Finite temperature free energy calculations in NWChem: Metadynamics and Umbrella Sampling-WHAM

Raymond Atta-Fynn, Eric J. Bylaska, and Wibe A. de Jong
Environmental Molecular Sciences Laboratory
Pacific Northwest National Laboratory
Richland, Washington 99352

Contents

1 Brief Introduction

Formally, the Helmholtz free energy, F, of the canonical (NVT) ensemble is defined as

$$F = -k_{\mathrm{B}} T \log Z \tag{1}$$

where the *partition function*, Z, is given by

$$Z = \frac{1}{N! h^{3N}} \int d\mathbf{X}\, d\mathbf{P}\, \exp\left[-\beta \mathcal{H}(\mathbf{X}, \mathbf{P})\right] \tag{2}$$

and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)^{\mathrm{T}}$ and $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N)^{\mathrm{T}}$ denotes the $3N$ coordinates and momenta respectively of the N particles in the system, $\beta = \frac{1}{k_{\mathrm{B}} T}$, k_{B} is the Boltzmann constant, h is Planck's constant, and $\mathcal{H}$ is the Hamiltonian of the system. Usually, one is interested in the free energy profile along some order parameter (or reaction coordinate), $\xi(\mathbf{X}, \mathbf{P})$. In such cases, the probability of finding the system at an arbitrary value, ξ_0, of $\xi(\mathbf{X}, \mathbf{P})$ is

$$P(\xi_0) = \frac{\int d\mathbf{X}\, d\mathbf{P}\, \delta\left[\xi(\mathbf{X}, \mathbf{P}) - \xi_0\right] \exp\left[-\beta \mathcal{H}(\mathbf{X}, \mathbf{P})\right]}{\int d\mathbf{X}\, d\mathbf{P}\, \exp\left[-\beta \mathcal{H}(\mathbf{X}, \mathbf{P})\right]} \tag{3}$$

where δ is the Dirac delta function. The corresponding free energy is

$$F(\xi_0) = -k_{\mathrm{B}} T \log P(\xi_0). \tag{4}$$

To make useful predictions and observations about a system, the *free energy difference*, $\Delta F_{\xi_A \to \xi_B}$, between some initial state, ξ_A, and a final state, ξ_B, is required:

$$\Delta F_{\xi_A \to \xi_B} = F(\xi_B) - F(\xi_A) = -k_{\mathrm{B}} T \log\left[\frac{P(\xi_B)}{P(\xi_A)}\right] \tag{5}$$

The obvious, and perhaps nice, feature of Eq. 5 is that we do not have to compute the denominator of Eq. 3 (that is, the unnormalized Z in Eq. 1); we only require the numerator of Eq. 3. However, the numerator must be obtained by integrating over the entire phase space volume. This is certainly not practical in molecular simulations. For practical purposes, we invoke the ergodic hypothesis to compute the probability densities. From a molecular simulation trajectory spanning a sufficiently long time period t, the probability density is computed as:

$$P(\xi_0) = \frac{1}{t} \int_0^t \delta\left[\xi(\mathbf{X}, t') - \xi_0\right] dt' \tag{6}$$

[Note (1)] and the free energy difference is given by Eq. 5. The caveat with the computation of Eq. 5 from Eq. 6 is that both the states $\xi = \xi_A$ and $\xi = \xi_B$ are sufficiently sampled (that is, there is sufficient statistics to generate a probability distribution). The commonly occurring problem in most molecular simulations is that the characteristic thermal energy is not sufficient to allow the system to sample regions of the energy landscape that are separated from low energy regions by large barriers (the barriers are much larger than $k_{\mathrm{B}} T$).

Figure 1 illustrates a scenario where the final state ξ_B cannot be reached from the initial state ξ_A in a molecular simulation.[Note (2)] The obvious reason is the rather high barrier which is can be several orders of magnitude larger than $k_{\mathrm{B}} T$. In a molecular simulation however, such a reaction would proceed at a very slow rate.[Note (3)] In other words, the system will reside at or near the state ξ_A during the

---

[Note (1)] Furthermore in practice, the right side of Eq. 6 is computed via a discretization over time steps or snapshots.

[Note (2)] Unless you let the simulation run forever, which, of course, is not practical.

[Note (3)] According to transition state theory, the reaction rate is proportional to $e^{-\frac{\Delta G^{\ddagger}}{RT}}$, where $\Delta G^{\ddagger}$ is the energy barrier, hence a large $\Delta G^{\ddagger}$ implies a slow reaction rate.
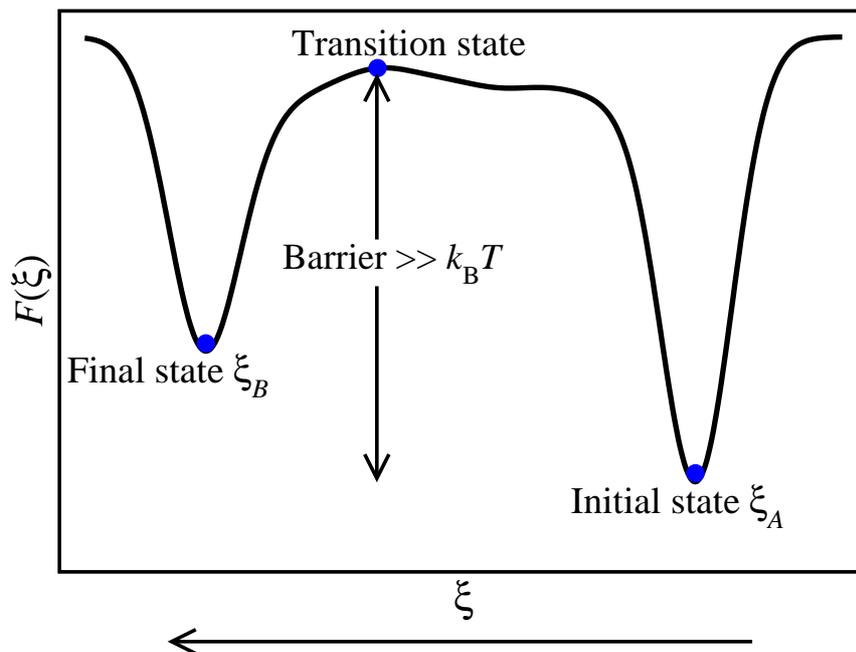
Figure 1: A generic free energy profile (one-dimensional) of an event or a reaction with an energy barrier much larger than $k_\text{B}T$. The blue ball denotes the system under study.

simulation. In literature, this scenario is known as a *rare event*, i.e., the event which corresponds to the system jumping from the energy well ξ_A into the well ξ_B rarely occurs.

In molecular simulations, the simplest and most common way to realistically sample the energy landscape of rare events or slow reactions characterized by ξ is to *bias* the system, i.e., feed some external potential energy, $V(\xi)$, along ξ into the system so that other regions of the energy landscape other than the region around ξ_A can also be sufficiently sampled. There is an avalanche of biased sampling techniques in literature. In this tutorial, we focus on umbrella sampling, potential of mean constrained force, and metadynamics. We will give an overview of each method followed illustrations with simple reactions with NWChem.

2 Metadynamics

2.1 Notations

- $\mathbf{S}(\mathbf{X})$: *Collective variable* space of dimension d which is a *generic* function of the system coordinates $\mathbf{X}$. More explicitly $\mathbf{S}(\mathbf{X}) = \{S_1(\mathbf{X}), S_2(\mathbf{X}), \ldots, S_d(\mathbf{X})\}$, where S_i, $i = 1, 2, \ldots, d$, are independent order parameters that are "collectively" capable of describing the event or reaction of interest.[Note (4)]

- σ_i: Width of the Gaussian associated with $s_i(\mathbf{X})$. σ_i has the units of S_i for each i.

- h_0: Effective Gaussian height. It is assumed in NWChem that $h_0 = \left(\prod_{i=1}^{d} h_i\right)^{\frac{1}{d}}$, where h_i is the height of the Gaussian associated with $s_i(\mathbf{X})$. h_0 has the units of energy.

- τ: Constant time interval between the addition of two successive Gaussians.

- V: History potential. $V \equiv V(\mathbf{S}(\mathbf{X}), t)$

- F: Free energy. $F \equiv F(\mathbf{S}(\mathbf{X}))$

2.2 A simplified perspective of metadynamics

Metadynamics is a non-equilibrium molecular dynamics method which accelerates the sampling of the multidimensional free energy surfaces of chemical reactions on a short-to-moderate simulation time scale [2, 1, 3, 4]. The accelerated sampling is achieved by adding an external time-dependent bias potential which is a function of the collective variables to the Hamiltonian of the system. The bias potential discourages the system from revisiting previous sampled regions of the energy landscape by depositing repulsive Gaussian potentials centered on the location of collective variables on the energy landscape. The accumulation of the bias potential in low energy wells allows the system to surmount energy barriers much more quickly compared to standard molecular dynamics, and hence the system is has the freedom to explore other regions (particularly high energy regions) of the energy landscape.

A simple way to understand how metadynamics works is to consider the one-dimensional free energy landscape in Figure 2(a) as a network of two connected empty wells with different widths and depths. In Figure 2(a), the system ("light" blue ball) is initially lying at the bottom of the deepest well. Our goal is to use water from an external source to "quickly" drive the ball from the initial state to the final state. Within the well-ball-water analogy, the basic principles of metadynamics are as follows:
(i) Locate the position of the ball.
(ii) Pour water from a "small Gaussian-shaped" cup into well such that the highest point of the cup directly faces the ball's position.
(iii) The presence of water in the well will cause the ball's position to rise up. Wait for a time period of τ so that the ball can settle down in it's new position and then repeat step (i) and (ii) above.

In this scenario, the water is acting as a bias which modifies the ball's free energy $F(s)$. Thus at any given time t, the total amount of water is a measure of the total external bias, $V(s, t)$, introduced on the energy landscape along the collective variable s. Obviously the size of V grows in time. In metadynamics, V is referred to as the *external history* potential.[Note (5)] Therefore at time t, the total energy of the ball will be $F(s) + V(s, t)$ [because the initial energy at the bottom of the well was $F(\mathbf{S})$ but the external energy, $V(s, t)$, supplied by the water caused it to rise to that location].

[Note (4)] In a strict sense a *collective variable* is not necessarily the same as a *reaction coordinate*. Specifically, all reaction coordinates are collective variables but the converse is not true. Collective variables are more generic. However, if the collective variable can distinguish between the reactant, product, and transition regimes, and in addition capture the associated reaction kinetics, then it may be regarded as a reaction coordinate.

[Note (5)] "History" here refers to that fact that at any time t, $V(s, t)$ is determined by all *prior* deposited Gaussians.

As more and more cups of water targeting the ball's position are poured onto the energy landscape, the ball will slowly escape the first well, make a transition over the barrier, and enter the second well (final state). If we continue to pour more water into the second well, both wells will be filled, and eventually the ball will freely diffuse along the surface of constant potential; by this time the energy landscape is flooded (think of the wells as being fully submerged in water and the ball rolling on the water surface). Ultimately (that is, in the long time limit) the ball's total energy, $F(s) + V(s,t) = C$, where C is a constant, and therefore, $F(s) = -V(s,t) + C$.
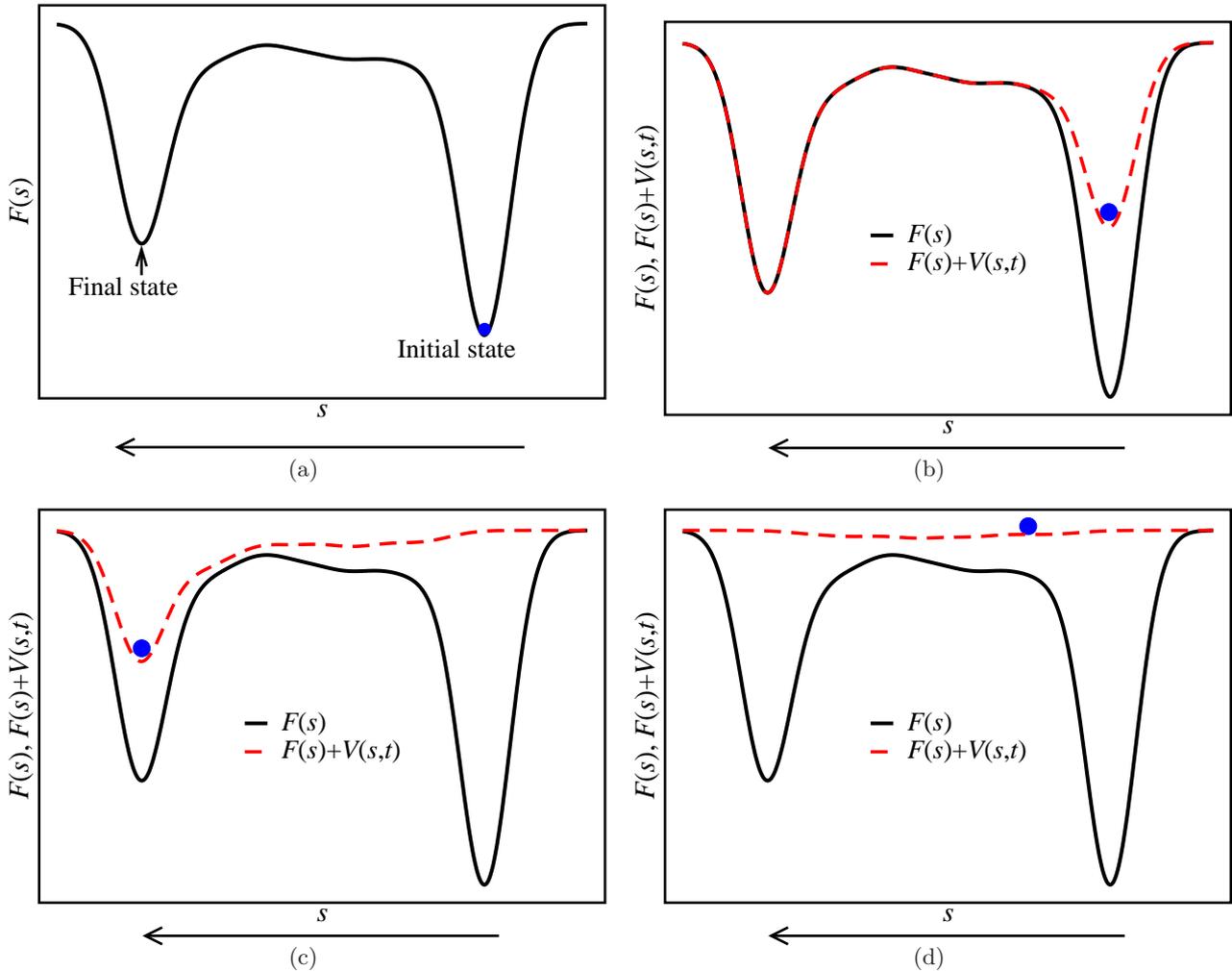
Figure 2: Depiction of the progressive modification of the one-dimensional free energy landscape in 2(a) by external Gaussian bias potentials (2(b)-2(d)). The blue ball denotes the location of collective variable, s, on the free energy surface. A Gaussian potential centered on the current position of the ball is added every τ time unit. $F(s)$ is the original free energy profile (solid black line). $F(s) + V(s,t)$ is the biased free energy profile (broken red line) corresponding to a total of (b) 65 Gaussians (total elapsed time=60τ) (c) 220 Gaussians (total elapsed time=220τ) (d) 300 Gaussians (total elapsed time=300τ).

Figures 2(b)-2(d) illustrates how metadynamics can be used to accelerate the sampling of the free energy corresponding to the scenario in Figure 2(a). In Figure 2(b) a total 60 Gaussian cups of water are used to is pushed from the bottom to near the midpoint of the well (broken red lines). By the time a total of 220 Gaussian cups of water is poured on the landscape (Figure 2(c)), the ball has enough energy to roll over the barrier and enter the second well. The ball then begins its departure from the second well. Finally after the addition 220 Gaussians (Figure 2(d)), both wells are fully filled and the free energy

$F(s)$ is nearly flattened by the bias potential $V(s, t)$, that is,

$$F(s) + V(s, t) \approx constant$$

and therefore the free energy can be recovered from a constant shift in $-V(s, t)$, that is,

$$F(s) \approx -V(s, t) + constant$$

In essence, Figure 2(d) indicates that the ball can move freely on the water surface–this is known *barrier-less diffusion*. In general, barrier-less diffusion of the collective variable(s) is the condition for terminating a metadynamics-driven free energy simulation. A short movie describing this metadynamics process can be found on the internet at http://www.youtube.com/watch?v=CtIrLkx6aNo.

The most attractive feature of metadynamics is that it is fairly easy to implement in a computer code. In sections to follow, we will discuss two variants of *direct metadynamics*, namely, *standard metadynamics* and *well-tempered metadynamics*.[Note (6)]

2.3   Technical aspects of Metadynamics

2.3.1   Metadynamics Equations

Having explained how metadynamics works using the simple one-dimensional free energy surface in the previous section, we are now ready to write down the metadynamics equations. Unless stated otherwise, we will assume the notations in Section 2.1.

Suppose that *prior* to any time t during a metadynamics simulations N Gaussians centered on the d-dimensional collective variable $\mathbf{S}^{t_g} = \{S_1^{t_g}, S_2^{t_g}, \ldots, S_d^{t_g}\}$ are deposited on the energy landscape described by $\mathbf{S}(\mathbf{X})$ at times $t_g = \tau, 2\tau, 3\tau, \ldots, N\tau$. Then the history potential, V, at an arbitrary value $\mathbf{s} = \{s_1, s_2, \ldots, s_d\}$ of $\mathbf{S}(\mathbf{X})$ at time t is given by

$$V(\mathbf{s}, t) = \sum_{t_g = \tau, 2\tau, \ldots, N\tau} h_0 \exp\left[-\sum_{i=1}^{d} \frac{\left(s_i - S_i^{t_g}\right)}{2\sigma_i^2} \right] \tag{7}$$

In the long time limit $(t \to \infty)$, $F(\mathbf{s}) + V(\mathbf{s}, t)$ is constant and therefore an estimate for the free energy is

$$F(\mathbf{s}) = -\lim_{t \to \infty} V(\mathbf{s}, t) \tag{8}$$

Equations 7 and 8 constitute *standard metadynamics*.

One issue with standard metadynamics is that often the bias, V, does not converge smoothly (it fluctuates about some average value). Another, and perhaps very crucial, issue with standard metadynamics is when to terminate the simulation. There is the of risk of driving the system outside the domain of interest into unwanted higher energy regions if the simulation is not terminated at the appropriate time. To alleviate this problem, *well-tempered metadynamics* [4, 5] was introduced. In well-tempered metadynamics, the smooth convergence of V is ensured by gradually decreasing the the height of the Gaussian (so that in the $t \to \infty$ regime the Gaussians will vanish). The amount of reduction is depends on two factors: (i) The amount of bias, V, that is already present (ii) A fictitious temperature ΔT, which, loosely speaking, limits the free energy exploration to the range $[k_\mathrm{B}T, k_\mathrm{B}(T + \Delta T)]$. The well-tempered metadynamics equations are as follows:

$$V(\mathbf{s}, t) = \sum_{t_g = \tau, 2\tau, \ldots, N\tau} h(t) \exp\left[-\sum_{i=1}^{d} \frac{\left(s_i - S_i^{t_g}\right)}{2\sigma_i^2} \right] \tag{9}$$

[Note (6)] "Direct" here implies that the bias potential acts directly on the collective variable(s). Currently only direct metadynamics is implemented in NWChem. There is, for example, Lagrangian metadynamics, in which the collective variable, s, is coupled to an auxiliary particle, $\tilde{s}$, of mass M via a harmonic spring of spring constant k. In fact direct metadynamics corresponds to the $k \to \infty$ and $M \to 0$ limits of Lagrangian metadynamics. See Ref. [3] and [4] for the different flavors of metadynamics.

where the time dependent height $h(t)$ is given by

$$h(t) = h_0 \exp\left[-\frac{V\left(\mathbf{S}^{t_g}, t\right)}{k_{\mathrm{B}} \Delta T}\right] \tag{10}$$

In this case the free energy is given by

$$F(\mathbf{s}) = -\left(1 + \frac{T}{\Delta T}\right) \lim_{t \to \infty} V(\mathbf{s}, t) \tag{11}$$

From Eq. 10, it is easy to see that in the limit $\Delta T \to \infty$, $h(t) = h_0$, and therefore Eq. 9 reduces to Eqs. 7. Similarly $\Delta T \to \infty$ reduces Eq. 11 to Eq. 8. Thus in the limit as $\Delta T \to \infty$ well-tempered metadynamics reduces to standard metadynamics. When $\Delta T \to 0$, $h(t) = 0$, and hence the simulation reduces to regular molecular dynamics (because according Eq. 9, the bias, $V=0$ for all values of t). A finite positive value of ΔT corresponds to well-tempered metadynamics. There is no hard-and-fast rule for choosing ΔT. However the rule of thumb is that if an estimate of highest energy barrier, E_b, is known *a priori*, then ΔT may be determined from $k_{\mathrm{B}} \Delta T \sim E_b$.

2.3.2 Collective variables

Choosing the right set of collective variables is the most important aspect of a metadynamics simulation; the success or failure of the reaction mechanism or event one wishes to simulate depends heavily on the choice of the collective variables. The collective variables should be able to clearly distinguish between the initial state, transition state, and final state. Usually, *a priori* knowledge of the mechanism under study is helpful in choosing the collective variables. It is also important to minimize the dimension, d, of the collective variable space as much as possible. In Most metadynamics simulations $1 \leq d \leq 3$. It is important to note that the cost of the simulation grows enormously with d [Note (7)] and convergence may also be problematic. The following collective variables are implemented in NWChem [8]:

1. Bond distance

2. Bond angle

3. Dihedral angle

4. Coordination number

5. Average distance of an adatom (or impurity) from a surface layer along the surface normal

2.3.3 Other issues: Choice of Gaussian height and widths

It has been shown that error ϵ in the free energy estimate from metadynamics is related to h_0, σ, and τ by $\epsilon \propto \sqrt{h_0 \sigma / \tau}$. If Gaussian of large height and width are deposited at a fast rate (large τ_G), the free energy surface will be explored at a fast pace, but the estimated profile will be affected by large errors. Conversely, if small Gaussian hills are deposited at a relatively slow rate, the free energy surface will be sampled more accurately but the computational burden will grow immensely. In fact, σ controls the resolution of the estimated free energy surface; if σ is large, the resolution is poor; if σ is small the resolution good. Finding optimal choice of $\{h_0, \sigma, \tau\}$ is not easy but our experience have shown that for *ab initio* metadynamics simulations with 5-10 kcal/mol barriers, h_0=0.0001-0.0003 a.u. ($\approx$0.06-0.19 kcal/mol); σ=0.05-0.1 (here σ is in the units of the collective variable) and τ=250-500 a.u. ($\approx$6.1-12.1 fs) works fairly well. For very large barriers (e.g. 100 kcal/mol) larger value of h_0 may be used. These are mere estimates and it is up to the user calibrate $\{h_0, \sigma, \tau\}$ to obtain reliable free energies.

[Note (7)] The computational cost is proportional to $1/\sigma^d$

2.4 Practical examples using NWChem

2.4.1 Example 1: Dissociation barrier of $CH_3Cl \rightarrow CH_3^+ + Cl^-$

We now illustrate how to estimate the barrier of the above reaction at 300 K from Car-Parrinello meta-dynamics simulations. The collective variable is the C-Cl bond distance, h_0=0.001 a.u., σ=0.1 a.u., and τ=500 a.u. The full input file is provided in **Appendix A**.

2.4.2 Results of Example 1

In Figure 3 the plot of the C-Cl distance versus the number of Gaussians is depicted. Complete dissociation occurs after the addition of 1100 Gaussians, with bond formation occurring briefly after the addition of 1250 Gaussians. Beyond this point, the collective variable diffuses over the energy landscape, that is, bonding and dissociation occur back-and-forth, and therefore the reaction is considered complete. If one is interested in the free energy error analysis, then the simulation must be allowed to run for long times, followed by block averaging (see Ref. [3] for the details of the error analysis). Note that no attempt was made to tune the simulation parameters (box size, kinetic energy cut-off, metadynamics parameters, etc.). In a real problem one may be have carefully calibrate the input parameters to obtain the best estimate for the free energies. In Figure 4, the metadynamics-derived free energy (reconstructed with
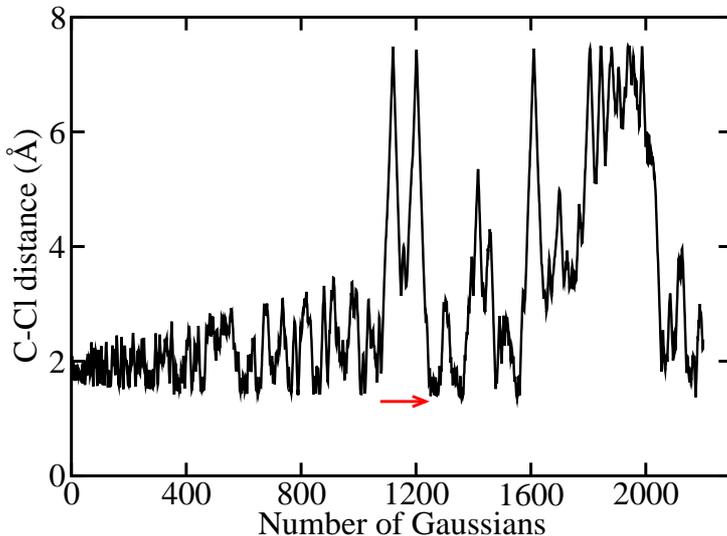
Figure 3: Evolution of the C-Cl bond distance as a function of the number of Gaussians.

1250 Gaussians) is compared to the O K curve from single point energy calculations. There is good agreement near and at the the minimum but a non-negligible discrepancy can be seen at $r >3$ Å and beyond due thermal and dynamical effects. The metadynamics dissociation barrier of 81 kcal/mol agrees well with the experimental value of 83.5 kcal/mol [9].

2.4.3 Example 2: $Cl^- + CH_3Cl \rightleftharpoons ClCH_3 + Cl^-$ S_N2 exchange reaction barrier

To model the barrier to the above exchange reaction, we require a collective variable of space of dimension 2 (since we need to describe the two C-Cl interactions independently). There are two choices of the collective variable: (i) the two C-Cl distances or (ii) the coordination number of C with respect to each Cl. Here we employ option (ii) since the reaction proceeds much quicker. The coordination number is defined as a Fermi-like function:

$$N(C - Cl) = \frac{1}{1 + \exp\left[n\left(r_{C-Cl} - r_0\right)\right]}$$
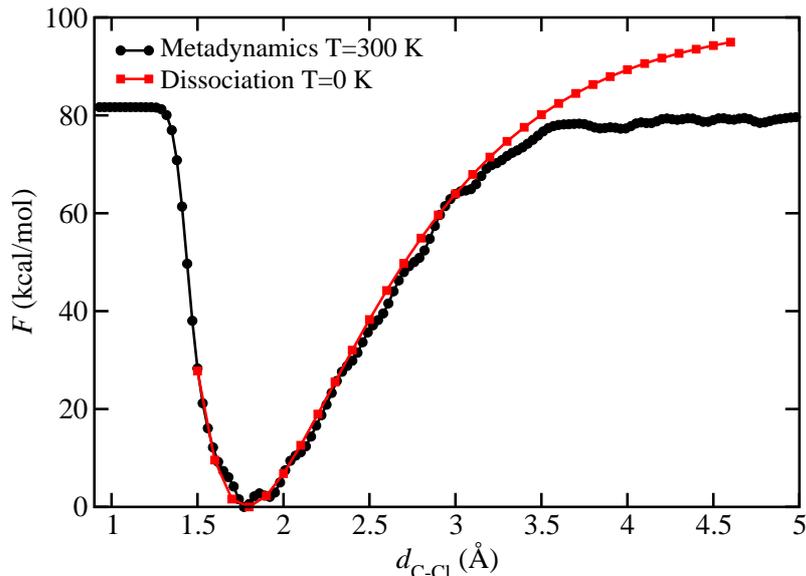
Figure 4: Comparison of 0 K dissociation curve to the 300 K metadynamics free energy curves of the reaction $CH_3Cl(g) \rightarrow CH_3(g)^+ + Cl(g)^-$.

where r_{C-Cl} is the C-Cl distance, r_0 denotes the C-Cl cut-off radius, and n measures the sharpness of the Fermi function.[Note (8)] We essentially want to drive the initial state $(n(C - Cl_1), n(C - Cl_2)) \approx (1,0)$ to the final state $(N(C - Cl_1), N(C - Cl_2)) \approx (0,1)$, where $N(A - B)$ is the coordination number of A with respect to B. Cl_1 denotes the Cl initially attached to C and Cl_2 denotes the initial attacking chloride. The metadynamics parameters were r_0=4.157397 a.u. (2.2 Å), κ=5.29177 a.u. (10 Å^{-1}), h_0=0.00025 a.u., σ=0.1 a.u., and τ=500 a.u. The full input file is provided in **Appendix B**.

2.4.4   Results of Example 2

Since the coordination number collective variable depends on the C-Cl distances, it is useful to monitor the reaction by tracking the variations in the distances as the Gaussian potentials are injected on to the energy landscape. In Figure 5, we depict the variations in the bond distances with respect to the Gaussian potentials. After the addition of 60 Gaussians, the reactant well is filled and the first exchange barrier crossing (BC) occurs. At the first BC, the C-Cl$_1$ and C-Cl$_2$ distances are 2.3 Å each, in agreement with theoretical estimates of 2.3-2.4 Å [10]. After the addition of a total of 133 Gaussians, the product well is filled and the second exchange BC occurs.

To obtain an accurate estimate of the exchange barrier, multiple exchanges are allowed to occur. We thus reconstructed the free energy surface with 330 Gaussians. This is shown in Figure 6. The two wells show some asymmetry with respect the transition point but due inadequate sampling; if the sampling is carried out for long periods a fully symmetric surface will be obtained. The exchange barrier from the reactant well ($N(C - Cl_1)$) to the transition state is computed to be 8.5 kcal/mol; the barrier from the product to the transition state is 9.2 kcal/mol. Both estimates are in good agreement the experimental estimate of 8.7 kcal/mol [11].

[Note (8)] Here, n, which has units of inverse length, is analogous to $\frac{1}{k_B T}$ in the Fermi-Dirac function. See the Pseudopotential Plane Wave section of the NWChem manual for more information on the coordination number collective variable.
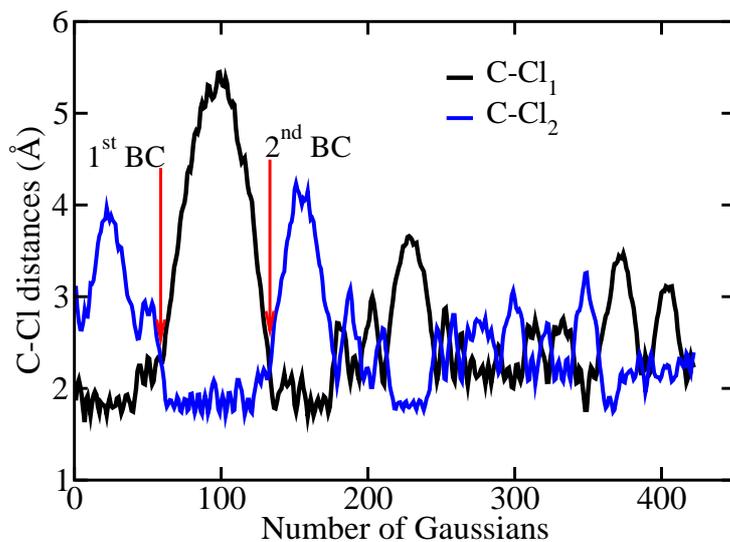
Figure 5: Evolution of the C-Cl distances as a function of the number of Gaussians. Cl_1 is the Cl initially attached to C and Cl_2 is the initial attacking chloride (that is, Cl^-). BC denotes "barrier crossing."
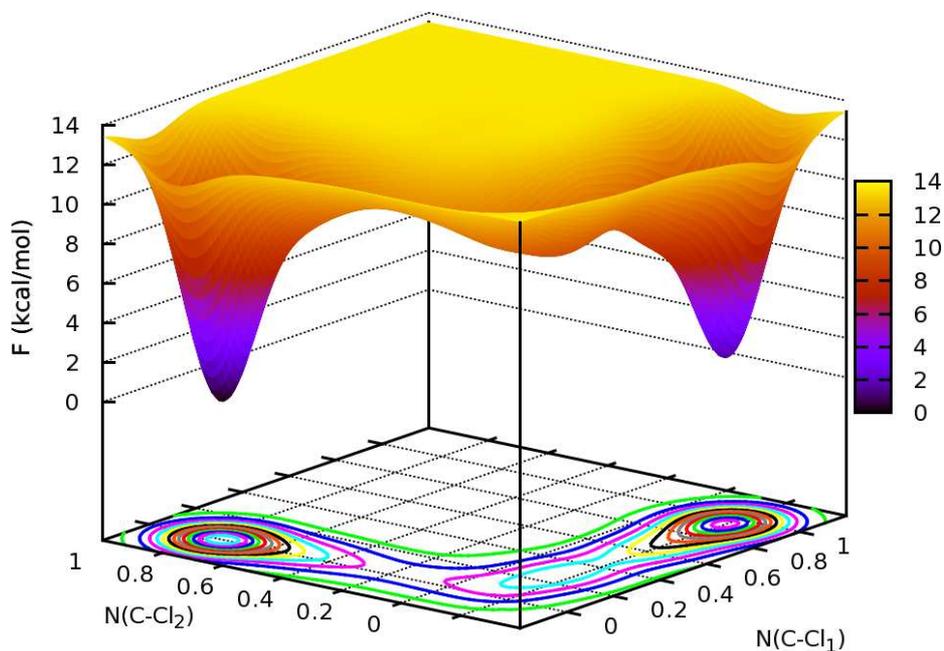
Figure 6: Free energy surface of the S_N2 exchange gas phase reaction $Cl^- + CH_3Cl \rightleftharpoons ClCH_3 + Cl^-$ from metadynamics using the coordination numbers of C with respect to each Cl atom as the collective variables. The free energy was generated using the first 330 Gaussians.

3 Umbrella sampling-WHAM

3.1 Notations

- The terms "window" and "simulation" can be used interchangeably
- N_w: Number of windows used in the umbrella sampling calculations
- ξ: Reaction coordinate
- ξ_i: Reaction coordinate characterizing the center of window i
- K_i: Spring constant describing window i
- $V_i(\xi)$: Biasing potential of the ith window
- $\rho(\xi)$: Optimal unbiased probability of finding the system at ξ
- $\rho_i^{(b)}(\xi)$: Biased probability of finding the system at ξ in window i
- n_i: Number of *snapshots* used to construct $\rho_i^{(b)}$ in simulation i [Note (9)]
- F_i: Free energy constant of window i
- β: $\frac{1}{k_B T}$

3.2 Short introduction

Simply speaking, umbrella sampling is piecewise bias sampling of the energy landscape. Each piece is defined by a biasing potential known as "umbrella" potential. The biasing potential, $V_i(\xi)$, serves to confine the variations in the reaction coordinate, ξ, around some fixed value ξ_i, and therefore enables a more efficient sampling in the region around ξ_i [6, 7]. The region sampled in the presence of the biasing potential is called a **window**. After windows are sampled, their respective probability distributions (these are the biased probability distributions) must be combined to yield a single unbiased optimal probability distribution. Once the unbiased probability distribution is known, the free energy can be estimated using Eq. 4 (or Eq. 5).

The weighted histogram analysis method (WHAM) is a scheme for "stitching" the biased probability distributions from the umbrella sampling together to yield a *single unbiased probability distribution*. We must mention that the probability distributions of neighboring windows usually overlap.

3.3 The WHAM equations

The WHAM equations for a one dimensional reaction coordinate are [6]:

$$\rho(\xi) = \frac{\displaystyle\sum_{i=1}^{N_w} n_i \rho_i^{(b)}(\xi)}{\displaystyle\sum_{j=1}^{N_w} n_j e^{-\beta(V_j(\xi) - F_j)}} \tag{12}$$

$$e^{-\beta F_i} = \int e^{-\beta V_i(\xi)} \rho(\xi) d\xi \tag{13}$$

Equations 12 and 13 must be solved self-consistently since the F_is are unknown. See Refs. [6] and [7] for the derivation of the WHAM equations. Extension to multidimensional cases is straightforward (See Ref. [6]).

[Note (9)] By *snapshots*, we mean the total number of times the geometry is saved in simulation i. It should also be pointed out that n_i can be different for each window since certain windows require longer sampling times.

3.4 Using the WHAM equations in computer simulations

1. Perform N_w simulations with the biasing (window) potentials, $V_i(\xi)$, centered on successive values, ξ_i, of the reaction coordinate, where $i = 1, 2, \cdots, N_w$. The most commonly used window potential is the harmonic function $V_i(\xi) = K_i(\xi - \xi_i)^2$.[Note (10)]

2. For each simulation i, compute the corresponding biased probability distribution $\rho_i^{(b)}(\xi)$ (normalized histogram). To construct the histograms, we consider a range, $[\xi_{\min}, \xi_{\max}]$, which contains every ξ from each simulation. We choose a sufficiently small bin width of $\Delta\xi$. The total number of bins, M, is therefore $M = 1 + \texttt{int}\left(\frac{\xi_{\max} - \xi_{\min}}{\Delta\xi}\right)$, where $\texttt{int}(f)$ denotes the integer portion of the real number f. In discussion to follow, $\rho_i^{(b)}(\xi_j)$ denotes the **biased probability** of finding the system at $\xi = \xi_j$ in window i.

3. The WHAM equations 12 and 13 must be solved iteratively. It is convenient to use the histogram parameters of the biased probability distributions above, that is, $\xi_{\min}$, $\xi_{\max}$, $\Delta\xi$, and M, to generate the **optimal unbiased probability** distribution $\rho(\xi)$. It is also convenient to set $e^{-\beta F_i} = \mu_i$. The discretized versions of equations 12 and 13 used in the computer code for calculating the PMF are:

$$\rho(\xi_j) = \frac{\sum_{i=1}^{N_w} n_i \rho_i^{(b)}(\xi_j)}{\sum_{i=1}^{N_w} \frac{n_i}{\mu_i} e^{-\beta V_i(\xi_j)}} \tag{14}$$

$$\mu_k = \sum_{i=1}^{M} \rho(\xi_i) e^{-\beta V_k(\xi_i)} \tag{15}$$

In equation 14, $j = 1, 2, \cdots, M$ (j labels the histogram bins) and in equation 15, $k = 1, 2, \cdots, N_w$ (k labels the simulation/window). Starting with an initial guess of $\mu_k^{\text{old}} = 1$ for each k, each $\rho(\xi_j)$ can be estimated from equation 14. The estimates for $\rho(\xi_j)$, is then used to obtain new values of μ_k^{new} using equation 15. If self-consistency is not reached, then replace μ_i^{old} with μ_i^{new} and continue the iteration procedure. The following criterion is used for self-consistency:

$$\sum_{j=1}^{N_w} \left(1 - \frac{\mu_j^{\text{new}}}{\mu_j^{\text{old}}}\right)^2 < \epsilon$$

where the tolerance, ϵ, is typically set to 10^{-7}. Please note that for cases where $\rho(\xi_j)=0$ in equation 14, we reset it a positive number practically close to zero (e.g. 1^{-15}) to ensure that $\ln(\rho(\xi_j))$ is finite.

4. Finally, the unbiased potential of mean force (PMF), A, is computed as:

$$A(\xi_j) = -\frac{1}{\beta}\ln(\rho(\xi_j)) + A_0$$

where $j = 1, 2, \cdots, M$ and A_0 is an arbitrary constant. We use $A_0 = \max\{\frac{1}{\beta}\ln(\rho(\xi_j))\}|_{j=1}^{M}$ so that the lowest free energy is zero.

[Note (10)]The prefactor $\frac{1}{2}$ on the rightside is omitted in NWChem.

3.5 Practical example using NWChem

In this section we illustrate how to compute the PMF for the reaction

$$CH_3Cl(g) \rightarrow CH_3(g)^+ + Cl(g)^-$$

using trajectories from Car-Parrinello molecular dynamics (CPMD) simulations at 300 K. The reaction coordinate, ξ, is the C-Cl bond distance. The biasing potential was $V_i(\xi) = K_i(\xi - \xi_i)^2$. The total number of windows N_w=29 and the window centers are 1.7 Å to 4.5 Å in increments of 0.1 Å, that is, ξ_1=1.7 Å, ξ_2=1.8 Å,......, ξ_{29}=4.5 Å. The spring constant K_i was set to 0.3 a.u for each i. (0.3 Ha/Bohr^2). As an example, the NWChem input file for the ξ_1=1.7 Å window is provided in **Appendix C**.

3.6 Results

In Figure 7, the biased probability distribution of each window is depicted. The overlap between neighboring windows can be seen. However, each window is fairly localized around its center, implying that the spring is tight (large K_i). However because the window centers are sufficiently close and so this is not a problem. In a costly simulation, where such a large number of windows cannot be used, the spacing between the windows and spring constant of each window must be carefully to ensure that there is sufficient overlap between neighboring windows while at the same time minimizing the computational cost.
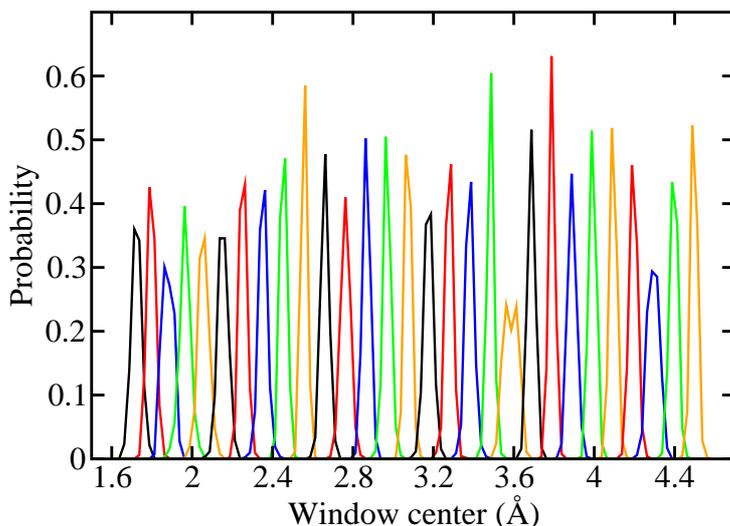
Figure 7: Biased probability distribution of each window. Different coloring schemes are used to distinguish between overlapping window.

Using the Fortran 77 code in **Appendix D**, we generated the WHAM free energy curve and compared to the 0 K dissociation. This is depicted in Figure 8. The WHAM dissociation energy of 80 kcal/mol is in good agreement with the experimental value of 83.5 kcal/mol but shows some deviation the curve generated from the single point energies in the large separation region.
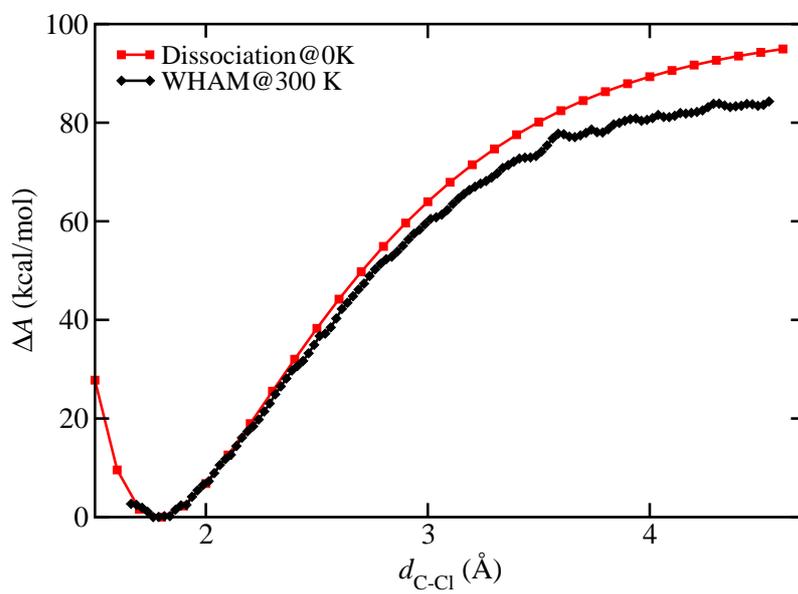
Figure 8: Comparison of 0 K dissociation curve to the 300 K umbrella-sampling WHAM free energy curves of the reaction $CH_3Cl(g) \rightarrow CH_3(g)^+ + Cl(g)^-$.

4 References

[1] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **99**, 12562 (2002).

[2] C. Micheletti, A. Laio, and M. Parrinello, Phys. Rev. Lett. **92**, 170601 (2004).

[3] A. Laio and F. L. Gervasio, Rep. Prog. Phys. **71**, 126601 (2008).

[4] A. Barducci, M. Bonomi, and M. Parrinello, Wiley Interdisciplinary Reviews: Computational Molecular Science **1** 826843 (2011).

[5] A. Barducci, G. Bussi, and M. Parrinello, Phys. Rev. Lett. **100**, 020603 (2008).

[6] B. Roux, Com. Phys. Comm. **91**, 275 (1995).

[7] S. Kumar, D. Bouzida, R. H. Swendsen, P. A. Kollman, and J. M. Rosenberg, J. Comp. Chem. **13**, 1011 (1992).

[8] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, Comp. Phys. Comm. **181**, 1477 (2010).

[9] R.C. West (Ed.), Handbook of Chemistry and Physics, 65th ed., Chemical Rubber Company, Boca Raton, FL, 1984.

[10] B. Ensing, E. J. Meijer, P. E. Blöchl, and E. J. Baerends, J. Phys. Chem. A **105**, 3300 (2001).

[11] R. C. Dougherty, J. Dalton, and J. D. Roberts, Org. Mass Spectrosc. **8**, 77 (1973).

Appendix A: Metadynamics input file for $CH_3Cl \rightarrow CH_3^+ + Cl^-$

```
title "CH3Cl"
echo
start CH3Cl
scratch_dir      ./perm_meta
permanent_dir    ./perm_meta
memory 1600 mb

charge 0

geometry units angstrom nocenter noautosym noautoz print xyz
C     0.000000    0.000000    0.000000
Cl    1.799895    0.017342   -0.008727
H    -0.323446   -0.918592    0.460310    mass 2.0
H    -0.340365    0.851532    0.565115    mass 2.0
H    -0.340579    0.057144   -1.020309    mass 2.0
end

nwpw
   simulation_cell units angstroms
     boundary_conditions periodic
     SC 15.0
   end
   cutoff 35.0
   mult 1
   lmbfgs
   mapping 2
   xc pbe96
   lcao_skip
end

nwpw
  rotation off
end

#Optimize the wave function
task pspw energy

#CPMD parameters
nwpw
   car-parrinello
     Nose-Hoover 1200.0 300.0 1200.0 300.0
     time_step   5.0
     fake_mass   600.0
     scaling     1.0 1.0
   end
end

#Briefly equilibrate
nwpw
   car-parrinello
     loop                10 100
     fei                 equi.fei
     xyz_filename        equi.xyz
     ion_motion_filename  equi.ion_motion
```

```
            emotion_filename       equi.emotion
       end
   end
   task pspw car-parrinello

   #Metadynamics begins here
   nwpw
       #This is not necessary but I prefer to define separate metadynamics output files
       #Otherwise the metadynamics output will be appended to equi.*
       car-parrinello
           loop                    10 2000 #inner-loop outer-loop
           fei                     meta.fei
           xyz_filename            meta.xyz
           ion_motion_filename     meta.ion_motion
           emotion_filename        meta.emotion
       end
       #Metadynamics block
       metadynamics
           print_shift  1              #prints the bias potential every 1000 time steps
           update       10             #adds a Gaussian every 10*(inner-loop)=100 time steps
           bond 1 2 w 0.001 sigma 0.10 #bond distance between C-Cl collective variable
       end
   end
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
   task pspw car-parrinello
```

Appendix B: Metadynamics input file for $Cl^+CH_3Cl \rightleftharpoons ClCH_3 + Cl^-$

It is important to note the following: In the `coord_number` lines inside the metadynamics block, `range -0.2 1.2` sets each coordination number range to be [-0.2,1.2]. We did this to ensure the range actual [0,1] of each collective variable is fully covered; if one sets `range 0.0 1.0`, for example, some of the information at the extreme points (1,0) and (1,0) may be missing. Furthermore `nrange 101` implies that the bias potential will be computed on a 2D grid with 101 points along each variable. For a wide range of coordination number large value `nrange` may be used.

```
title "SN2-Exchange"
echo
start CH3Cl+Cl-
scratch_dir      ./perm_sn2_cn
permanent_dir    ./perm_sn2_cn
memory 1950 mb
charge -1

geometry units angstrom nocenter noautosym noautoz print xyz
C     0.000000    0.000000    0.000000
Cl    1.799895    0.017342   -0.008727
Cl   -4.000000    0.017342   -0.008727
H    -0.323446   -0.918592    0.460310    mass 2.0
H    -0.340365    0.851532    0.565115    mass 2.0
H    -0.340579    0.057144   -1.020309    mass 2.0
end

nwpw
   simulation_cell units angstroms
     boundary_conditions periodic
     SC 15.0
   end
   cutoff 30.0
   mult 1
   lmbfgs
   mapping 2
   xc pbe96
   lcao_skip
end

nwpw
  rotation off
end

task pspw energy

nwpw
   car-parrinello
   Nose-Hoover 1200.0 300.0 1200.0 300.0
   time_step    5.0
   fake_mass    600.0
   scaling      1.0 1.0
   end
end

# Briefly equilibrate
```

```
nwpw
   car-parrinello
     loop 10              100
     fei                  equi.fei
     xyz_filename         equi.xyz
     ion_motion_filename  equi.ion_motion
     emotion_filename     equi.emotion
   end
end
task pspw car-parrinello

# Begin Metadynamics here
nwpw
   car-parrinello
      loop                10 1000
      fei                 meta_sn2_cn.fei
      xyz_filename        meta_sn2_cn.xyz
      ion_motion_filename meta_sn2_cn.ion_motion
      emotion_filename    meta_sn2_cn.emotion
    end
    metadynamics
     print_shift  1
     update       10
coord_number sprik index1 1 index2 2 n 5.29177 r0 4.157397 w 0.00025 sigma 0.10 range -0.2 1.2 nrange 101
coord_number sprik index1 1 index2 3 n 5.29177 r0 4.157397 w 0.00025 sigma 0.10 range -0.2 1.2 nrange 101
   end
end
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
```

Appendix C: Umbrella sampling input file for $CH_3Cl \rightarrow CH_3^+ + Cl^-$

```
title "Umbrella sampling xi=1.7 Ang"
echo
start CH3Cl
scratch_dir       ./perm_us_1.7
permanent_dir     ./perm_us_1.7

#Memory
memory 1950 mb

#Charge
charge 0

#Begin PSPW block
nwpw

#  Box size
   simulation_cell units angstroms
     boundary_conditions periodic
     SC 11.4
   end

#  Plane KE cut-off
   cutoff 35.0

#  multiplicity
   mult 1

#  Minimizer
   lmbfgs
   mapping 2

#  XC Functional
   xc pbe96
   lcao_skip

#  CPMD block
   car-parrinello
     Nose-Hoover          1200.0 300.0 1200.0 300.0
     time_step            5.0
     fake_mass            600.0
     loop 10              1000
     scaling              1.0 1.0
     fei                  1.7.fei
     xyz_filename         1.7.xyz
     ion_motion_filename  1.7.ion_motion
     emotion_filename     1.7.emotion
   end

#  Disable molecular rotation
   rotation off

#  Biasing window potential
   auxiliary_potentials
     nonfragment
       index_start 1
```

```
          bond_spring 1 2 0.3  3.21253418 # index1, index2, K (in a.u.), window_center (in a.u.)
      end
   end
end
#End PSPW block

#Atomic positions; Hydrogen is replaced with Deuterium by making the mass 2.0
geometry units angstrom nocenter noautosym noautoz print xyz
  C      0.000000     0.000000     0.000000
  Cl     1.699897     0.016569    -0.008709
  H     -0.361760    -0.910123     0.456135    mass 2.0
  H     -0.378098     0.842982     0.559997    mass 2.0
  H     -0.377787     0.056198    -1.010534    mass 2.0
end

# Optimize wavefunction
task pspw energy

#CPMD
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
task pspw car-parrinello
```

Appendix D: Fortran 77 Codes for 1D WHAM Calculations

The FORTRAN 77 code, WHAM_1D.f, for computing the PMF is given below. WHAM_1D.f contains a subroutine PROB. PROB computes the following for a given window i for specified range $[\xi_{min}, \xi_{max}]$ for ξ using n_i snapshots:

(a) The biased probability distribution, $\rho_i^{(b)}(\xi)$, (histogram technique)

(b) The biasing potential $V_i(\xi) = K_i(\xi - \xi_i)^2$

The code assumes that the spring constant `KS` is constant for each window. KS must be converted into a 1D array if different spring constants are used (e.g. DATA statement).

```
!       COMPUTES THE FREE ENERGY DIFFERENCE FOR A SINGLE REACTION COORDINATE
!       USING THE "WEIGHTED HISTOGRAM ANALYSIS METHOD" (WHAM).
!       REFERENCE: B. ROUX, COMPUTER PHYSICS COMMUNICATIONS VOL. 91, PP. 275-282 (1995)

!       THE CODE ASSUMES THAT THE TRAJECTORIES 1.7.xyz, 1.8.xyz,....., 4.5.xyz ARE PRESENT

!       RAYMOND ATTA-FYNN
!       EMSL, PACIFIC NORTHWEST NATION LAB
!       ORIGINALLY WRITTEN ON OCTOBER 23, 2009
!       MODIFIED ON JUNE 1, 2011 WITH COMMENTS



        PROGRAM          WHAM_1D
        IMPLICIT NONE
        INTEGER          NW, I, J, K, N, NMAX1, NMAX2, NSAMP
        PARAMETER        (NMAX1=300,NMAX2=5000)
!                         NMAX1 IS THE MAX WINDOW DIMENSION
!                         NMAX2 IS THE MAX NUMBER OF HISTOGRAM BINS
        DOUBLE PRECISION XI_MAX, XI_MIN, DXI, XI_I, KB, TEMP, BETA
        DOUBLE PRECISION L,DT,TCUT,KS
        DOUBLE PRECISION NI(NMAX1)      ! SAMPLED SNAPSHOTS IN A GIVEN WINDOW
        DOUBLE PRECISION W(NMAX1,NMAX2) ! BIASING POTENTIAL
        DOUBLE PRECISION P_BIASED(NMAX1,NMAX2) ! BIASED DISTRIBUTION
        DOUBLE PRECISION P_UNBIASED(NMAX2) ! OPTIMAL ESTIMATE OF UNBIASED DISTRIBUTION
        DOUBLE PRECISION FI(NMAX1) ! FI = EXP(-fi*BETA)
        DOUBLE PRECISION HIST(NMAX2) ! DISTRIBUTION (HISTOGRAM) OF A GIVEN SIMULATION
        DOUBLE PRECISION EPS,TOL,FI_OLD,PMIN,TMP1,TMP2
        DOUBLE PRECISION V(NMAX2),FAC,NUMERATOR,DENOMINATOR,FI_NEW
        CHARACTER        F*3 ! THIS WAS DEFINED FOR FILE READING PURPOSES

*****************************EDIT THIS SECTION*********************************
        XI_MIN=1.65D0                   ! MIN REACTION COORDINATE IN ANGSTROM
        XI_MAX=4.55D0                   ! MAX REACTION COORDINATE IN ANGSTROM
        DXI   =0.025D0                  ! BIN WIDTH
        N=1+DINT((XI_MAX-XI_MIN)/DXI)   ! TOTAL NUMBER OF BINS
        L     =11.4D0                   ! SIMULATION BOX LENGTH IN ANGSTROM
        DT    =30.D0                    ! TIME INTERVAL FOR SAVING CONFIGURATIONS IN A.U.
        DT    =DT/(41.3413733D+3)       ! CONVERT TIME STEP FROM AU TO PICOSECONDS
        TCUT  =1.D0                     ! EQUILIBRATION TIME IN PICOSECONDS
        KS    =0.3D0                    ! SPRING CONSTANT IN A.U. (SAME FOR EACH WINDOW)
        NSAMP =2                        ! SAMPLE EVERY OTHER POINT.
!                                         TO SAMPLE AT A LOWER RATE INCREASE NSAMP.
        KB    =0.316679D-5              ! BOLTZMANN'S CONSTANT IN A.U./KELVIN
```

```fortran
      TEMP  =300.D0                  ! SIMULATION TEMPERATURE IN KELVIN
      BETA  =1.D0/KB/TEMP            ! INVERSE Kb*T
      FAC   =627.509D0               ! CONVERSION FACTOR FROM A.U. TO KCAL/MOL
      NW    =29                      ! TOTAL NUMBER OF WINDOWS
      TOL   =1.D-7                   ! SCF TOLERANCE
****************************END EDITING*************************************

!********************COMPUTATION BEGINS HERE*********************************

      DO I = 1, NW ! LOOP OVER WINDOWS
         XI_I = 1.7D0 + (DBLE(I)-1.D0)*0.1D0 ! COMPUTE CENTRAL COORDINATE OF WINDOW I
         WRITE(F,'(F3.1)')XI_I ! CHARACTER F TAKES VALUES 1.7, 1.8, 1.9,......,4.5
!        XYZ FILES NAMED 1.7.xyz, 1.8.xyz,......, 4.5.xyz WILL BE OPENED SUCCESSIVELY
!        INSIDE THE SUBROUTINE PROB BELOW
         CALL PROB(XI_MIN,DXI,N,XI_I,L,DT,TCUT,KS,NSAMP,F,K,HIST,V) ! CALL SUBROUTINE
         NI(I)=DBLE(K)
         DO J = 1, N ! LOOP OVER HISTORGRAM POINTS IN WINDOW I
            P_BIASED(I,J)=HIST(J)  ! BIASED DISTRIBUTION OF WINDOW I AT COORDINATE XI_J
            W(I,J)=DEXP(-BETA*V(J)) ! RESTRAINING POTENTIAL OF WINDOW I AT COORDINATE XI_J
         END DO
      END DO

!     INITIAL GUESS OF UNITY FOR FI = DEXP(-fi*BETA)
      DO I = 1, NW
         FI(I) = 1.D0
      END DO

100   CONTINUE

      DO J = 1, N
         NUMERATOR   = 0.D0
         DENOMINATOR = 0.D0

!        COMPUTING THE DENOMINATOR OF EQUATION (1)
         DO I  = 1, NW
            DENOMINATOR = DENOMINATOR + NI(I)*W(I,J)/FI(I)
         END DO

!        COMPUTING THE NUMERATOR OF EQUATION (1)
         DO I  = 1, NW
            NUMERATOR = NUMERATOR + NI(I)*P_BIASED(I,J)
         END DO

         TMP1=NUMERATOR/DENOMINATOR

!        CONVERT ZERO PROBABILITY TO VERY SMALL POSITIVE NUMBER
         IF(TMP1.EQ.0.D0)TMP1=1.D-15

         P_UNBIASED(J)=TMP1

      END DO

!     COMPUTE NEW FI BASED AND THE OLD USE EPS=SUM{(1-FI_NEW/FI_OLD)**2}
!     AS THE CONVERGENCE PARAMETER

      EPS = 0.D0
      DO I = 1, NW
```

```fortran
      FI_OLD = FI(I)
      FI_NEW   = 0.D0
      DO J = 1, N
          FI_NEW = FI_NEW + W(I,J)*P_UNBIASED(J)
      END DO
      FI(I)=FI_NEW
      EPS = EPS + (1.D0 - FI_NEW/FI_OLD)**2
      END DO
      WRITE(*,*)EPS
      IF(EPS.GT.TOL)GOTO 100

      WRITE(*,*)'CONVERGED'


!     FIND FREE ENERGY SHIFT
      PMIN = -1.D+8
      DO J = 1, N
          PMIN=MAX(PMIN,(1.D0/BETA)*DLOG(P_UNBIASED(J)))
      END DO


!     PRINT PMF
      OPEN(10,FILE='FREE_ENER.dat',STATUS='UNKNOWN')
      DO J = 1, N
          TMP1= XI_MIN + DXI*(DBLE(J)-0.5D0) ! REACTION COORDINATE IN ANGSTROM
          TMP2=(-(1.D0/BETA)*DLOG(P_UNBIASED(J))+PMIN)*FAC ! PMF IN KCAL/MOL
          WRITE(10,1000)TMP1,TMP2
      END DO
1000  FORMAT(2F12.6)
      CLOSE(10)
      WRITE(*,*)'PMF printed to FREE_ENER.dat'

      STOP

      END




!*******************SUBROUTINE PROB BEGINS HERE********************************
!=============================================================================
!     FOR A GIVEN WINDOW, THIS SUBROUTINE RETURNS:
!       (a) THE NUMBER OF SAMPLED SNAPSHOTS (NI)
!       (b) THE NORMALIZED PROBABILITY DISTRIBUTION (HIST)
!       (c) THE BIASING POTENTIAL AROUND THE WINDOW CENTER (V)



      SUBROUTINE PROB(XI_L,DXI,N,XI_I,L,DT,TCUT,KS,NSAMP,F,NI,HIST,V)
      IMPLICIT NONE
      INTEGER          I,J,N,NC,NAT,BIN,NSAMP,T_TOT,NMAX,NI
      PARAMETER        (NMAX=5)
      DOUBLE PRECISION L,XI_L,XI_U,DXI,DT,TCUT,FAC,XI_I,KS
      DOUBLE PRECISION POS(3,NMAX),TMP,DIST
      DOUBLE PRECISION HIST(N),V(N),TMP2
      CHARACTER        F*3
      DATA             FAC/1.889725988578923D0/
```

```fortran
!     OPEN FILES 1.7.xyz, 1.8.xyz,........,4.5.xyz (based on F)
      OPEN(10,FILE=F//'.xyz',STATUS='OLD')

!     THE OUTPUT NORMALIZED PROBABILITY DISTRIBUTION WILL STORED IN
!     THE FILES PROB_1.7.dat, PROB_1.8.dat, AND SO ON.
      OPEN(11,FILE='PROB_'//F//'.dat',STATUS='UNKNOWN')

      DO I = 1, N                    ! INITIAL HISTOGRAM VALUES TO ZERO
         HIST(I) = 0.D0
      END DO

      NC = 0 ! INITIALIZE TOTAL NUMBER OF SNAPSHOTS TO ZERO
      NI = 0 ! INITIALIZE TOTAL NUMBER OF SAMPLED SNAPSHOTS TO ZERO

!     READING SNAPSHOTS UNTIL END OF FILE IS REACHED
      DO WHILE(.TRUE.)
         READ(10,*,ERR=30,END=40)NAT
         READ(10,*,ERR=30,END=40)
         DO I = 1, NAT
            READ(10,*,ERR=30,END=40)F,(POS(J,I),J=1,3),TMP,TMP,TMP
         END DO
         NC = NC + 1
         T_TOT=DT*DBLE(NC)              ! TOTAL SIMULATION TIME UP TILL NOW
         IF(MOD(NC,NSAMP).NE.0)GOTO 50 ! SAMPLING
         IF(T_TOT.LT.TCUT)GOTO 50      ! DISCARD PRE-EQULIBRATION SNAPSHOTS
         NI = NI + 1
         DIST = 0.D0                    ! INITIALIZE C-Cl DISTANCE TO ZERO
         DO J = 1, 3                    ! COMPUTE DIST (IN ANGSTROM) SUBJECT TO PBC
            TMP  = POS(J,1)-POS(J,2)
            TMP  = TMP - L*DNINT(TMP/L)
            DIST = DIST + TMP*TMP
         END DO
         DIST=DSQRT(DIST)
         DIST=DIST-XI_L  ! SHIFT DISTANCE RELATIVE LOWER BOUND
         BIN = 1 + DINT(DIST/DXI) ! LOCATE BIN
         IF(BIN.GT.N)GOTO 50       ! DISCARD POINTS GREATER THAN XI_L
         HIST(BIN)=HIST(BIN) + 1.D0 ! PLACE IN APPROPRIATE BIN
50       CONTINUE
      END DO

 30   CONTINUE
      WRITE(*,*)'ERROR IN READING INPUT FILE'

 40   CONTINUE
      CLOSE(10)

!     COMPUTE NORMALIZATION FACTOR
      TMP  = 0
      DO I = 1, N
         TMP  = TMP  + HIST(I)
      END DO
!     NOTE THAT, IN PRINCIPLE, TMP=DBLE(NI)

!     COMPUTE NORMALIZED DISTRIBUTION AND BIASING POTENTIAL
      WRITE(11,'(A)')'# COORDINATE    POTENTIAL     PROBABILITY'
      DO I = 1, N
         TMP2     = XI_L + DXI*(DBLE(I)-0.5D0)    ! REACTION COORDINATE IN ANGSTROM
```

```fortran
      HIST(I) = HIST(I)/TMP                 ! NORMALIZED PROBALITY AT TMP2
      V(I)    = KS*(TMP2-XI_I)**2           ! BIASING WINDOW POTENTIAL
      V(I)    = V(I)*FAC**2                 ! CONVERT POTENTIAL TO A.U.
      WRITE(11,'(3F12.6)')TMP2,HIST(I),V(I) ! WRITE OUT INFORMATION
    END DO
    CLOSE(11)

    RETURN
    END
```