

$$\hat{A}_i = -\ln \sum_{j=1}^{K} \sum_{n=1}^{N_j} \frac{\exp\left[-U_i(x_{jn})\right]}{\sum_{k=1}^{K} N_k \exp\left[\,\hat{A}_k - U_k(x_{jn})\right]}\,, \tag{16}$$

where N_x is the number of samples from the respective state x. Because the normalization constants are unique only up to a multiplicative factor, the resulting free energy estimates will have an additive constant, again leading to only meaningful free energy *differences*, i.e.,

$$\Delta\hat{A}_{ij} = \hat{A}_j - \hat{A}_i\,. \tag{17}$$

MBAR owes its name to the fact that it can be considered a generalization of BAR to multiple states. Consider the $K \times N$ (with N being the total number of samples summed over all states) weight matrix $\mathbf{W}$. The weights are given by:

$$W_{ni} = \hat{Q}_i^{-1} \frac{\exp\left[-U_i(\mathbf{x}_n)\right]}{\sum_{k=1}^{K} N_k \hat{Q}_k^{-1} \exp\left[-U_k(\mathbf{x}_n)\right]}\,, \tag{18}$$

where $\hat{Q}_x$ denotes the configurational partition function estimate of state x. From these weights, an overlap matrix $\mathbf{O}$ of dimensions $K \times K$ can be constructed. An example of such an overlap matrix is illustrated in Table 2.1.

Table 2.1: Overlap matrix for nine states. The entries represent the estimates for the relative phase space overlap. MBAR considers contributions due to overlap in non-adjacent states, represented by non-zero entries with $|i - j| \geq 2$.

State	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	0.93	0.07	0	0	0	0	0	0	0
2	0.07	0.87	0.06	0	0	0	0	0	0
3	0	0.06	0.82	0.09	0.03	0	0	0	0
4	0	0	0.09	0.58	0.21	0.07	0.05	0	0
5	0	0	0.03	0.21	0.47	0.18	0.11	0	0
6	0	0	0	0.07	0.18	0.57	0.18	0	0
7	0	0	0	0.05	0.11	0.18	0.60	0.06	0
8	0	0	0	0	0	0	0.06	0.85	0.09
9	0	0	0	0	0	0	0	0.09	0.91

The overlap matrix **O**, as shown in Table 2.1 is given by:

$$\mathbf{O} = (\mathbf{W}^{\mathsf{T}}\mathbf{W})\mathrm{diag}(\mathbf{n})\,, \tag{19}$$

where diag(**n**) is the diagonal matrix with vector $\mathbf{n} = [N_1, N_2, ..., N_k]$ along its diagonal, resulting in a scaling of the weights column-wise. Then, the entries O_{ij} are estimates of the relative phase space overlap between states i and j, so that each column and row sum is equal to one.

The strength of MBAR is that it considers overlaps between all states. Consider, for example, the overlap between states 4 and 7 in Table 2.1. Now it is trivial to see that when only two states are considered, the only off-diagonal entries are for immediately adjacent states, which means that using BAR pairwise over the transformation path is losing out on the aforementioned additional contributions.

4 Alchemical transformation

The accuracy of calculated free energy differences benefits from inserting intermediate states between the endstates. Estimators like FEP, BAR, and MBAR rely on *phase space overlap*, that is, the set of possible position and momentum configurations (the *phase space*) must be shared to some extent. However, for any two sufficiently different endstates (which is the case for all systems under investigation in this work), the phase space overlap is too small to compute a free energy difference reliably. TI also benefits from additional points along the integration path, especially for integrands with significant curvature over the range.

The solution is to construct a series of intermediate states that represent the change from the 'initial' to the 'final' state. Because free energy is a state function, any path we take within the constraints of the given ensemble is valid in principle. This allows for 'non-physical' intermediate states, in favor of phase space overlap. The two 'alchemical' transformation schemes used in this thesis are 'λ-states' (already broached in Thermodynamic integration) and 'serial atom insertion' (SAI). The use of SAI is in part motivated by avoiding free energy simulation routines that are specific to the particular MD program. One example of such a routine is the use of soft-core potentials for λ-states. These are used to address the so-called 'van der Waals endpoint problem', which is outlined below.

4.1 The van der Waals endpoint problem

The simplest approach to constructing λ-states is by linear interpolation:

$$U(\lambda) = \lambda U_1 + (1-\lambda)U_0\,, \tag{20}$$

where U_0 and U_1 represent the potential energies of the system in the non-interacting and fully interacting state of the solute, respectively. This approach runs into a problem as the coupling parameter approaches zero. The reason for this is the functional form of the Lennard-Jones (LJ) potential. As λ comes close to zero, the second derivative in the region around r_{min} becomes large, i.e., there is an ever increasingly sharp 'kink' in the potential curve. The position of r_{min} also shifts towards zero. Figure 2.2 illustrates the issue.

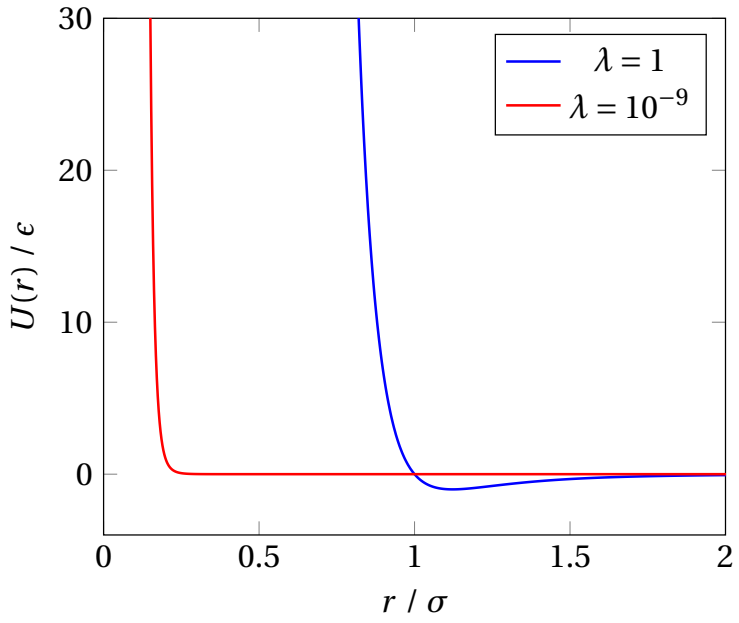

Figure 2.2: The effect of scaling the LJ potential. As λ approaches zero, the steepness of the curve as it moves into the repulsive range becomes much more pronounced.

In combination, this means that the van der Waals forces are now modeled by small, hard spheres. This is problematic because within any reasonable time step, the repulsive forces between particles can easily go from essentially zero to extremely large. Decreasing the time step enough to alleviate this issue would render the simulation impractical, as it would be too computationally expensive.

4.2 λ-states with soft-core potentials

With soft-core potentials, the dependence on the coupling parameter is no more simply linear. Instead, the potentials of the initial and final state both depend on λ explicitly, i.e.,

$$U(\lambda) = \lambda U_1(\lambda) + (1-\lambda)U_0(\lambda). \tag{21}$$

In CHARMM's PERT[22] module, the soft-core potential between two particles i and j is realized with a λ-scaled off-set added to the interparticle distance, i.e.,

$$u_{ij}^{SC}(\lambda) = \lambda\epsilon_{ij}\left(\frac{(R_{ij}^{\min})^{12}}{\left(r_{ij}^2 + (1-\lambda)\delta\right)^6} - 2\frac{(R_{ij}^{\min})^{6}}{\left(r_{ij}^2 + (1-\lambda)\delta\right)^3}\right) \tag{22}$$

where δ has a default value of 5 Å^2. Figure 2.3 shows what this modified LJ soft-core potential looks like at different values of λ.

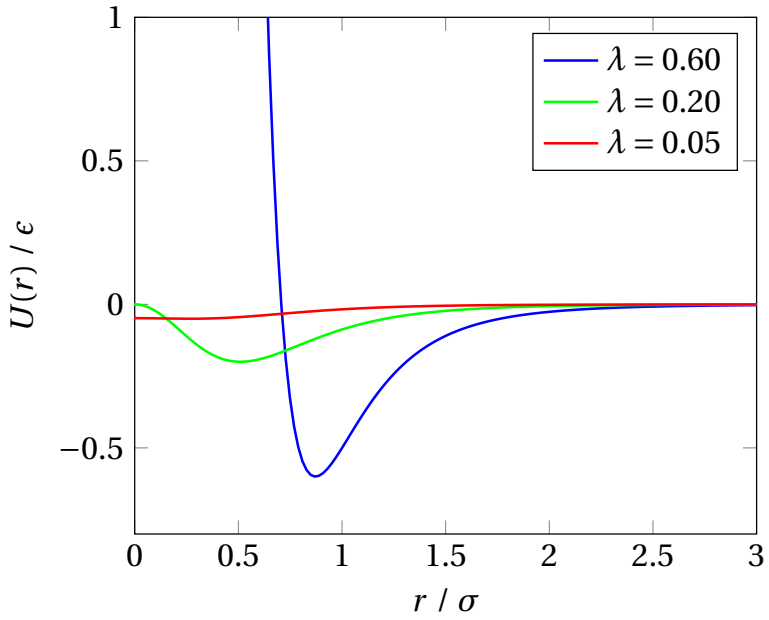

Figure 2.3: A soft-core potential as a function of λ has no singularity and ensures numerical stability as the coupling parameter approaches zero.

As λ decreases, the soft-core potential ensures there are no numerical instabilities during the simulation by keeping the graph smooth; without kinks and discontinuities. Just like in the linear scaling scheme, the graph is flattened and the minimum distance decreases, but at the same time, the repulsive part of the potential is scaled down to remove the singularity.

To summarize, when using the PERT module with soft-cores, the potential energy is scaled between the endstates according to Equation 21 and Equation 22, defining an 'alchemical' path of discrete λ-states, having values between 0 and 1. A schematic representation of this alchemical path over five λ-states is shown in Figure 2.4.

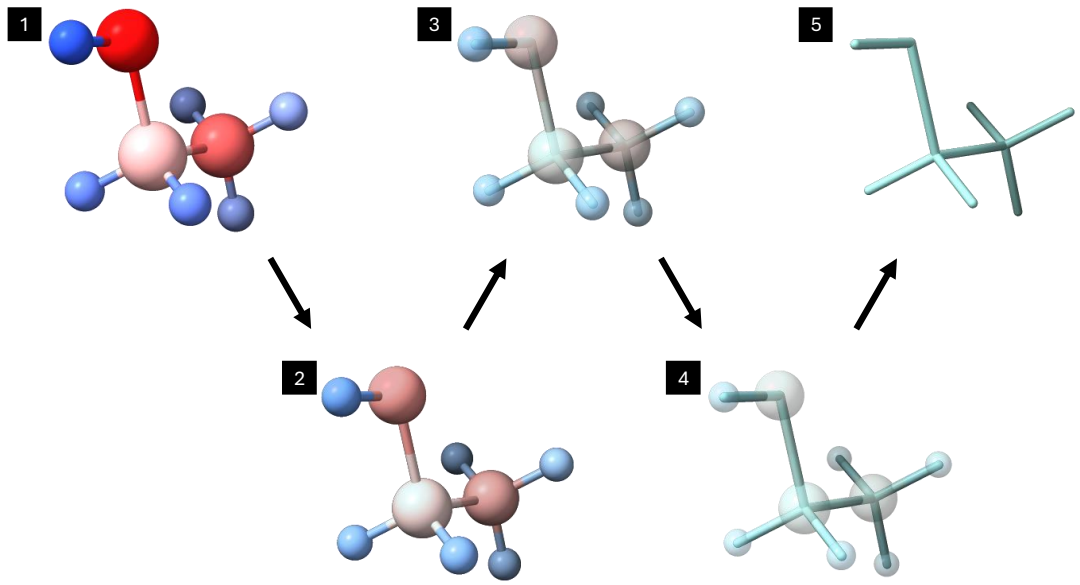

Figure 2.4: Schematic representation of alchemical transformation employing λ-states with soft-cores. States 1 and 5 are the endstates. State 1 is the native molecule with fully active coulombic and LJ interactions, state 5 is the annihilated molecule with coulombic and LJ interactions scaled to zero. The intermediate states (2-4) are generated by interpolation between the endstates, and the use of a soft-core potential.

In the native state (corresponding to the state labeled with '1'), the solute has fully active coulombic and LJ interactions, shown as solid van der Waals spheres with colors corresponding to the atoms' partial charges. The final state (labeled with '5') has the non-bonded interactions turned off. What remains are the bonded interactions. The final state is therefore depicted as a charge-neutral stick model. Accordingly, the scaling of the charges is shown as a loss in saturation, and the scaling of the LJ interactions as both a shrinking of the van der Waals spheres, corresponding to linear interpolation, and a simultaneous loss of opacity, which represents the impact of the soft-core potential. As depicted, both the charges and LJ interactions of the whole solute are scaled simultaneously. Alternatively, one can also split the scaling of the charges and the LJ interactions. This approach is in principle equivalent, and should yield the same free energy differences over the series of intermediate states as in the simultaneous scaling approach.

4.3 Serial atom insertion

In the SAI[30] approach, the scaling of LJ interactions is not achieved by a 'smooth', global interpolation between the initial and final state. Instead, they are scaled down atom-by-atom. In this way, the use of FES-specific routines, such as the soft-core potential, can be avoided. More generally, the intermediate states generated by SAI can be used as input for plain MD, which can be run on GPU and is therefore massively accelerated. The approach is shown schematically in Figure 2.5.

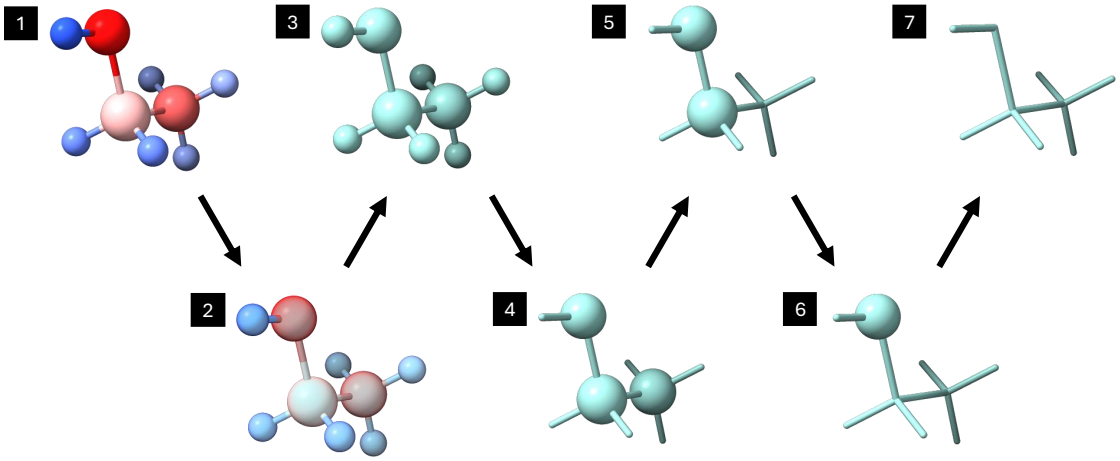

Figure 2.5: General scheme of the transformation path implementing serial atom insertion (SAI). The charges of the native, fully interacting molecule (1) are scaled down stepwise (2,3). Then, the Lennard-Jones interactions are turned off with the SAI approach, first targeting all hydrogens (4), and then the heavy atoms one by one (5-7), leaving only the bonded terms (7).

First, all partial charges are scaled down in a few steps. Then, the LJ interactions are removed; first all hydrogens simultaneously, and then the heavy atoms one by one. In the process, the affected atoms change type to become dummy atoms which have no non-bonded interactions.

5 Polarizability

The explicit treatment of polarizability enhances the physical realism of an MD simulation. In additive force fields, the effects of induced polarization are incorporated only in an averaged manner through the parametrization of effective partial charges. Therefore, from a theoretical standpoint, treating polarizability explicitly should also yield more accurate results, which has been demonstrated in computational studies for various properties and systems,[11,12,14,15] including the ASFEs of a test set of small molecules in water.[13]

In MD simulations, the classical treatment of polarizability employs three commonly used approaches: the induced point dipole,[31] fluctuating charge,[32] and Drude oscillator models.[16] Each comes with certain advantages and disadvantages.

In the induced point dipole model, atoms have fixed charges, and the inducible dipoles are modeled as vectors originating at the atom positions.[31] This model accurately captures the effects of polarizability for a range of applications, but suffers from computational overhead and the fact that it requires specific subroutines.[33]

In the fluctuating charge model, the atomic partial charges themselves are variable.[32] An advantage is that this incurs less computational overhead than the induced point dipole model does. However, it is unable to capture out-of-plane polarizability, inherently limiting its application.[33]

In all three models, the most accurate determination of induced dipoles depends on solving a set of self-consistent field (SCF) equations. This is computationally expensive. So called 'extended Lagrangian' methods, which treat the polarizable degree of freedom under the dynamics regime, accelerate the process.[34]

For the Drude oscillator model, it has been demonstrated that the extended Lagrangian approach successfully reproduces the SCF regime to a high degree of accuracy.[35] Using this, the Drude model offers a method for the explicit treatment of polarizability that is both relatively fast, and straightforward in its implementation.[36] Because it is the method chosen for this thesis, a more detailed description of the Drude model and aspects of its implementation follows.

5.1 Drude oscillator model

In the Drude oscillator model, polarizability is achieved by redistributing the charge of the original non-polarizable atom across a 'Drude pair.' This pair consists of a small, mobile particle, referred to as the 'Drude particle,' and a larger, stationary particle, here termed the 'parent atom'. The principle is shown in Figure 2.6.

The Drude particle is kept close to the parent atom by a harmonic potential with an equi-

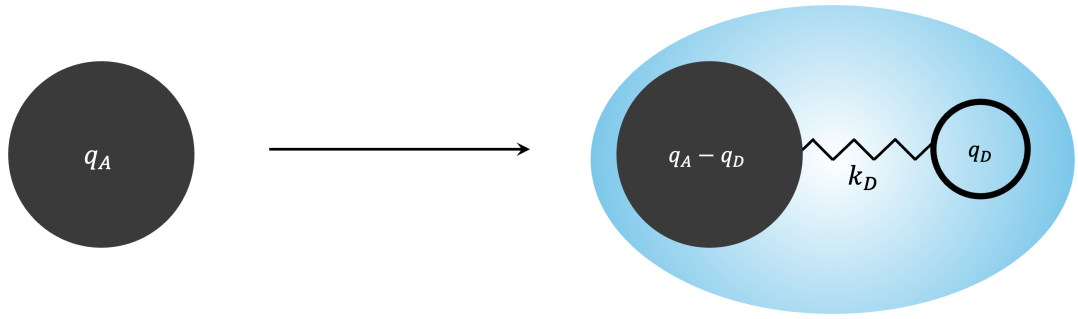

Figure 2.6: Schematic representation of the Drude oscillator model. The non-polarizable atom is split into a parent atom and Drude particle pair, connected by a harmonic potential, here pictured as a spring. The charge of the pair is equal to that of the non-polarizable atom.

librium distance of zero. The partial charges are split such that their sum across the pair is equal to the partial charge on the non-polarizable atom. The atomic polarizability is given by:

$$\alpha = \frac{{q_D}^2}{k_D}, \tag{23}$$

where k_D is the spring constant of the harmonic bond potential between the pair.[35] In the original formulation, the Drude particles are massless, and their positions are solved self-consistently at each step before moving on to the next, modeling the instantaneous relaxation required by the Born-Oppenheimer approximation.[35]

As already mentioned, this incurs significant computational overhead. An alternative is the extended Lagrangian approach. The Drude particles are assigned a small mass m_D, which then allows the treatment of the Drude particles under the classical dynamics regime of the rest of the system, which is possible in one step.[35,36] A disadvantage is that the need to assign the Drude particles a small mass precludes their use for hydrogens.

5.2 CHARMM Drude force field

CHARMM's polarizable force field[36] is an implementation of the Drude model with the extended Lagrangian approach. The mass of the Drude particle m_D has a default value of 0.4 amu, which, like the partial charge, is subtracted from the parent atom.[35] The contribution to the force field (cf. Equation 2) due to the Drude particles is given by the term:

$$U_{\text{Drude}} = \frac{1}{4\pi\epsilon_0}\left(\sum_{i<j}\frac{q_{D,i}q_j}{\|\mathbf{r}_{D,i}-\mathbf{r}_j\|} + \sum_{i<j}\frac{q_{D,i}q_{D,j}}{\|\mathbf{r}_{D,i}-\mathbf{r}_{D,j}\|}\right) + \frac{1}{2}\sum_{\text{i}} k_D\|\mathbf{r}_{D,i}-\mathbf{r}_i\|^2, \tag{24}$$

where subscript D denotes a Drude particle, i refers to a polarizable atom, and j runs over all interacting sites, polarizable or not.[37]

Polarizable (heavy) atoms are defined in the residue topology by their isotropic atomic polarizability α. Together with a set value for k_D, by default 500 kcal mol^{-1}Å^{-2}, the Drude partial charge q_D is determined by Equation 23. In the additive force field, non-bonded interactions are excluded for 1,2 and 1,3 pairs. The Drude force field also has these exclusions in place, but does allow for interactions between Drude particles belonging to these atom pairs. These are, however, damped by a 'Thole-like' screening function:

$$S_{ij}(r_{ij}) = 1 - \left(1 + \frac{(a_i + a_j)r_{ij}}{2(\alpha_i\alpha_j)^{1/6}}\right)\exp\left[-\frac{(a_i + a_j)r_{ij}}{(\alpha_i\alpha_j)^{1/6}}\right], \tag{25}$$

where a_i and a_j are damping parameters specific to the respective atoms.[37,38]

The CHARMM Drude FF is further refined by accounting for the anisotropy of charge distribution and polarizability. The former is accounted for by the inclusion of lone pairs; these are negative point charges set at a fixed distance from their atom. Lone pairs are only applied to certain atom types, like hydrogen bond acceptors. Anisotropy of polarizability is implemented with a force constant tensor $\mathbf{K_D}$, derived from the anisotropic polarizability tensor $\boldsymbol{\alpha}$ (given in the residue topology) by $\mathbf{K_D} = -q_D^2\boldsymbol{\alpha}^{-1}$. The polarizability tensor is diagonal, which results in three different force constants, one for each spatial dimension. The Thole damping is still applied isotropically, even if anisotropic Drudes are involved.[37]

5.3 Drude integrators

In the extended Lagrangian approach, Drude particles are subject to the same thermal fluctuations as all the other physical particles in the system. It has been shown that thermostatization of the Drude degrees of freedom at system temperature (usually around 300 K) leads to energy drift and stability issues.[35] The solution is to keep the thermal kinetic energy of the Drude oscillators as low as possible. To avoid overdamping, the value cannot be arbitrarily close to absolute zero. The commonly employed value is 1 K. This is achieved by a dual thermostat, which applies thermostatization to the atomic and Drude degrees of freedom separately. Currently, there exist two of these dual thermostats in CHARMM: the dual Nose-Hoover and the dual Langevin thermostat. These are also available in OpenMM. It has been shown that Langevin dynamics are not well-suited for dynamic properties, both in the polarizable and non-polarizable systems.[15,39] However, this does not necessarily have any bearing on the choice of integrator when one is interested in a static property like the free energy.

5.4 Water model

The majority of polarizable sites in a biochemical system reside in the solvent molecules, which lends great significance to the choice of solvent model. The water model employed in this work is SWM4-NDP.[40,41] It has four interaction sites: the two hydrogens, a virtual 'M' site, and the oxygen atom. A schematic representation is shown in Figure 2.7.

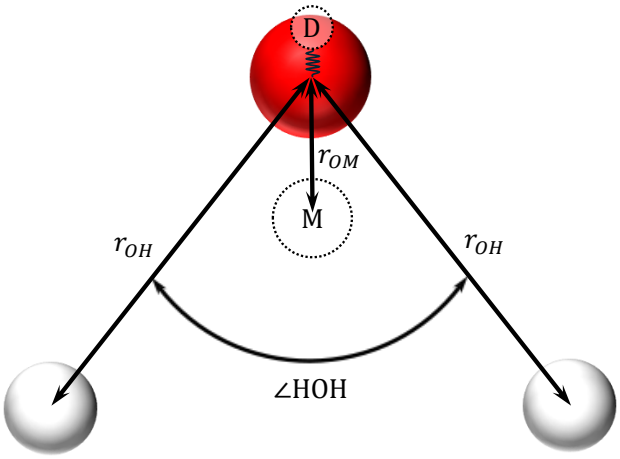

Figure 2.7: Schematic representation of the SWM4-NDP model.

The model aligns with experimental gas phase parameters, with an HOH angle of 104.52° and a dipole of 1.85 D when isolated. The oxygen carries no net partial charge, but models induced polarization with a Drude pair. The permanent charge distribution is achieved with fixed partial charges $q_H \approx 0.557$ on the hydrogens, and partial charge $-2q_H$ on the 'M' site, which is placed between the oxygen and hydrogens along the HOH bisector, off-set by $r_{OM} \approx 0.24$ Å from the oxygen atom.

3 Methods

The test set of molecules used in this thesis is pictured in Figure 3.1. These compounds were already studied in the reference paper by König et al.;[13] only aniline is missing since in the most recent version (2023) of the CHARMM Drude polarizable force field,[42] no parameters for this molecule are available.

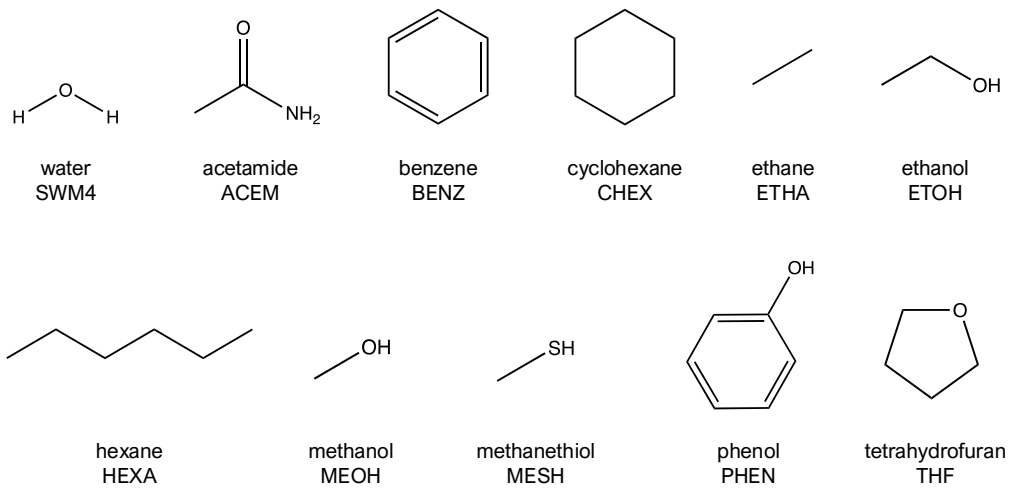

Figure 3.1: The test set of molecules used in this work. Both their trivial and residue names are given.

The set covers a wide range of hydration free energies, and includes several side-chain analogs (acetamide, benzene, methanol, methanethiol, and phenol). Below the compound names in Figure 3.1, we also list their abbreviations,[1] which will be used in the tables and figures in the discussion of the results.

[1] These are simply the residue names used in the residue topology file.

Reference calculations using CHARMM[20] were performed using two distinct approaches:

1. **PERT**: Utilizing the PERT module to simulate λ-states and calculate results via thermodynamic integration. This most closely resembles the method used by König et al.[13]

2. **CHARMM+SAI**: Employing serial atom insertion (SAI), with intermediate states generated by `transformato`. This method involves conducting plain MD simulations followed by post-processing and analysis using the BAR method.

The approach to be validated will be referred to as:

> **OMM+SAI**: This method combines SAI with OpenMM (using the CHARMM Drude force field). It uses the same intermediate states as the CHARMM+SAI approach, but simulations are carried out with OpenMM and analysis is carried out using MBAR.

Additional details of each method are provided below. The generation of intermediate states using `transformato` is covered in a separate section. Table 6.1 in the appendix gives a comparison of the settings used in the three approaches.

1 Simulation parameters

First, this section specifies the 'general' settings used. These are identical for the CHARMM methods (PERT and CHARMM+SAI). The OMM+SAI settings are similar, but there are backend-specific differences. Next, the simulation protocols (intermediate states, equilibration steps, and step sizes) are outlined for each of the approaches.

1.1 General settings

The CHARMM reference simulations, **PERT** and **CHARMM+SAI**, employed the dual Langevin thermostat (as implemented in TPCONTROL[43]) alongside the VV2 integrator. The target temperature for standard degrees of freedom was set to 298.15 K, while Drude degrees of freedom were maintained at 1 K. The friction coefficients were 5 ps^{-1} and 10 ps^{-1}, respectively. The Anderson barostat is set to 1 atm, with a response time of 0.2 ps. In the gas phase, no truncation of interactions was applied. For the water boxes, the potential-based switching function VSWITCH was used for Lennard-Jones truncation between 10 and 12 Å. Coulombic interactions were treated with particle-mesh Ewald summation (PME), using $\kappa = 0.34$ $Å^{-1}$, a spline order of 6, and a $24 \times 24 \times 24$ grid. The timestep in all simulations was 1 fs.

The **OMM+SAI** simulations employed the DrudeLangevinIntegrator.[44] The system temperature was set to 303.15 K, the Drude temperature to 1 K, and the respective friction coefficients were 10 ps^{-1} and 200 ps^{-1}. In the gas phase, there is no cutoff for the non-bonded interactions. In the water box, the default OpenMM potential switching function was used, with an onset of 10 Å and a cutoff of 12 Å. For electrostatics, the PME method is employed in the water box, with an error tolerance of 0.0005. The PME parameters are set internally by OpenMM based on the error tolerance and the cutoff used by the switching function. The simulations were carried out using OpenCL with single-precision floating-point arithmetic.

In all three approaches, the distance between Drude particles and their parent atoms was limited to 0.2 Å with a "hard wall" constraint. All simulation series were repeated five times.

1.2 Simulation protocols

PERT

The initial states were the same ones used as the starting inputs for `transformato` (cf. Inputs), i.e., a PSF of the native state and a pre-equilibrated coordinate file, generated with CHARMM. Simulations used 21 λ-values (0, 0.05, 0.10, 0.15, ..., 0.95, 1.0), with five repetitions conducted for each. In the gas phase, equilibration consisted of 500k steps, followed by 4.5M production steps per λ-state. For the water boxes, equilibration lasted 100k steps, followed by 200k production steps. Longer protocols were used for hexane and cyclohexane simulations, with 5M equilibration steps and 45M production steps in the gas phase, and 200k equilibration steps followed by 500k production steps in the water box.

CHARMM+SAI

The intermediate states generated by `transformato` were employed (see Generation of intermediate states with `transformato` below). For the gas phase, simulations used 1M equilibration steps followed by 10M production steps, water box simulations ran for 100K equilibration steps and 1M production steps. Longer protocols were again applied for hexane and cyclohexane, with 1M equilibration steps and 20M production steps in the gas phase, and 200k equilibration steps followed by 2M production steps in the water box.

OMM+SAI

The same intermediate states as those for CHARMM+SAI were used. As a stability measure, a "ramping protocol" was applied, where the simulation was run for 100k steps at time steps of 0.1, 0.2, 0.3, and 0.4 fs each, totaling 400k steps. Both the gas phase and solvated systems

ran for an additional 10M, or in the case of hexane and cyclohexane, 100M steps, using a production timestep of 0.5 fs.

2 Analysis

2.1 PERT

To carry out thermodynamic integration, first, averages $\langle \partial U/\partial \lambda \rangle_\lambda$ were extracted from the output. Then, instead of integrating with commonly used numerical quadrature schemes like the Simpson's rule or Clenshaw-Curtis, the $\langle \partial U/\partial \lambda \rangle_\lambda$ values, as a function of λ, were fitted to a spline, and the resulting spline function was integrated analytically. For details and how errors were estimated, see the Supporting Information in Fleck et al.[45]

2.2 CHARMM+SAI

The free energy differences were obtained as the sum of free energy differences between neighboring states, calculated using BAR. The needed energy differences were obtained in a post-processing pass, resulting in 20k energy differences from state i to $i+1$ and $i+1$ to i, respectively (40k energy differences for hexane and cyclohexane). The energy differences from the five repeats were concatenated, and the free energy difference obtained with BAR from these 5×20k (5×40k) energy differences was considered the most reliable free energy estimate between states i and $i+1$. Free energy differences were also calculated for each of the five repeats separately; the standard deviation between these five values was considered the error estimate. The total free energy difference in gas phase and aqueous solution was then computed as the sum $\sum_i \Delta G_{i,i+1}$. Gaussian error propagation was used to obtain the error estimate for the free energy of solvation.

2.3 OMM+SAI

The analysis script is generated by `transformato`. Each trajectory is composed of 5k coordinate frames (50k in the longer protocols for hexane and cyclohexane). The first quarter was discarded to ensure the analysis is performed on an equilibrated state. The script uses the pymbar[28] library to carry out MBAR (see Multistate Bennett acceptance ratio method). Each of the five runs is processed independently of the others, producing five free energy estimates. The mean of these is considered to be the final free energy estimate, and the standard deviation between them the error estimate.

3 Generation of intermediate states with `transformato`

3.1 Inputs

The starting point of the `transformato` workflow is a structure (PSF) and pre-equilibrated coordinate file, which, for all the systems in this study, were generated with CHARMM. The structure files for the solvated systems were generated by placing the solute in a $26 \times 26 \times 26$ Å box of 583 SWM4-NDP[41] waters, and deleting molecules overlapping with the solute. The coordinates for the gas phase were optimized with the adopted basis Newton-Raphson (ABNR) minimizer. Those for the water box were minimized first by the steepest descent (SD), then by the ABNR method. Additionally, `transformato` requires a structure file of the small molecule, without Drude particles, in the SDF or MOL2 format.

3.2 Generation of intermediate states

Based on the above described inputs, `transformato` creates intermediate states, implementing the general SAI approach already discussed (see Serial atom insertion).

For the systems in this study, the charges were scaled down in three steps ($1.00 \rightarrow 0.67 \rightarrow 0.33 \rightarrow 0.00$). Then, the hydrogens' LJ parameters were scaled down in three steps ($1.00 \rightarrow 0.75 \rightarrow 0.38 \rightarrow 0.00$). The LJ parameters of the heavy atoms were subsequently turned off in one step each, except for the last one, which was scaled down in two steps ($1.00 \rightarrow 0.50 \rightarrow 0.00$). In practice, when scaling down the LJ parameters of an atom, its atom type is replaced in the structure file by a dummy atom, and all required parameter values for this dummy atom type are written into a separate parameter file, specifically for dummy atoms. Any additional scaling of the atom only alters the values of this new entry. Each of the steps, both for Coulomb and LJ interactions, corresponds to a "standalone" intermediate state, which can be simulated independently of the others.

The variable parts of the structure files are then merged with a CHARMM-generated template file (of the native state) to ensure inclusion of Drude force field specific sections and CHARMM compatible formatting (see Considerations for the Drude force field). This merge is done outside the `transformato` framework.

4 Results and discussion

To validate the implementation of the SAI approach in OpenMM with the CHARMM polarizable Drude force field, the results are compared to well-validated reference values. To this end, the reference results by König et al.[13] were corroborated by reproducing them in CHARMM with the PERT module, and with the SAI approach, as described in the Methods section. Deviations, their significance, and their possible origins are discussed. All free energy values tabulated and discussed below are in kcal/mol.

1 CHARMM calculations

The results of the CHARMM calculations conducted are summarized in Table 4.1. The reference ASFE values obtained by König et al. are listed under $\Delta G_{\text{König}}$, given as mean values, along with their error estimates. The same is true for the PERT and CHARMM+SAI results, which are in the ΔG_{PERT} and $\Delta G_{\text{CHARMM+SAI}}$ columns, respectively. The difference between these ΔG values and the reference values by König et al. are listed in the respective $\delta\Delta$ columns. The column containing the reference values is shaded grey for clarity. The root-mean-square errors (RMSE) compared to the reference values are presented in the last row. These formatting choices will also apply to the other results tables.

Table 4.1: Comparison of the results of the CHARMM calculations (PERT and SAI) versus reference values by König et al.

Molecule [a]	ΔG_{PERT} [b]	$\delta\Delta G_{\mathrm{PERT}}$ [c]	$\Delta G_{\mathrm{CHARMM+SAI}}$ [d]	$\delta\Delta G_{\mathrm{CHARMM+SAI}}$ [e]	$\Delta G_{\mathrm{König}}$ [f]
SWM4	-5.77 ± 0.05	0.00	-5.75 ± 0.10	0.02	5.77 ± 0.02
ACEM	-10.90 ± 0.14	-0.26	-10.49 ± 0.26	0.15	-10.64 ± 0.06
BENZ	-1.25 ± 0.12	0.11	-1.45 ± 0.17	-0.09	-1.36 ± 0.05
CHEX	2.20 ± 0.12	0.05	2.06 ± 0.10	-0.09	2.15 ± 0.04
ETHA	2.16 ± 0.11	0.00	2.03 ± 0.07	-0.13	2.16 ± 0.02
ETOH	-4.75 ± 0.09	-0.10	-4.72 ± 0.27	-0.07	-4.65 ± 0.05
HEXA	2.71 ± 0.13	0.17	2.51 ± 0.16	-0.03	2.54 ± 0.08
MEOH	-4.95 ± 0.11	-0.05	-4.91 ± 0.09	-0.01	-4.90 ± 0.03
MESH	-1.01 ± 0.10	0.03	-0.94 ± 0.16	0.10	-1.04 ± 0.03
PHEN	-7.26 ± 0.21	-0.19	-7.07 ± 0.13	0.00	-7.07 ± 0.03
THF	-3.29 ± 0.15	-0.17	-3.33 ± 0.14	-0.21	-3.12 ± 0.03
RMSE [g]		0.13		0.10	

[a] The abbreviations used in this column are the residue names defined in the residue topology file (cf. Figure 3.1) [b] ASFE mean values and error estimates of the PERT approach. [c] Differences in ASFE mean values between the PERT approach and the reference calculations [d] ASFE mean values and error estimates of the CHARMM+SAI approach. [e] Differences in ASFE mean values between the CHARMM+SAI approach and the reference calculations [f] ASFE mean values and error estimates of the reference calculations by König et al.[13] [g] Root-mean-square error compared to the reference values.

Both the PERT and CHARMM+SAI approaches yield quite small deviations from the reference values, with RMSEs of 0.13 and 0.10 kcal/mol, respectively. Applying Welch's tests to the summary statistics reveals that, at a confidence level of 95%, PERT shows statistically significant differences only in the values for acetamide ($p = 0.011$) and hexane ($p = 0.043$). For CHARMM+SAI, it is only ethane ($p = 0.012$) and tetrahydrofuran ($p = 0.027$).

Differences in the approach by König et al. are the use of replica exchange to increase sampling, as well as the use of much longer protocols, with a production time of 5 ns per λ-state in the aqueous phase, compared to, for example, 0.2 ns (or 0.5 ns for hexane and cyclohexane) in our PERT approach.

The CHARMM-based calculations are in good agreement with the reference values. The differences to $\Delta G_{\text{König}}$, listed in Table 4.1, are shown in Figure 4.1 as bars, with the error estimates indicated as whiskers.

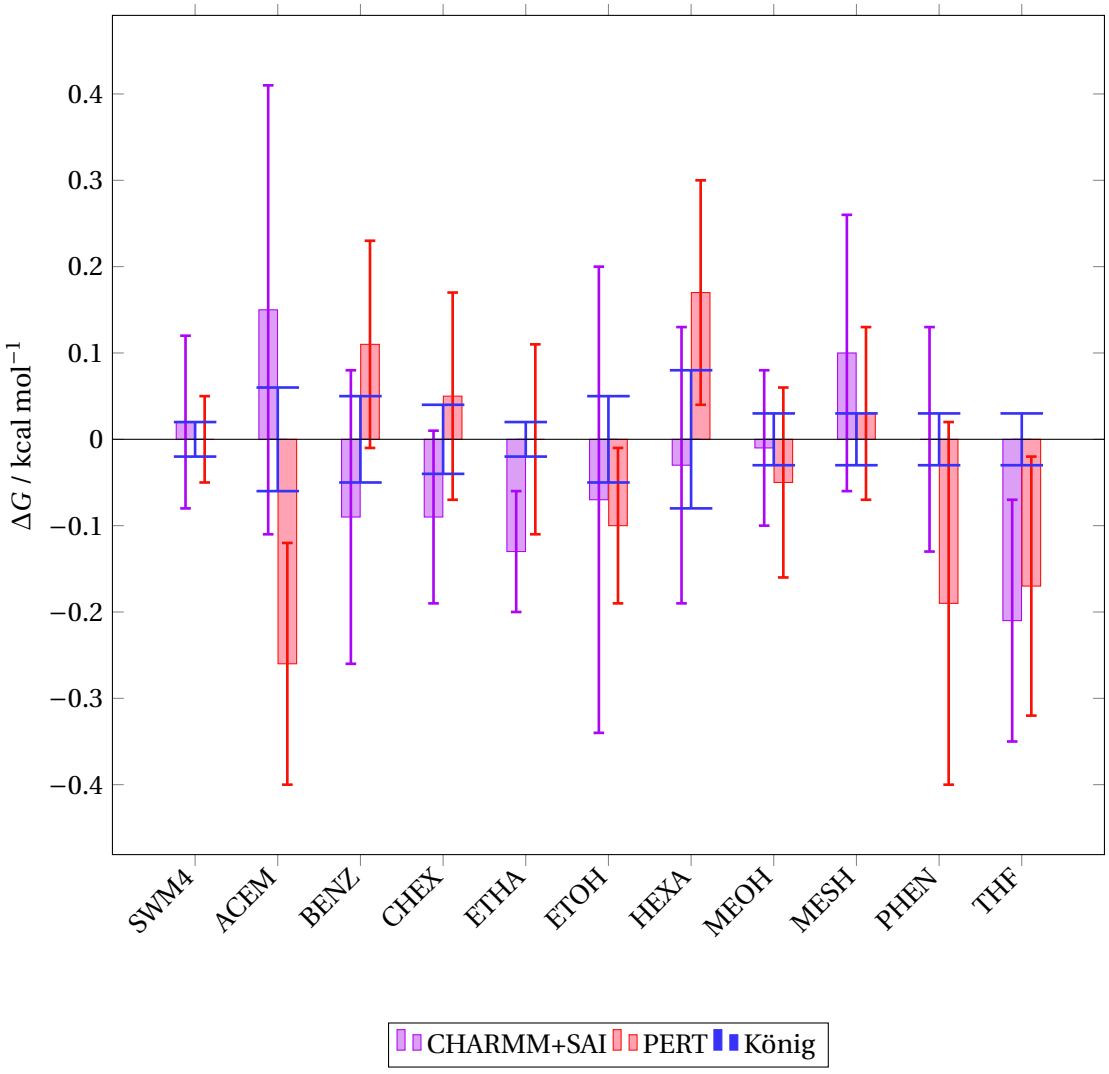

Figure 4.1: Differences of ΔG values and their error ranges for CHARMM+SAI (purple) and PERT (red) calculations, compared to reference values by König et al.[13] (blue whiskers).

2 SAI with OpenMM

Next, the validity of the SAI approach implemented with OpenMM is assessed. To this end, the results are once again compared with the reference values by König et al.. A summary of the results is shown in Table 4.2.

Table 4.2: Comparison of the SAI approach with OpenMM versus reference values by König et al.

Molecule [a]	$\Delta G_{\text{OMM+SAI}}$ [b]	$\delta\Delta G_{\text{OMM+SAI}}$ [c]	$\Delta G_{\text{König}}$ [d]
SWM4	-5.90 ± 0.10	-0.13	-5.77 ± 0.02
ACEM	-10.08 ± 0.10	0.56	-10.64 ± 0.06
BENZ	-1.06 ± 0.12	0.30	-1.36 ± 0.05
CHEX	2.18 ± 0.04	0.03	2.15 ± 0.04
ETHA	2.14 ± 0.07	-0.02	2.16 ± 0.02
ETOH	-4.33 ± 0.11	0.32	-4.65 ± 0.05
HEXA	3.00 ± 0.07	0.46	2.54 ± 0.08
MEOH	-4.60 ± 0.14	0.30	-4.90 ± 0.03
MESH	-1.03 ± 0.20	0.01	-1.04 ± 0.03
PHEN	-6.72 ± 0.14	0.35	-7.07 ± 0.03
THF	-3.11 ± 0.05	0.01	-3.12 ± 0.03
RMSE [e]		0.29	

[a] The abbreviations used in this column are the residue names defined in the residue topology file (cf. Figure 3.1). [b] ASFE mean values and error estimates of the OMM+SAI approach. [c] Differences in ASFE mean values between the OMM+SAI approach and the reference calculations. [d] ASFE mean values and error estimates of the reference calculations by König et al.[13] [e] Root-mean-square error compared to the reference values.

With an RMSE of 0.29 kcal/mol, the results of the OMM+SAI approach do not agree as well with the reference values compared to the results obtained with the CHARMM calculations. Interestingly, the results are quite mixed, either showing very good agreement or (highly) significant differences. Again, the data from Table 4.2 are shown in Figure 4.2 as bar plots with whiskers.

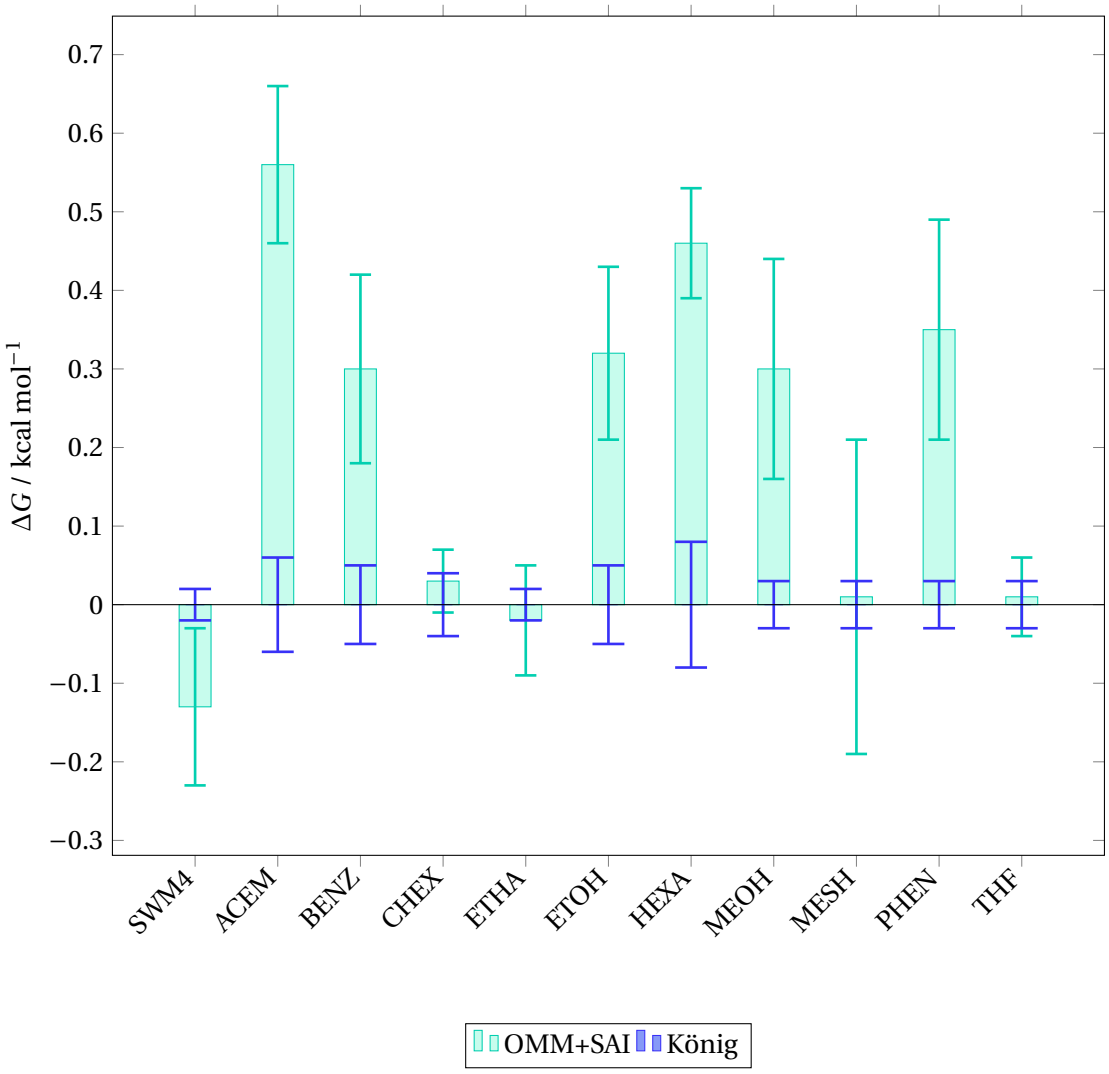

Figure 4.2: Differences and error ranges of OMM+SAI values (turquoise) compared to the reference values by König et al. (blue whiskers).

It is clear to see that the values for cyclohexane, ethane, methanethiol, and tetrahydrofuran are in very good agreement ($p = \{0.27, 0.57, 0.92, 0.71\}$, respectively). The difference in the values for water is significant ($p = 0.042$) at a 95% confidence interval. The differences for the rest of the compounds are highly significant ($p << 0.05$).

Interestingly, all the values that differ significantly do so in the positive direction. Additionally, it is worth noting that the OMM+SAI approach consistently exhibits broader error ranges. This is despite the fact that the OMM+SAI simulation protocols are just as long, or — in the case of cyclohexane and hexane — significantly longer.

Next, the two SAI approaches are compared. Since the intermediate states used are identical[1], significant deviations can be attributed to the use of two different MD programs. The results obtained with OpenMM are compared to those generated directly in CHARMM, as shown in Table 4.3 under $\Delta G_{\text{OMM+SAI}}$ and $\Delta G_{\text{CHARMM+SAI}}$, respectively.

Table 4.3: Comparison of the SAI approach with OpenMM versus SAI implemented with CHARMM directly.

Molecule [a]	$\Delta G_{\text{OMM+SAI}}$ [b]	$\delta\Delta G_{\text{OMM+SAI}}$ [c]	$\Delta G_{\text{CHARMM+SAI}}$ [d]
SWM4	-5.90 ± 0.10	-0.15	-5.75 ± 0.10
ACEM	-10.08 ± 0.10	0.41	-10.49 ± 0.26
BENZ	-1.06 ± 0.12	0.39	-1.45 ± 0.17
CHEX	2.18 ± 0.04	0.12	2.06 ± 0.10
ETHA	2.14 ± 0.07	0.11	2.03 ± 0.07
ETOH	-4.33 ± 0.11	0.39	-4.72 ± 0.27
HEXA	3.00 ± 0.07	0.49	2.51 ± 0.16
MEOH	-4.60 ± 0.14	0.31	-4.91 ± 0.09
MESH	-1.03 ± 0.20	-0.09	-0.94 ± 0.16
PHEN	-6.72 ± 0.14	0.35	-7.07 ± 0.13
THF	-3.11 ± 0.05	0.22	-3.33 ± 0.14
RMSE [e]		0.31	

[a] The abbreviations used in this column are the residue names defined in the residue topology file (cf. Figure 3.1). [b] ASFE mean values and error estimates of the OMM+SAI approach. [c] Differences in ASFE mean values between the OMM+SAI approach and the reference CHARMM+SAI calculations. [d] ASFE mean values and error estimates of the CHARMM+SAI calculations [e] Root-mean-square error compared to the reference values.

With an RMSE of 0.31 kcal/mol, the overall agreement of OMM+SAI with CHARMM+SAI is about as good as it is with the reference values by König et al. However, except for methanethiol ($p = 0.46$) and cyclohexane ($p = 0.053$), they all differ significantly at 95% confidence. The corresponding bar plot with whiskers is shown in Figure 4.3.

[1] There is one additional charge scaling step for phenol in the CHARMM+SAI approach because without it, there was insufficient overlap for BAR.

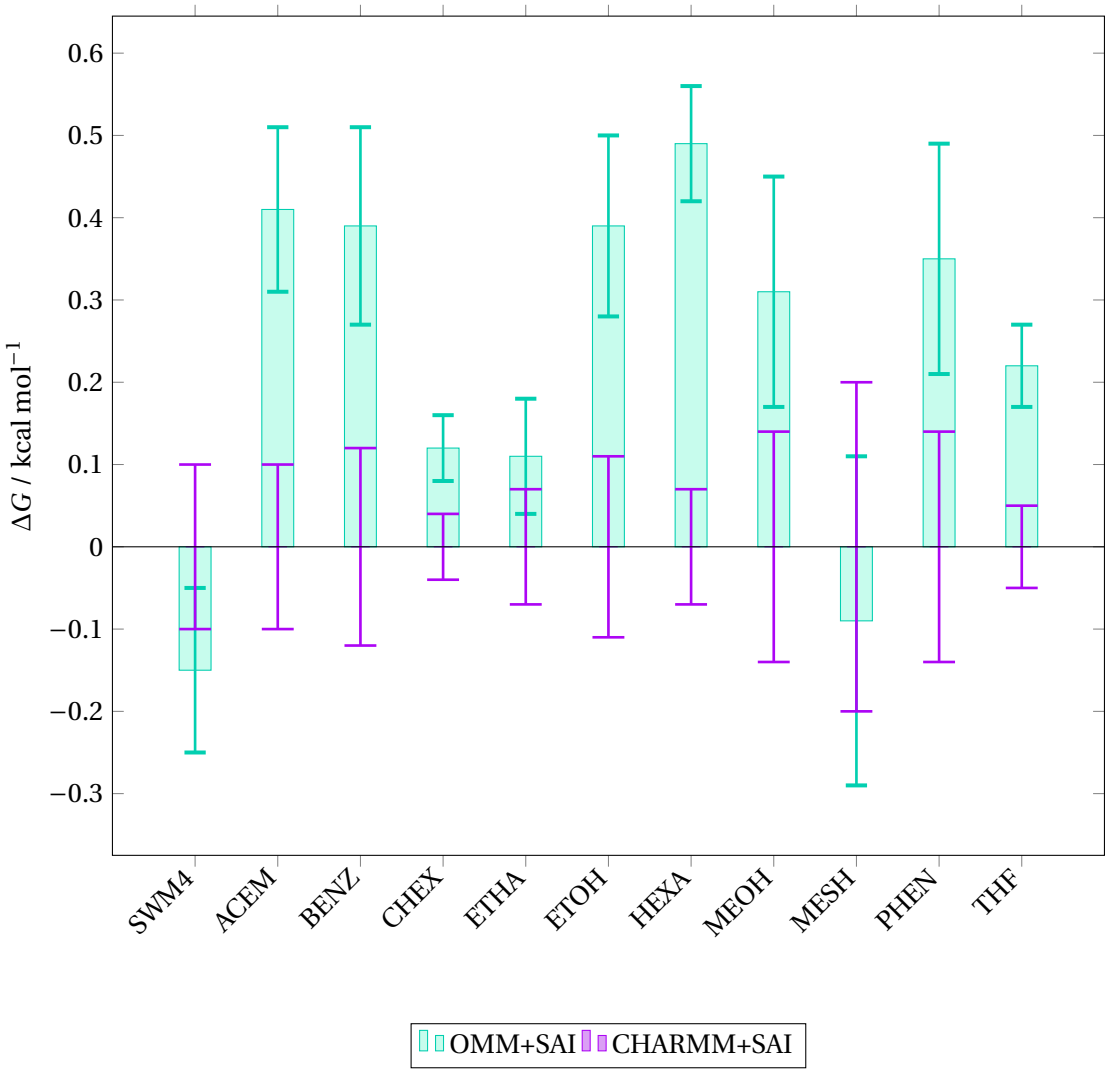

Figure 4.3: Differences and error ranges of OMM+SAI values (turquoise) compared to the CHARMM+SAI values (purple whiskers).

All the differences lie within ±0.5 kcal/mol, and again, all the significant differences are in the positive direction, suggesting a systematic bias in the OMM+SAI approach. There is a wide range of possible causes for this bias, such as the different LJ switching functions used, the specific implementation of dual integrators in OpenMM versus CHARMM, and the specificities in the treatment of Drude particles in the two engines. Any detailed analysis should be carried out on a larger dataset.

3 Agreement with experiment

Table 4.4 shows the results of the SAI approaches next to those by König et al. and compares them to experimental values from the latest Mobley set (FreeSolv database),[46] which is a collection of experimental free energies of hydration for hundreds of compounds from a variety of sources. The experimental value for water was taken from the Minnesota Solvation Database.[47,48] Because for most of these values, there is no originally published error estimate available, they are omitted (in these cases, in the Mobley set, a 'standard error' of 0.6 kcal/mol is assumed).

Table 4.4: SAI approaches and reference values by König et al. compared to experimental values.

Molecule [a]	$\Delta G_{\text{OMM+SAI}}$ [b]	$\delta\Delta G_{\text{OMM+SAI}}$ [c]	$\Delta G_{\text{CHARMM+SAI}}$ [d]	$\delta\Delta G_{\text{CHARMM+SAI}}$ [e]	$\Delta G_{\text{König}}$ [f]	$\delta\Delta G_{\text{König}}$ [g]	ΔG_{Exp} [h]
SWM4	-5.90 ± 0.10	0.91	-5.75 ± 0.10	1.06	-5.77 ± 0.02	1.04	-6.81
ACEM	-10.08 ± 0.10	-0.37	-10.49 ± 0.26	-0.78	-10.64 ± 0.06	-0.93	-9.71
BENZ	-1.06 ± 0.12	-0.16	-1.45 ± 0.17	-0.55	-1.36 ± 0.05	-0.46	-0.90
CHEX	2.18 ± 0.04	0.95	2.06 ± 0.10	0.83	2.15 ± 0.04	0.92	1.23
ETHA	2.14 ± 0.07	0.31	2.03 ± 0.07	0.20	2.16 ± 0.02	0.33	1.83
ETOH	-4.33 ± 0.11	0.67	-4.72 ± 0.27	0.28	-4.65 ± 0.05	0.35	-5.00
HEXA	3.00 ± 0.07	0.52	2.51 ± 0.16	0.03	2.54 ± 0.08	0.06	2.48
MEOH	-4.60 ± 0.14	0.50	-4.91 ± 0.09	0.19	-4.90 ± 0.03	0.20	-5.10
MESH	-1.03 ± 0.20	0.17	-0.94 ± 0.16	0.26	-1.04 ± 0.03	0.16	-1.20
PHEN	-6.72 ± 0.14	-0.12	-7.07 ± 0.13	-0.47	-7.07 ± 0.03	-0.47	-6.60
THF	-3.11 ± 0.05	0.36	-3.33 ± 0.14	0.14	-3.12 ± 0.03	0.35	-3.47
RMSE [i]		0.53		0.54		0.58	

[a] The abbreviations used in this column are the residue names defined in the residue topology file (cf. Figure 3.1) [b] ASFE mean values and error estimates of the OMM+SAI approach. [c] Differences between ASFE mean values of the OMM+SAI approach and experimental values [d] ASFE mean values and error estimates of the CHARMM+SAI calculations [e] Differences between ASFE mean values of the CHARMM+SAI approach and experimental values [f] ASFE mean values and error estimates of the caculations by König et al.[13] [g] Differences between ASFE mean values of the calculations by König et al. and experimental values [h] Experimental ASFE values [i] Root-mean-square error compared to the experimental values.

From the RMSE values, it is clear that the overall performance of both SAI approaches and the reference calculations is very similar. It cannot be said that any of them is clearly superior. However, the reference calculations by König et al. are mostly more precise than either SAI implementation. This is expected when comparing to CHARMM+SAI, which has shorter simulation protocols; but as already mentioned, this is not the case for OMM+SAI. This points to factors outside the SAI approach itself (engine-specific, post-processing,...) as the cause of the larger error ranges.

Given the unavailability of meaningful error estimates for most of the compounds, Welch's tests were not performed. Instead, the data are shown again as bar plots with whiskers in

Figure 4.4, to facilitate visual inspection of the differences.

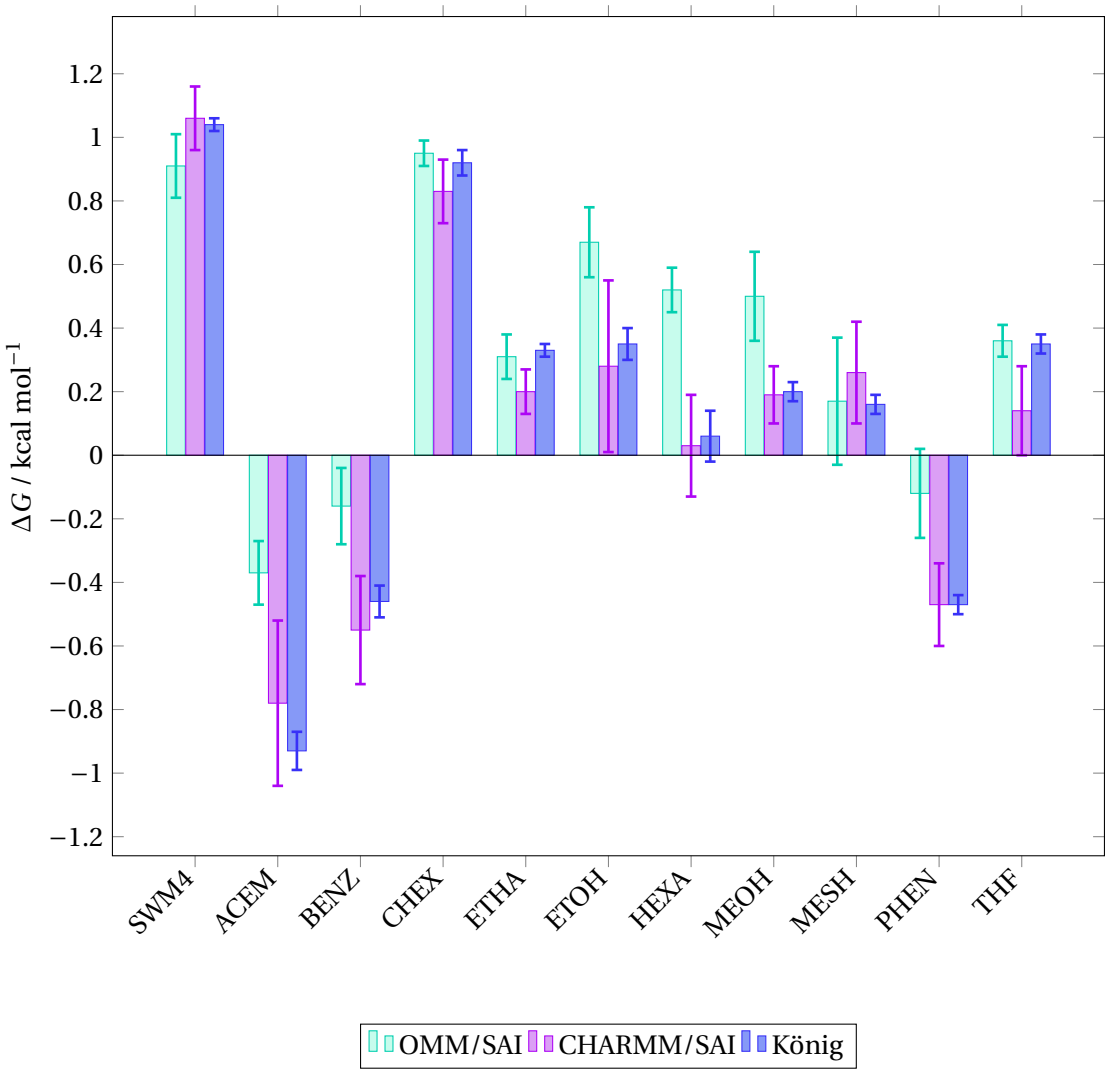

Figure 4.4: Comparison of differences in ΔG values and their error ranges for OMM+SAI (turquoise), CHARMM+SAI (purple) and reference (blue) calculations versus experimental data (zero line).

It can be appreciated that none of the approaches is clearly better than the others. While the differences for OMM+SAI are within experimental error ranges in eight out of the ten cases, they are also off by the largest margin in six of them. CHARMM+SAI is firmly within error ranges in only five cases, but is also in the best agreement in all of these. The results by König et al. are similar to those from CHARMM+SAI, but deviate more in most cases by a small margin. They are within the experimental error ranges for six compounds, but closest only for one.

As already mentioned, for this test set, the polarizable ASFE calculations offer a significant improvement in terms of experimental agreement when compared to those in an additive force field. The study by König et al. has demonstrated this for traditional λ-states. In a previous study by Karwounopoulos et al.,[49] the hydration free energies of most compounds in the FreeSolv[50] database were calculated with the SAI approach as implemented in `transformato`. The calculated ASFEs include our test set, with the exception of water. With regard to these ten compounds, the additive results by König et al. have an RMSE = 0.95 kcal/mol. Those by Karwounopoulos et al. have an RMSE = 1.10 kcal/mol, which is improved to 0.89 kcal/mol when employing long-range correction (see Figure 6.1). In contrast, for the same ten compounds, the approaches using the Drude polarizable FF have RMSEs of 0.51 (König et al.), 0.48 (OMM+SAI) and 0.45 (CHARMM+SAI) kcal/mol compared to experimental values.

4 Considerations for the Drude force field

The use of the Drude force field introduces some specific challenges compared to the additive case. For one, polarizable simulations are less robust. One essential step towards addressing this is the addition of a 'hard wall' constraint on the distance between Drude particle and parent atom.

For the OpenMM simulations, another important factor determining stability seems to be the switching function. Initially, the OpenMM implementation of the force-switching function VFSWitch[51] was used (with the same onset and cutoff distances of 10 and 12 Å, respectively), but these simulations failed to produce stable runs even down to time steps of 0.1 femtoseconds. All failed runs were water box simulations and produced a "particle coordinate is NaN" error.

In an effort to stabilize the simulations using VFSwitch, several strategies were employed, including setting the floating-point type to mixed and double precision, lowering the constraint tolerance, and extending equilibration protocols. Nevertheless, none of these approaches, even in combination, yielded consistently stable simulations across all tested systems. By contrast, employing the default OpenMM potential switching function instead of VFSwitch nearly eliminates stability issues at a time step of 1 fs, while reducing the time step to 0.5 fs prevents even rare crashes.

Further, there remain challenges in achieving an automated workflow comparable to what `transformato` already is capable of for additive force fields.

First, most of the small molecules (six out of eleven) considered in this work contain a heavy atom with anisotropic polarizability. The relevant section (NUMANISO) is present in the water box structure file generated by CHARMM, but is not correctly regenerated for the gas phase. This is because it is ignored by the parmed[52] utility used in `transformato`. As outlined in the section "CHARMM Drude force field", the 1,2 and 1,3 non-bonded exceptions do not apply to the dipole-dipole interactions. Therefore, the omission of the anisotropic effect in the gas phase would affect even the smallest of the tested compounds.

Second, the construction of the transformation path, i.e., the particular order in which LJ interactions are turned off, is not yet applicable to arbitrary structures as it is in the additive case. For this, `transformato` requires an SDF or MOL2 file (which it may generate on the fly from a PDB file). These files provide the bond information from which a molecular graph is constructed for further manipulation. However, these file formats do not include Drude particles. Therefore, they will not be indexed and the transformation path will be faulty. While the same applies to lone pairs, `transformato` already has a method to handle these, which should be extended to Drude particles.

5 Conclusion

The results of this thesis validate the SAI approach for FES with the CHARMM Drude polarizable force field. Well validated reference results using λ-states by König et al. were first recalculated with the PERT module of CHARMM. The results of the CHARMM+SAI approach agree just as well with the reference values as the PERT calculations do. Indeed, agreement is exceptionally good, given that the approach by König et al. employed simulation protocols one to two orders of magnitude longer.

Using OpenMM introduces a significant, albeit small, positive bias compared to the CHARMM calculations. We suspect this to be due to minor differences in the force calculation, particularly the different implementations of the LJ potential-switching function. A more apt comparison might have been the use of the force-switching function VFSwitch both in the CHARMM and OpenMM approaches, but the already discussed stability issues when using VFSwitch with OpenMM made this impractical. Even so, OMM+SAI performs just as well as the other methods when compared to the experimental values.

The `transformato` package already offers a highly automated workflow for setting up SAI intermediate states for FES with additive force fields. It was successfully adapted to offer an enhanced workflow with the Drude polarizable force field for the test set, though challenges remain in achieving a level of automation comparable to the additive case.

The polarizable simulations conducted by König et al. perform significantly better on the test set compared to additive ones. This has also been reproduced by our findings. The obtained values are in better agreement than even the long-range corrected ones obtained with the SAI approach in the additive CHARMM force field by Karwounopoulos et al. This should motivate the application of this approach, and the `transformato`-assisted workflow, to a larger data set.

Bibliography

[1] Ramu Anandakrishnan, Aleksander Drozdetski, Ross C. Walker, and Alexey V. Onufriev. Speed of conformational change: comparing explicit and implicit solvent molecular dynamics simulations. *Biophysical Journal*, **2015**, *108*, 1153–1164.

[2] Niels Hansen and Wilfred F. van Gunsteren. Practical aspects of free-energy calculations: a review. *Journal of Chemical Theory and Computation*, **2014**, *10*, 2632–2647.

[3] Arieh Ben-Naim. *Solvation Thermodynamics*. Plenum Press, New York, **1987**.

[4] Edward King, Erick Aitchison, Han Li, and Ray Luo. Recent developments in free energy calculations for drug discovery. *Frontiers in Molecular Biosciences*, **2021**, *8*.

[5] J. M. Rickman and R. LeSar. Free-energy calculations in materials research. *Annual Review of Materials Research*, **2002**, *32*, 195–217.

[6] E. J. Maginn and J. R. Elliott. Historical perspective and current outlook for molecular dynamics as a chemical engineering tool. *Industrial & Engineering Chemistry Research*, **2010**, *49*, 3059–3078.

[7] Pavel V. Klimovich, Michael R. Shirts, and David L. Mobley. Guidelines for the analysis of free energy calculations. *Journal of Computer-Aided Molecular Design*, **2015**, *29*, 397–411.

[8] Pnina Dauber-Osguthorpe and A. T. Hagler. Biomolecular force fields: Where have we been, where are we now, where do we need to go and how do we get there? *Journal of Computer-Aided Molecular Design*, **2019**, *33*, 133–203.

[9] Marcus Wieder, Markus Fleck, Benedict Braunsfeld, and Stefan Boresch. Alchemical free energy simulations without speed limits. A generic framework to calculate free energy differences independent of the underlying molecular dynamics program. *Journal of Computational Chemistry*, **2022**, *43*, 1151–1160.

[10] Erik C. Neyts and Annemie Bogaerts. Combining molecular dynamics with Monte Carlo simulations: implementations and applications. *Theoretical Chemistry Accounts*, **2012**, *132*, 1320.

[11] Yuchen Li, Yalong Cong, Guoqiang Feng, Susu Zhong, John Z. H. Zhang, et al. The impact of interior dielectric constant and entropic change on HIV-1 complex binding free energy prediction. *Structural Dynamics*, **2018**, *5*, 064101.

[12] Rui Qi, Zhifeng Jing, Chengwen Liu, Jean-Philip Piquemal, Kevin N. Dalby, et al. Elucidating the phosphate binding mode of phosphate-binding protein: the critical effect of buffer solution. *The Journal of Physical Chemistry B*, **2018**, *122*, 6371–6376.

[13] Gerhard König, Frank C. Pickard, Jing Huang, Walter Thiel, Alexander D. MacKerell, et al. A comparison of QM/MM simulations with and without the Drude oscillator model based on hydration free energies of simple solutes. *Molecules*, **2018**, *23*, 2695.

[14] Christian Schröder, Alex Lyons, and Steven W. Rick. Polarizable MD simulations of ionic liquids: How does additional charge transfer change the dynamics? *Physical Chemistry Chemical Physics*, **2020**, *22*, 467–477.

[15] Esther Heid, Stefan Boresch, and Christian Schröder. Polarizable molecular dynamics simulations of ionic liquids: influence of temperature control. *The Journal of Chemical Physics*, **2020**, *152*, 094105.

[16] B. G. Dick and A. W. Overhauser. Theory of the dielectric constants of alkali halide crystals. *Physical Review*, **1958**, *112*, 90–103.

[17] P. J. Mitchell and D. Fincham. Shell model simulations by adiabatic dynamics. *Journal of Physics: Condensed Matter*, **1993**, *5*, 1031.

[18] Johannes Karwounopoulos, Marcus Wieder, and Stefan Boresch. Relative binding free energy calculations with transformato: a molecular dynamics engine-independent tool. *Frontiers in Molecular Biosciences*, **2022**, *9*.

[19] Johannes Karwounopoulos. *Alchemical Free Energy Calculations: Enhancements and Applications*. Ph.D. thesis, University of Vienna, **2023**.

[20] Wonmuk Hwang, Steven L. Austin, Arnaud Blondel, Eric D. Boittier, Stefan Boresch, et al. CHARMM at 45: enhancements in accessibility, functionality, and speed. *The Journal of Physical Chemistry B*, **2024**, *128*, 9976–10042.

[21] Peter Eastman, Raimondas Galvelis, Raúl P. Peláez, Charlles R. A. Abreu, Stephen E. Farr, et al. OpenMM 8: molecular dynamics simulation with machine learning potentials. *The Journal of Physical Chemistry B*, **2024**, *128*, 109–116.

[22] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, et al. CHARMM: the biomolecular simulation program. *Journal of Computational Chemistry*, **2009**, *30*, 1545–1614.

[23] A. D. Jr. MacKerell, D. Bashford, M. Bellott, R. L. Jr. Dunbrack, J. D. Evanseck, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, **1998**, *102*, 3586–3616.

[24] Michael Feig, Alexander D. MacKerell, and Charles L. Brooks. Force field influence on the observation of π-helical protein structures in molecular dynamics simulations. *The Journal of Physical Chemistry B*, **2003**, *107*, 2831–2836.

[25] John G. Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of Chemical Physics*, **1935**, *3*, 300–313.

[26] Robert W. Zwanzig. High-temperature equation of state by a perturbation method. I. Nonpolar gases. *The Journal of Chemical Physics*, **1954**, *22*, 1420–1426.

[27] Charles H Bennett. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, **1976**, *22*, 245–268.

[28] Michael R. Shirts and John D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of Chemical Physics*, **2008**, *129*, 124105.

[29] Philippe H. Hünenberger. *Thermostat Algorithms for Molecular Dynamics Simulations*, pages 105–149. Springer Berlin Heidelberg, Berlin, Heidelberg, **2005**.

[30] Stefan Boresch and Stefan Bruckner. Avoiding the van der Waals endpoint problem using serial atomic insertion. *Journal of Computational Chemistry*, **2011**, *32*, 2449–2458.

[31] Jon Applequist, James R. Carl, and Kwok-Kueng Fung. Atom dipole interaction model for molecular polarizability. Application to polyatomic molecules and determination of atom polarizabilities. *Journal of the American Chemical Society*, **1972**, *94*, 2952–2960.

[32] Steven W. Rick, Steven J. Stuart, and B. J. Berne. Dynamical fluctuating charge force fields: application to liquid water. *The Journal of Chemical Physics*, **1994**, *101*, 6141–6156.

[33] Christopher M. Baker. Polarizable force fields for molecular dynamics simulations of biomolecules. *WIREs Computational Molecular Science*, **2015**, *5*, 241–254.

[34] Pedro E. M. Lopes, Benoit Roux, and Alexander D. MacKerell. Molecular modeling and dynamics studies with explicit inclusion of electronic polarizability. Theory and applications. *Theoretical chemistry accounts*, **2009**, *124*, 11–28.

[35] Guillaume Lamoureux and Benoît Roux. Modeling induced polarization with classical Drude oscillators: theory and molecular dynamics simulation algorithm. *The Journal of Chemical Physics*, **2003**, *119*, 3025–3039.

[36] Justin A. Lemkul, Jing Huang, Benoît Roux, and Alexander D. MacKerell. An empirical polarizable force field based on the classical Drude oscillator model: development history and recent applications. *Chemical Reviews*, **2016**, *116*, 4983–5013.

[37] K. Vanommeslaeghe and A. D. MacKerell. CHARMM additive and polarizable force fields for biophysics and computer-aided drug design. *Biochimica et Biophysica Acta (BBA) - General Subjects*, **2015**, *1850*, 861–871.

[38] Edward Harder, Victor M. Anisimov, Troy Whitfield, Alexander D. MacKerell, and Benoît Roux. Understanding the dielectric properties of liquid amides from a polarizable force field. *The Journal of Physical Chemistry B*, **2008**, *112*, 3509–3521.

[39] Ben Leimkuhler, Emad Noorizadeh, and Oliver Penrose. Comparing the efficiencies of stochastic isothermal molecular dynamics methods. *Journal of Statistical Physics*, **2011**, *143*, 921–942.

[40] Guillaume Lamoureux, Alexander D. MacKerell, and Benoît Roux. A simple polarizable model of water based on classical Drude oscillators. *The Journal of Chemical Physics*, **2003**, *119*, 5185–5197.

[41] Guillaume Lamoureux, Edward Harder, Igor V. Vorobyov, Benoît Roux, and Alexander D. MacKerell. A polarizable model of water for molecular dynamics simulations of biomolecules. *Chemical Physics Letters*, **2006**, *418*, 245–249.

[42] Yalun Yu, Richard M. Venable, Jonathan Thirman, Payal Chatterjee, Anmol Kumar, et al. Drude polarizable lipid force field with explicit treatment of long-range dispersion: parametrization and validation for saturated and monounsaturated zwitterionic lipids. *Journal of Chemical Theory and Computation*, **2023**, *19*, 2590–2605.

[43] CHARMM development project. TPCNTRL documentation, CHARMM version c48b2. https://academiccharmm.org/documentation/version/c48b2/tpcntrl. Retrieved: 2024-11-26.

[44] Peter Eastman, Jason Swails, John D. Chodera, Robert T. McGibbon, Yutong Zhao, et al. OpenMM 7: rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, **2017**, *13*, e1005659.

[45] Markus Fleck, Marcus Wieder, and Stefan Boresch. Dummy atoms in alchemical free energy calculations. *Journal of Chemical Theory and Computation*, **2021**, *17*, 4403–4419.

[46] David L. Mobley. Experimental and calculated small molecule hydration free energies, **2013**. Retrieved from `https://escholarship.org/uc/item/6sd403pz`.

[47] Casey P. Kelly, Christopher J. Cramer, and Donald G. Truhlar. SM6: A density functional theory continuum solvation model for calculating aqueous solvation free energies of neutrals, ions, and solute-water clusters. *Journal of Chemical Theory and Computation*, **2005**, *1*, 1133–1152.

[48] Aleksandr V. Marenich, Casey P. Kelly, Jason D. Thompson, Gregory D. Hawkins, Candee C. Chambers, et al. Minnesota solvation database (MNSOL) version 2012, **2020**.

[49] Johannes Karwounopoulos, Åsmund Kaupang, Marcus Wieder, and Stefan Boresch. Calculations of absolute solvation free energies with transformato—application to the FreeSolv database using the CGenFF force field. *Journal of Chemical Theory and Computation*, **2023**, *19*, 5988–5998.

[50] David L. Mobley and J. Peter Guthrie. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design*, **2014**, *28*, 711–720.

[51] Sunhwan Jo, Taehoon Kim, Vidyashankara G. Iyer, and Wonpil Im. CHARMM-GUI: a web-based graphical user interface for CHARMM. *Journal of Computational Chemistry*, **2008**, *29*, 1859–1865.

[52] Michael R. Shirts, Christoph Klein, Jason M. Swails, Jian Yin, Michael K. Gilson, et al. Lessons learned from comparing molecular dynamics engines on the SAMPL5 dataset. *Journal of Computer-Aided Molecular Design*, **2017**, *31*, 147–161.

6 Appendix

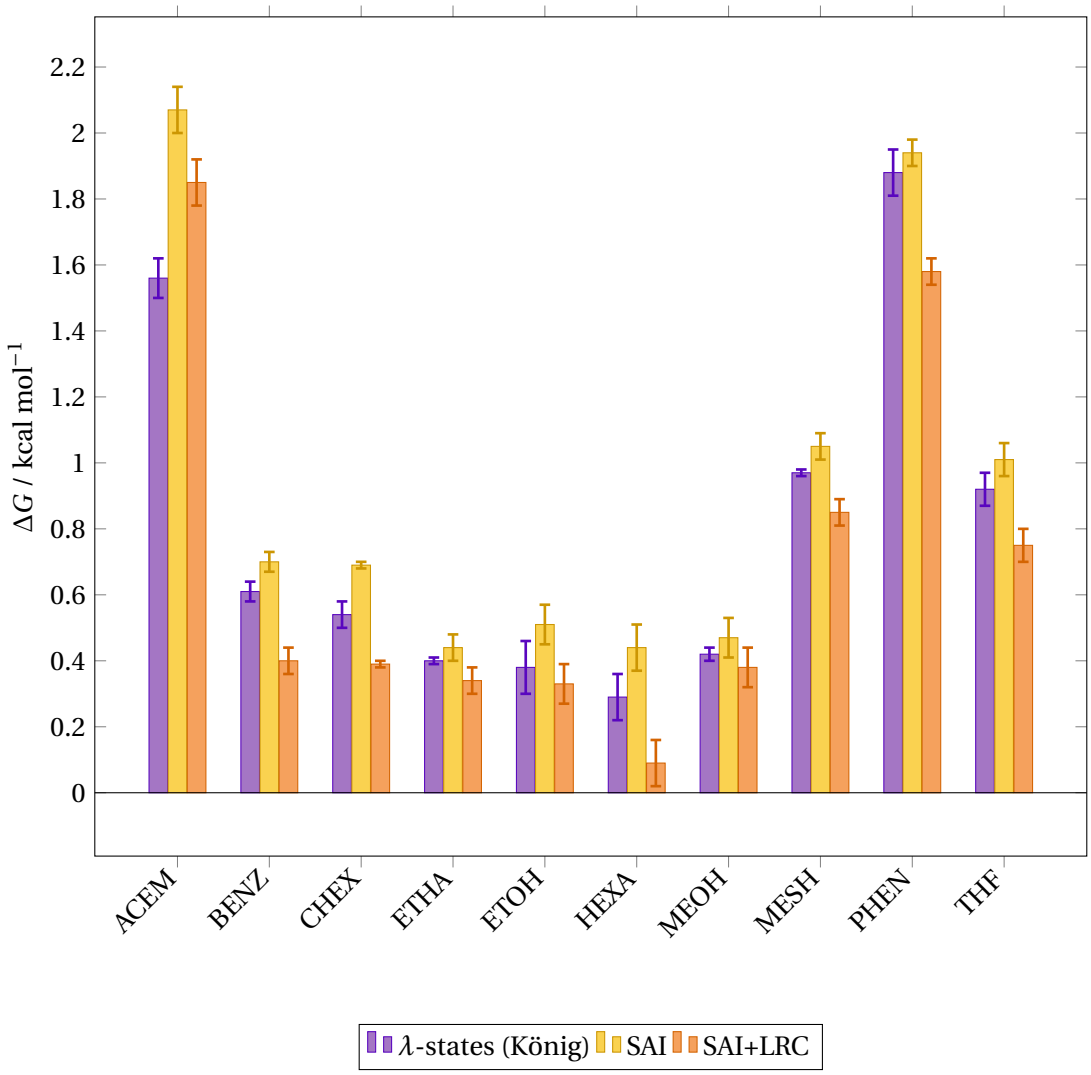

Figure 6.1: Differences in ΔG values and their error ranges of the additive FES compared to experimental data (zero line). The results by König et al. (violet) employed traditional λ-states. Those by Karwounopoulos et al. employed SAI, with long-range-correction (orange), and without (yellow).

Table 6.1: Detailed comparison of simulation parameters for the three approaches. Symbols used: γ: friction coefficients, T: temperatures, with subscripts S for system and D for Drude. τ is the characteristic response time of the barostat. r_{on} is the on-set distance of the tapering function, and r_{off} is the distance where the potential is scaled to zero. κ is the width of the Gaussian screening function in particle-mesh Ewald. In the 'Equilibration steps' and 'Production steps' rows, values in parentheses () refer to the longer protocols employed for hexane and cyclohexane.

Parameter	PERT	CHARMM+SAI	OpenMM+SAI
Temperature control	VV2 integrator with dual Langevin thermostat (TPCONTROL)		DrudeLangevinIntegrator
	$T_S = 298.15$ K, $T_D = 1$ K, $\gamma_S = 5$ ps^{-1}, $\gamma_D = 10$ ps^{-1}		$T_S = 303.15$ K, $T_D = 1$ K, $\gamma_S = 10$ ps^{-1}, $\gamma_D = 200$ ps^{-1}
Pressure control	Anderson barostat (TPCONTROL)		MonteCarloBarostat
	p = 1 atm, $\tau = 0.2$ ps		p = 1 bar, $\tau = 100\Delta$t
LJ treatment	Potential-switching function VSwitch		Native potential-switching function
	$r_{on} = 10$ Å, $r_{off} = 12$ Å		$r_{on} = 10$ Å, $r_{off} = 12$ Å
Electrostatics	Particle-mesh Ewald summation		Particle-mesh Ewald summation
	$\kappa = 0.34^{-1}$, order 6 spline, $24 \times 24 \times 24$ grid		Error tolerance = 0.0005
Time step	Δt = 1 ts		Δt = 0.5 fs
Equilibration steps	Gas phase: 500k (5M) Water box: 100k (200k)	Gas phase: 1M (1M) Water box: 100k (200k)	Both phases: 2.6M[1] (25.1M)[1]
Production steps	Gas phase: 4.5M (45M) Water box: 200k (500k)	Gas phase: 10M (20M) Water box: 1M (2M)	Both phases: 7.8M (75.3M)

[1] Includes the 'ramping protocol': 100k steps at 0.1, 0.2, 0.3, and 0.4 fs each, for a total of 400k steps, to ensure stability. In general, the equilibration steps are not set explicitly before the simulation. Instead, the analysis script simply treats the first quarter of the trajectory as equilibration by discarding it. The total number of steps for the trajectories, including the ramping protocol, was 10.4M (100.4M).