Supporting Information for: BLaDE: A Basic Lambda Dynamics Engine for GPU Accelerated Molecular Dynamics Free Energy Calculations

Ryan L. Hayes[†] and Charles L. Brooks III[*,†,‡]

[†]*Department of Chemistry, University of Michigan, Ann Arbor, Michigan 48109, United States*

[‡]*Biophysics Program, University of Michigan, Ann Arbor, Michigan 48109, United States*

E-mail: brookscl@umich.edu

Phone: (734) 647-6682

Simulation Details

Unless otherwise noted, BLaDE simulations utilized a friction constant of 0.1 ps^{-1}, as numerical experiments suggested higher values slowed convergence of simulations. Standalone BLaDE simulations used an alchemical friction coefficient of 1 ps^{-1}, while BLaDE in CHARMM and DOMDEC in CHARMM simulations used an alchemical friction coefficient of 5 ps^{-1}. Monte Carlo pressure coupling moves were attempted every 100 steps and coupled to a 1 atmosphere pressure bath with volume changes sampled from a Gaussian distribution with a standard deviation of 100 Å^3. PME electrostatics used $\beta^{-1} = 3.125$ Å, an interpolation order of 6 and a maximum grid spacing of 1.1 Å (the box was divided by the minimum integer having powers of 2 times 16, 18, 20, 24, or 27 that satisfied the maximum grid spacing). Van der Waals force switching used a switch radius of 9 Å and force cutoff of 10 Å. A time step of 2 fs was used in all simulations.

For constant energy simulations and time step stability simulations of the 216 TIP3P water molecule box and DHFR, a different friction coefficient of 1 ps^{-1}, a force cutoff of 10 Å and a force switch radius of 8.5 Å, and a temperature of 300 K were used. For these simulations, DHFR also used an interpolation order of 6. For constant energy simulations, initial equilibration was 1 ns in an NVT, followed by 1 ns of NVE production. Time step stability simulations of the 216 TIP3P water molecule box used a 1 ns NVT equilibration followed by NVT production for 500000 steps at the indicated time step, while DHFR simulations used a 1 ns equilibration followed by a 1 ns production.

Profiling runs were run for 1 ns in sets of 10 with varying software and hardware configurations. DHFR profiling runs mimicked the stripped down nature of the standard DHFR benchmark system, except they utilized force switching for van der Waals interactions because this is the only option currently available in BLaDE. A PME interpolation order of 4 was used, and van der Waals force switching used a switch radius of 7.5 Å and a cutoff radius of 9 Å. The use of force switching rather than the standard potential switching for the DHFR benchmark was shown to have no effect on computational cost for DOMDEC and OpenMM. Unlike the alchemical simulations, DHFR profiling was also run at constant volume, which substantially improved the efficiency of DOMDEC, but had little effect on BLaDE and OpenMM. The temperature was 298 K, and the SHAKE tolerance was 1×10^{-8}. Other profiling simulations used the longer cutoffs and higher interpolation order listed previously, a temperature of 298.15 K, and a more stringent SHAKE tolerance of 1×10^{-9}.

Free energy comparisons utilized the same options. For T4L and HSP90, full flattening runs of both ensembles (folded and unfolded or complex and solvent) were run, followed by production. For RNase H, only the folded ensemble was studied, using the same set of previously determined biases for all simulations. Simulation setups have been described in previous publications.[1–3] T4L utilized five independent trials of 40 ns for production, RNase H utilized twelve independent trials of 400 ns for production, and HSP90 utilized five independent trials of 30 ns for production. Production was rerun in T4L as described previously if simulations were judged to be uncoverged based on adjustments of more than 1.2 kcal/mol to any of the fixed biases after the production simulation. In addition to comparing with a reference simulation run with the same options as DOMDEC in CHARMM, simulations were also compared to experiment for the 18 sequences in T4L, the 12 sequences in RNase H, and the 9 ligands in HSP90 with measured free energies (Table S1).

Table S1: RMS Difference of MSλD Free Energies from Experiment (kcal/mol)

	T4L	RNase H	HSP90
	Protein MSλD	Protein MSλD	Ligand MSλD
Standalone BLaDE	0.95	1.25	0.45
BLaDE in CHARMM	0.91	1.27	0.68
DOMDEC in CHARMM	0.93	1.25	0.60
DOMDEC in CHARMM (reference)	0.97	1.18	0.46

Direct Nonbonded Kernels

In the main text, we noted that previous work found performance improvements when writing separate nonbonded kernels for blocks with more than 12 atoms using a staggered approach and less than 12 atoms using a reduction approach.[4] Reduction means each thread operates on the same j atom simultaneously, and the result is then summed and stored, while staggering means each thread works on a different j atom, so forces on j can be accumulated without the need for a reduction over all 32 threads before moving on to the next j atom. In BLaDE we tried both methods (without splitting the blocks into two groups of those with fewer and more atoms), and found always using the reduction approach was faster than always using the staggered approach.

We also implemented a kernel that switched between the two approaches at an empirically determined break-even point. We found the break even point to be 26 j atoms, which is much higher than the previously identified break even point of 12 atoms.[4] This suggests reduce operations are less expensive than they once were, possibly due to the ability to reduce with shuffle operations instead of shared memory. The kernel switching between reduction and staggering failed to result in a significant speedup, achieving almost exactly the same performance as the pure reduction kernel. The overhead of bookkeeping whether to use reduction or staggering eliminated the gains from the reduced amount of work, and resulted in more complicated code, and was therefore abandoned.

Another nonbonded kernel was written in an attempt to eliminate extra operations. While noninteracting j atoms are identified by whether they are too far from the bounding box of i atoms and skipped, noninteracting i atoms still occupy a thread, thus the full interaction between the i and j blocks must be computed, even if there is only one interacting i atom. Therefore, we load only interacting j atoms from as many blocks as necessary to achieve a full complement of 32 atoms to occupy all 32 threads, and then instead of looping over j atoms, we loop over i atoms, and skip non-interacting i atoms as well. This introduced substantially more bookkeeping and hence more registers of local variables. On RTX 2080 Ti GPUs, this reduced the occupancy slightly meaning fewer pairs of blocks could run at the same time on the GPU, but did improve performance by almost 5%; however, earlier GPUs had fewer registers available, so occupancy decreased substantially, resulting in substantially slower execution. Consequently, this approach was abandoned as well, even though it gave slight improvements on RTX 2080 Ti GPUs, because of its inconsistency across GPUs and because of the substantial extra code complexity.

References

(1) Hayes, R. L.; Vilseck, J. Z.; Brooks, C. L., III *Protein Science* **2018**, *27*, 1910–1922.

(2) Hayes, R. L.; Nixon, C. F.; Marqusee, S.; Brooks, C. L., III *Journal* **In preparation**, *Vol*, Page.

(3) Raman, E. P.; Paul, T. J.; Hayes, R. L.; Brooks, C. L., III *Journal of Chemical Theory and Computation* **2020**, *16*, 7895–7914.

(4) Eastman, P.; Pande, V. S. *Journal of Computational Chemistry* **2010**, *31*, 1268–1272.