Contents lists available at ScienceDirect

Computer Physics Communications

GROMACS implementation of free energy calculations with non-pairwise Variationally derived Intermediates☆,☆☆

Martin Reinhardt, Helmut Grubmüller *

Max Planck Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany

ARTICLE INFO

ABSTRACT

Gradients in free energies are the driving forces of physical and biochemical systems. To predict free energy differences with high accuracy, Molecular Dynamics (MD) and other methods based on atomistic Hamiltonians conduct sampling simulations in intermediate thermodynamic states that bridge the configuration space densities between two states of interest ('alchemical transformations'). For uncorrelated sampling, the recent Variationally derived Intermediates (VI) method yields optimal accuracy. The form of the VI intermediates differs fundamentally from conventional ones in that they are non-pairwise, i.e., the total force on a particle in an intermediate states cannot be split into additive contributions from the surrounding particles. In this work, we describe the implementation of VI into the widely used GROMACS MD software package (2020, version 1). Furthermore, a variant of VI is developed that avoids numerical instabilities for vanishing particles. The implementation allows the use of previous non-pairwise potential forms in the literature, which have so far not been available in GROMACS. Example cases on the calculation of solvation free energies, and accuracy assessments thereof, are provided.

Program summary
Program Title: GROMACS-VI-Extension
CPC Library link to program files: https://doi.org/10.17632/7yvc8mmnyv.1
Developer's repository link: https://www.mpibpc.mpg.de/gromacs-vi-extension and https://www.gitlab.gwdg.de/martin.reinhardt/gromacs-vi-extension
Licensing provisions: LGPL
Programming language: C++14, CUDA
Nature of problem: The free energy difference between two states of a thermodynamic system is calculated using samples generated by simulations based on atomistic Hamiltonians. Due to the high dimensionality of many applications as in, e.g., biophysics, only a small part of the configuration space can be sampled. The choice of the sampling scheme critically affects the accuracy of the final free energy estimate. The challenge is, therefore, to find the optimal sampling scheme that provides best accuracy for given computational effort.
Solution method: Sampling is commonly conducted in intermediate states, whose Hamiltonians are defined based on the Hamiltonians of the two states of interest. Here, sampling is conducted in the variationally derived intermediate states that, under the assumption of uncorrelated sample points, yield optimal accuracy. These intermediates differ fundamentally from the common intermediates in that they are non-pairwise, i.e., the forces on a particle are only additive in the end state, whereas the total force in the intermediate states cannot be split into additive contributions from the surrounding particles.

References
[1] M. Reinhardt, H. Grubmüller, Determining Free-Energy Differences Through Variationally Derived Intermediates, *Journal of Chemical Theory and Computation* 16 (6) 3504-3512 (2020).

1. Introduction

Thermodynamic systems are driven by free energy gradients. Hence, knowledge thereof is key to the molecular understanding of a wide range of biophysical and chemical processes, as well

---

as to applications in the pharmaceutical [1–3] and material sciences [4–6]. Consequently, *in silico* calculations of free energies are popular in providing complementary insights to experiments or assisting the selection of chemical compounds in the early stages of drug discovery projects.

The microscopic calculation of the free energy,

$$\Delta G = -\beta^{-1} \ln Z \tag{1}$$

$$= -\beta^{-1} \ln \int_{-\infty}^{\infty} e^{-\beta H(\mathbf{x})} d\mathbf{x} , \tag{2}$$

requires integration over all positions $\mathbf{x}$ of all particles in the system, where Z denotes the partition sum, $\beta = 1/(k_B T)$ the thermodynamic β, k_B the Boltzmann constant, T the temperature and $H(\mathbf{x})$ the Hamiltonian. As an exact integration is not feasible for high-dimensional $\mathbf{x}$ in case of many particles, sampling based approaches such as Monte-Carlo (MC) or Molecular Dynamics (MD) simulations are commonly used. Furthermore, in practice, it oftentimes suffices to know only the free energy difference between two states, which can be calculated much more accurately. The most basic approach,

$$\Delta G_{A,B} = -\beta^{-1} \ln \left\langle e^{-\beta[H_B(\mathbf{x})-H_A(\mathbf{x})]} \right\rangle_A \tag{3}$$

rests on the Zwanzig formula [7]. The brackets $\langle \rangle_A$ indicate an ensemble average over A is calculated. Another method with close relations to Eq. (3) that use samples from both A and B is the Bennett Acceptance Ratio (BAR),

$$\Delta G_{A,B} = \beta^{-1} \ln \frac{\left\langle f[\beta(H_A(\mathbf{x}) - H_B(\mathbf{x}) + C)] \right\rangle_B}{\left\langle f[\beta(H_B(\mathbf{x}) - H_A(\mathbf{x}) - C)] \right\rangle_A} + C , \tag{4}$$

where it has been shown that the variance of the free energy estimate is minimal $f(x) = 1/(1 + \exp(x))$ is the Fermi function, and if $C \approx \Delta G - \beta^{-1} \ln \left(\frac{n_A}{n_B} \right)$, where n_A and n_B denote the number of sample points in A and B, respectively. A further extension is the multistate BAR (MBAR) method [8].

For sampling based approaches, the accuracy of a free energy difference estimate between two states A and B generally improves when sampling is not only conducted in A and B, but also in intermediate states. Commonly, a mostly linear interpolation between the end state Hamiltonians $H_A(\mathbf{x})$ and $H_B(\mathbf{x})$ is used,

$$H_{lin}(\mathbf{x}, \lambda) = (1 - \lambda)H_A(\mathbf{x}) + \lambda H_B(\mathbf{x}) , \tag{5}$$

where $\lambda \in [0, 1]$ denotes the path variable. Oftentimes, a λ dependence of the end state Hamiltonians is introduced that enables the use of soft-core potentials [9–11]. These potentials are chosen such that divergences in case of vanishing particle for, e.g., the calculation of solvation free energies (where the molecules "vanishes" from solution), are avoided. A step-wise summation,

$$\Delta G_{AB} = \sum_{i=1}^{N-1} \Delta G_{i,i+1} \tag{6}$$

yields the total free energy difference, where N denotes the total number of states. In the sum of Eq. (6), $i = 1$ corresponds to state A and $i = N$ to state B, respectively. Alternatively, for many steps the difference can be calculated with Thermodynamic Integration (TI) [12],

$$\Delta G_{AB} = \int_0^1 \left\langle \frac{\partial H(\mathbf{x}, \lambda)}{\partial \lambda} \right\rangle_\lambda d\lambda . \tag{7}$$

Importantly, advantageous definitions of intermediate states exist that go beyond the definition of Eq. (5). For example, variationally derived intermediates (VI) [13,14] minimize the mean

squared error (MSE) of free energy estimates using FEP and BAR. An easily parallelizable approximation for a small number of states is

$$H_{VI}(\mathbf{x}, \lambda) = -\frac{1}{2\beta} \ln \left\{ (1 - \lambda) \exp\left[-2\beta H_A(\mathbf{x}) \right] \right.$$
$$\left. + \lambda \exp\left[-2\beta \left(H_B(\mathbf{x}) - C \right) \right] \right\} , \tag{8}$$

where, similar to BAR, the free energy difference estimate is optimal if $C \approx \Delta G$. It is similar in shape to the minimum variance path (MVP) [15–17] for TI [12] (2 vs. 1/2 in the exponents). Enveloping Distribution Sampling (EDS) [18,19], and extensions such as Accelerated EDS [20,21] use a reference potential similar in shape to Eq. (8) to calculate the free energy difference between two or more end states.

Note a particular characteristic of the VI sequence and related methods: Its Hamiltonians cannot be formulated as the pair-wise sum of interaction potentials for all particles. To see this, firstly consider the force on a particle i for the linear interpolation scheme

$$\mathbf{F}_i^{lin}(\mathbf{x}, \lambda) = (1 - \lambda)\mathbf{F}_i^A(\mathbf{x}) + \lambda \mathbf{F}_i^B(\mathbf{x}) , \tag{9}$$

where $\mathbf{F}_i^A(\mathbf{x})$ and $\mathbf{F}_i^B(\mathbf{x})$ denote the sum of the forces on particle i in end state A and B, respectively. Here, only the particles with a position close enough to particle i such that their pair-wise force is non-negligible contribute to $\mathbf{F}_i^{lin}(\mathbf{x})$. In contrast, the VI force obtained through the derivative of Eq. (8)

$$\mathbf{F}_i^{VI}(\mathbf{x}, \lambda) = -\frac{\partial H_{VI}(\mathbf{x})}{\partial \mathbf{x}_i} \tag{10}$$

$$= \exp\left[2\beta H_{VI}(\mathbf{x}) \right]$$
$$\left\{ (1 - \lambda) \exp\left[-2\beta H_A(\mathbf{x}) \right] \mathbf{F}_i^A(\mathbf{x}) \right. \tag{11}$$
$$\left. + \lambda \exp\left[-2\beta \left(H_B(\mathbf{x}) - C \right) \right] \mathbf{F}_i^B(\mathbf{x}) \right\} .$$

still depends on the Hamiltonians $H_A(\mathbf{x})$ and $H_B(\mathbf{x})$, and therefore depends on the configurations of all particles in the system. As a consequence, changes in any part of the system will affect the force on every particle, regardless of the interaction distance. Hence, the non-linear VI formulation introduces a coupling between all particles that goes beyond the physical interactions.

In this work, we, firstly, describe our implementation of the VI approach, and, by extension, also the MVP and basic principles of the EDS methods for two end states, into GROMACS [22–24]. It is among the most widely used MD software packages; however, none of the above approaches are available so far in GROMACS. Secondly, we introduce an approach to avoid singularities for vanishing particles with VI.

2. Avoiding end state singularities

Interestingly, the VI sequence, Eq. (8), already exhibits soft-core characteristics for vanishing particles, as shown in Fig. 1(a) on the example of a two-particle Lennard-Jones (LJ) potential. However, divergences can still occur when configurations from the decoupled states are evaluated at foreign states, i.e., the ones that no sampling is conducted in, but that the Hamiltonian is evaluated at such as, e.g., state B in Eq. (3). Furthermore, when two particles start to overlap, very small changes in their separation r lead to large changes in force, which causes instabilities due to finite integration steps.

To avoid these divergences, a dependence of the end state Hamiltonians on λ analogous to common soft-core potentials [9] is introduced, i.e., $H_A = H_A(\mathbf{x}, \lambda)$ with $H_A(\mathbf{x}, 0) = H_A(\mathbf{x})$, and $H_B = H_B(\mathbf{x}, \lambda)$, with $H_B(\mathbf{x}, 1) = H_B(\mathbf{x})$, respectively. For two particle i and j with distance r_{ij}, the Coulomb and Lennard-Jones
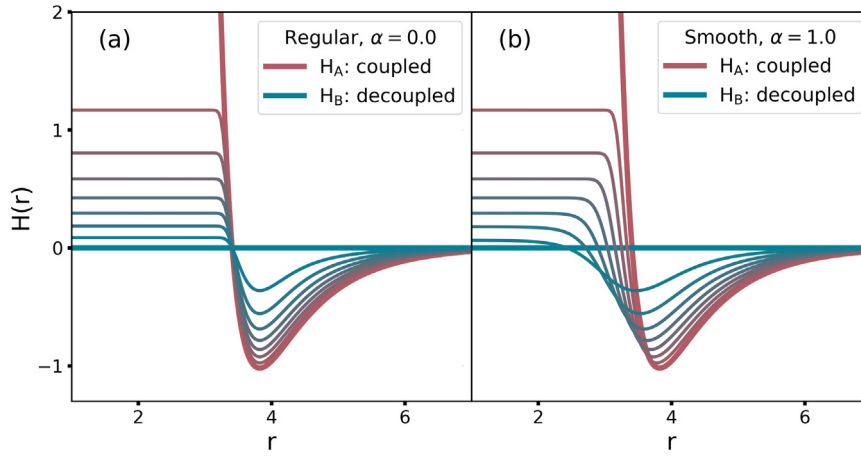
Fig. 1. Intermediate VI states for a vanishing particle system. The thick red line shows the Lennard-Jones potential between two particles. The blue one shows the decoupled end state, i.e., the particles do not "see" each other anymore. The interpolated colors represent the intermediate states. (a) The VI sequence without and (b) with λ dependent end states.

interactions in state A and B are calculated based on the modified distances r_A and r_B, respectively, that are defined as

$$r_A(r_{ij}, \lambda) = \left(\alpha \sigma_{ij}^6 \lambda^p + r_{ij}^6 \right)^{\frac{1}{6}}, \tag{12}$$

$$r_B(r_{ij}, \lambda) = \left(\alpha \sigma_{ij}^6 (1-\lambda)^p + r_{ij}^6 \right)^{\frac{1}{6}}, \tag{13}$$

where α and p are soft-core parameters to be specified by the user, and σ_{ij} the Lennard-Jones parameter in the coupled state. For a system of two Lennard-Jones particles, Fig. 1 shows the resulting VI states without (a) and with (b) the use of λ dependent end states. As can be seen, the transition to the overlap region becomes markedly smoother.

Secondly, for increasingly complex molecules, the likelihood of barriers between the relevant parts of configuration space of the end states rises. Aside of additional techniques such as replica exchange, or meta-dynamics, the factor 2 in the exponent can be replaced by a user specific smoothing factor s introduced in the EDS [18,19] method. In the limit of small s, a series expansion of the exponential terms yields the conventional pathway, i.e., Eq. (5). The modified VI sequence thus reads as

$$H_{VI}(\mathbf{x}, \lambda) = -\frac{1}{s\beta} \ln \left\{ (1-\lambda) \exp\left[-s\beta H_A(\mathbf{x}, \lambda) \right] \right.$$
$$\left. + \lambda \exp\left[-s\beta \left(H_B(\mathbf{x}, \lambda) - C \right) \right] \right\}, \tag{14}$$

and the modified force on particle i,

$$\mathbf{F}_i^{VI}(\mathbf{x}, \lambda) = -\frac{\partial H_{VI}(\mathbf{x}, \lambda)}{\partial \mathbf{x}_i} \tag{15}$$

$$= \exp\left[s\beta H_{VI}(\mathbf{x}, \lambda) \right]$$
$$\left\{ (1-\lambda) \exp\left[-s\beta H_A(\mathbf{x}, \lambda) \right] \mathbf{F}_i^A(\mathbf{x}, \lambda) \right. \tag{16}$$
$$\left. + \lambda \exp\left[-s\beta \left(H_B(\mathbf{x}, \lambda) - C \right) \right] \mathbf{F}_i^B(\mathbf{x}, \lambda) \right\}.$$

Along similar lines, the derivate with respect to λ reads as

$$\frac{\partial H_{VI}(\mathbf{x}, \lambda)}{\partial \lambda} = \frac{\exp\left[s\beta H_{VI}(\mathbf{x}, \lambda) \right]}{\beta s}$$
$$\left\{ \left((1-\lambda)s\beta \frac{\partial H_A(\mathbf{x}, \lambda)}{\partial \lambda} + 1 \right) \exp\left[-s\beta H_A(\mathbf{x}, \lambda) \right] \right.$$
$$\left. + \left(\lambda s\beta \frac{\partial H_B(\mathbf{x}, \lambda)}{\partial \lambda} - 1 \right) \exp\left[-s\beta (H_B(\mathbf{x}, \lambda) - C) \right] \right\} \tag{17}$$

and depends on the derivatives $\partial H_A(\mathbf{x}, \lambda)/\partial \lambda$ and $\partial H_B(\mathbf{x}, \lambda)/\partial \lambda$ in the end states. Eq. (17) is used for TI [12].

Due to the dependence of Eq. (14) on C, where the accuracy is optimal if $C \approx \Delta G_{AB}$, the free energy difference has to be determined in an iterative process,

$$C^{n+1} = \Delta G_{AB'} + C^n, \tag{18}$$

where C^n denotes the free energy guess at iteration step n. The free energy difference $\Delta G_{AB'}$ is obtained from simulations between state A and B', where the latter denotes the end state shifted by the constant C, i.e., that is governed by $H_B'(\mathbf{x}, \lambda) = H_B(\mathbf{x}, \lambda) - C$. The difference $\Delta G_{AB'}$ converges to zero, such that the desired quantity $\Delta G_{AB} = \Delta G_{AB'} + C^n \approx C^n$ at the end of the iteration process.

3. Program structure and usage

The end states Hamiltonians,

$$H_A(\mathbf{x}, \lambda) = H_A^\lambda(\mathbf{x}, \lambda) + H^c(\mathbf{x}) \tag{19}$$

$$H_B(\mathbf{x}, \lambda) = H_B^\lambda(\mathbf{x}, \lambda) + H^c(\mathbf{x}), \tag{20}$$

can be split into the λ-dependent energy contributions $H_A^\lambda(\mathbf{x}, \lambda)$ and $H_B^\lambda(\mathbf{x}, \lambda)$, respectively, and the common contributions summarized by $H^c(\mathbf{x})$ that are equal in both end states, such as water–water interactions. To calculate $H_A(\mathbf{x}, \lambda)$ and $H_B(\mathbf{x}, \lambda)$, GROMACS only evaluates the λ-dependent contributions separately for the end states, whereas $H^c(\mathbf{x})$ is calculated only once. Note that, due to the λ dependence of the end states, $H_A^\lambda(\mathbf{x}, \lambda)$ and $H_B^\lambda(\mathbf{x}, \lambda)$ differ for different intermediates for $\alpha > 0$.

The same holds for the VI sequence, Eq. (14). Inserting Eqs. (19) and (20), yields

$$H_{VI}(\mathbf{x}, \lambda) = H_{VI}^\lambda(\mathbf{x}, \lambda) + H^c(\mathbf{x}), \tag{21}$$

where $H_{VI}^\lambda(\mathbf{x}, \lambda)$ is described by Eq. (14), where the end states Hamiltonians $H_A(\mathbf{x}, \lambda)$ and $H_A(\mathbf{x}, \lambda)$ have been replaced by the parts $H_A^\lambda(\mathbf{x}, \lambda)$ and $H_B^\lambda(\mathbf{x}, \lambda)$, respectively, that only sum over λ-dependent interactions. The same principle applies to the calculation of the forces and λ-derivatives. Therefore, the computational effort of VI is very close to the using conventional intermediates.

However, in the current GROMACS implementation structure, all force and energy contributions from different interaction types are interpolated between the end states right after they have
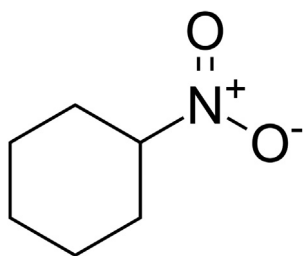
Fig. 2. Structure of nitrocyclohexane, which is used as an example case.

been calculated, i.e., the overall calculation has the form,

$$H_{lin}^{\lambda}(\mathbf{x}, \lambda) = \underset{\substack{interaction \\ type\ k}}{\sum} ... \underset{\substack{particles \\ i,j}}{\sum} (1-\lambda)H_A^k(\mathbf{x}_{i,j}, \lambda) + \lambda H_B^k(\mathbf{x}_{i,j}, \lambda) \quad (22)$$

$$F_{\lambda}^i(\mathbf{x}) = \underset{\substack{interaction \\ type\ k}}{\sum} ... \underset{particles\ j}{\sum} (1-\lambda)F_A^k(\mathbf{x}_{i,j}, \lambda) + \lambda F_B^k(\mathbf{x}_{i,j}, \lambda) . \quad (23)$$

Whereas this has the least memory requirement, for VI, the full Hamiltonians and forces in the end states need to be known before the individual forces can be calculated. Therefore, the end states Hamiltonians and forces are stored separately. After all λ-dependent contributions have been collected, first the Hamiltonian and subsequently the forces are calculated.

The implementation was built based on the GROMACS 2020 version 1 (forked on October 19th, 2019 from the master branch of the developer's repository). VI can be used with the new following entries in the mdp (i.e., input parameter) file:

```
variational–morphing = 1
```

```
smoothing–factor        = 2.
deltag–estimate         = 10.3   ; in kJ / mol
```

Furthermore, the option

```
nstcalcenergy           = 1
```

should be set, as the force calculation requires the Hamiltonians of the end state. The λ dependence of the end state Hamiltonians for VI are controlled via the already existing soft-core infrastructure,

```
sc–alpha                = 0.7
sc–r–power              = 6
sc–coul                 = no
sc–sigma                = 0.3
```

By nature of Eq. (14), the transformation only takes place along a single λ variable, to be specified by the mdp parameter `fep-lambdas`. As such, it is not possible to decouple several interactions simultaneously with different λ spacing for each type. It is, of course, possible to decouple electrostatic and LJ interactions in a sequence, that can be defined via `coul-lambdas` and `vdw-lambdas`, respectively, whereas the other is set to either zero (full interaction) or one (no interaction) for all intermediate states.

4. Example and test cases

When VI is switched off, all interactions are calculated as in Eqs. (22), (23) and (17). To test that VI collects all contributions correctly, for the following options in the mdp file,

```
variational–morphing    = 1
linear–test             = 1
```
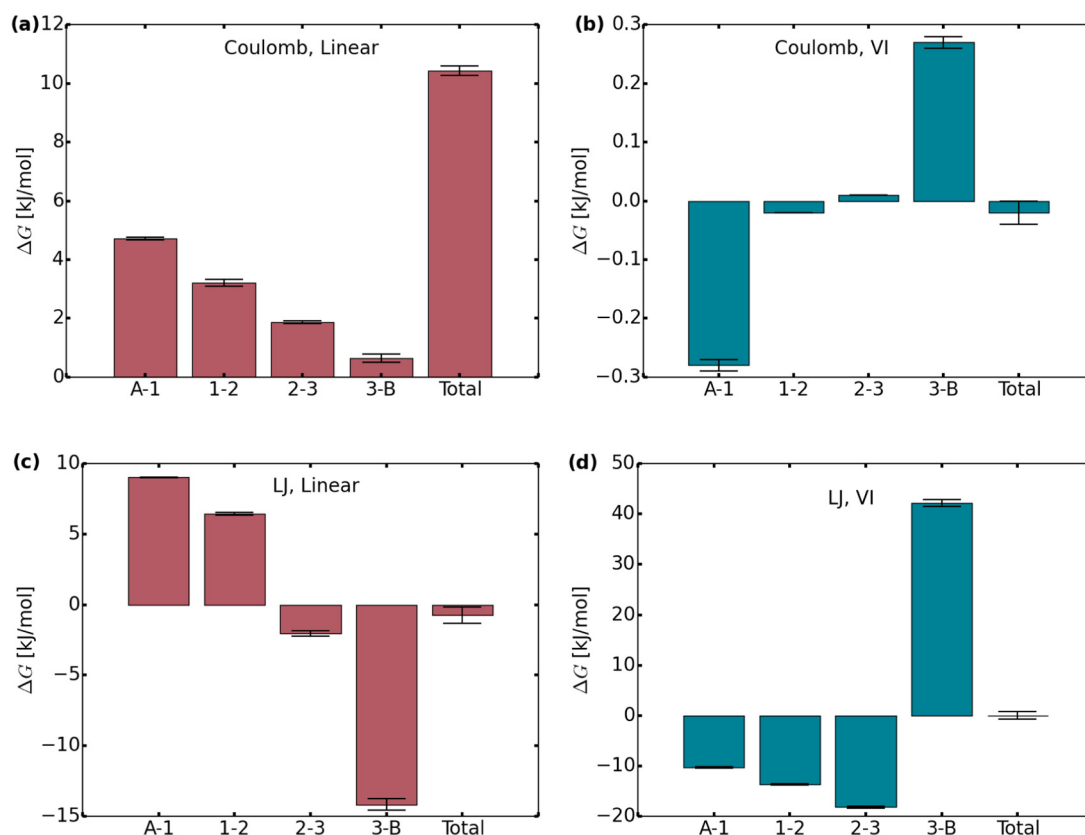

Fig. 3. Free energy differences along intermediate states between *A* (coupled state) and *B* (decoupled state). The bars show the differences between the states denoted below. The conventional linear interpolation method, panels (a) and (c), is shown in red, whereas VI is shown in blue (panels (b) and (d)). Coulomb interactions were decoupled first (with LJ interactions still turned on), LJ interactions second (Coulomb interactions switched off).

Gromacs-VI calculates the intermediate Hamiltonian based on,

$$
H_{VI}^{\lambda}(\mathbf{x}, \lambda) = (1 - \lambda) \underbrace{\sum_{\substack{interaction \\ type \ k}} \dots \sum_{\substack{particles \\ i,j}} H_A^k(\mathbf{x}_{i,j}, \lambda)}_{H_A^{\lambda}(\mathbf{x}, \lambda)}
$$
$$
+ \lambda \underbrace{\sum_{\substack{interaction \\ type \ k}} \dots \sum_{\substack{particles \\ i,j}} H_B^k(\mathbf{x}_{i,j}, \lambda)}_{H_B^{\lambda}(\mathbf{x}, \lambda)} , \tag{24}
$$

and likewise, for the forces and λ derivatives. Setting the seed to a fixed value such as,

```
ld-seed    =    1
```

it can be validated that all energies required for the free energy calculation that are stored in the `dhdl.xvg` file match between the implementation of the VI and the conventional sequence.

4.1. Solvation free energy calculation

As an example case, the solvation free energy of nitrocyclohexane in water was calculated (structure shown in Fig. 2). All input parameter files, topologies and starting structures have been provided with the program files available in the CPC Library and in our code repository (see code and data availability). Briefly, the topologies of the solvation toolkit package [25] were used, which had been created with the Generalized AMBER Force Field [26] and the TIP3P water model [27]. Upon energy minimization using the steepest decent algorithm, 2 ns NVT (constant volume and temperature) and 4 ns NPT (constant pressure and temperature) equilibration were conducted, followed by 100 ns production runs, using a time step of 2 fs. All simulations were conducted at a temperature of 298.15 K and pressure of 1.01325 bar. Temperature coupling was conducted via the stochastic dynamics integrator with $\tau_T = 1$ ps and pressure coupling using the Parrinello-Rahman [28] barostat with $\tau_p = 1$ ps and a compressibility of $4.5 \cdot 10^{-5}$ bar^{-1}. Long-range interactions were accounted for with the Particle-Mesh-Ewald [29] method with a Coulomb cutoff of 1.2 nm. For van-der-Waals interactions a cutoff of 1.2 nm was used with the potential switch starting at 1 nm. Hydrogen bonds were constrained using the LINCS algorithm [30].

To assess whether the VI implementation yields accurate results consistent with the ones from conventional intermediates, first, through extensive sampling with 101 states (i.e., λ steps of 0.01), a reference value value of (9.85 ± 0.02) kJ/mol was obtained. It can be divided into (10.46 ± 0.01) kJ/mol electrostatic, and (-0.61 ± 0.02) kJ/mol LJ contributions. Next, a set of simulations with 5 states, i.e., λ steps of 0.25, were conducted. In all reported cases, the free energy difference between adjacent states was calculated using BAR, i.e., Eq. (4).

The distribution of the free energy estimates between the different states is shown for Coulomb and LJ interactions in Fig. 3 and differs considerably between the two methods. The bars denote the free energy difference between the states denoted at the bottom. Again, *A* represents the coupled, and *B* the decoupled state. The plots shown for VI were created based on the runs where *C* was set to the respective reference value, and, as such, sum up to about zero. When decoupling Coulomb interactions with a conventional linear interpolation method, shown in panel (a), the largest differences between the states occur in the first steps and gradually decreases. For VI (b), the free energy path along the intermediates has be become very small (note the differing unis on the axis). In contrast, for LJ interactions, the
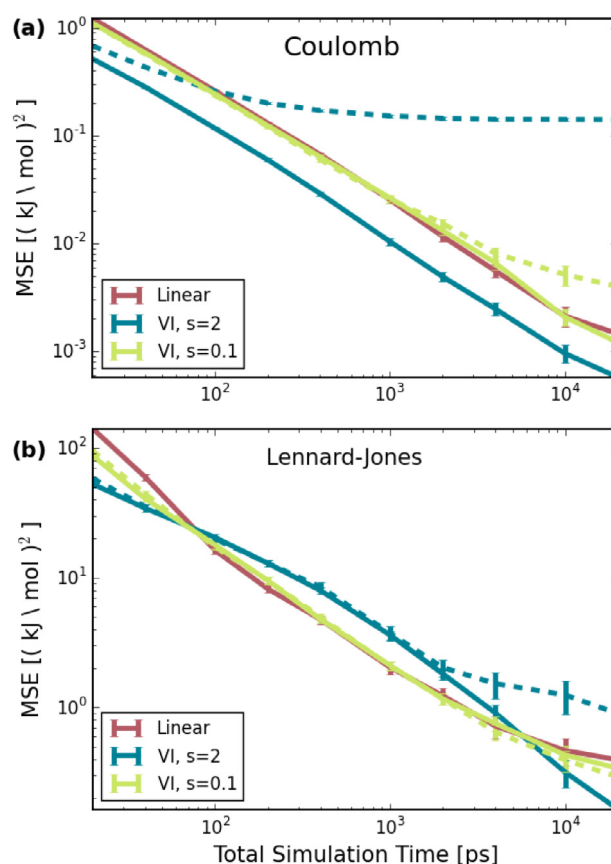

Fig. 4. MSEs as a function of simulation time for decoupling (a) Coulomb and (b) Lennard-Jones interactions. The red line indicates the use of the conventional linear interpolation method, the blue and green line the VI approach, Eq. (14), using two different *s* values. The solid line indicate the MSEs that were obtained by using an exact initial guess, whereas a guess of 1 kJ/mol is indicated by the dashed lines.

differences for VI (d) become larger than for the linear interpolation (c). The reason is, most likely, that the differences in the contributions from the attractive and the repulsive part of the LJ potential do not cancel for all intermediates.

To compare the accuracy of both methods, Fig. 4 shows the MSEs with total simulation time, distributed equally over all five states. The MSEs were obtained by dividing the trajectories of the production runs into smaller ones, and comparing the resulting free energy difference to the reference value. For VI, two different smoothing values were considered (blue and green lines), as well as an exact initial estimate (solid line) and one that is 1 kJ/mol too low (dashed lines).

For electrostatic interactions, the MSEs in Fig. 4(a) are significantly better for VI with $s = 2$ and an estimate close to the exact one than the MSE obtained with linear intermediates, thereby validating the result of Ref. [13]. However, in this case the MSEs are quite sensitive to the initial guess. For Lennard-Jones interactions, Fig. 4(b), VI and linear intermediates yield similar MSEs, but the VI estimates are less sensitive to the initial guess. In both cases, the MSEs corresponding to VI with a smoothing factor of 0.1 are close to the linear ones and insensitive to the initial guess for most of the trajectory lengths in Fig. 4. As such, it is advantageous to start the iteration process with a smaller smoothing factor that is gradually increased with an improved estimate for *C*.

5. Summary

We have implemented the VI sequence of states into the GROMACS MD software package. For Coulomb interactions, our implementation yields significantly smaller MSEs and, in this sense, higher accuracy as compared to linearly interpolated intermediates. This result requires a sufficiently accurate initial estimate, which for the test cases presented here requires only a few percent of the overall simulation time. Furthermore, using the λ dependence of the end states added to VI, for LJ interactions, similar MSEs as for conventional soft-core approaches are achieved. Given the many stepwise improvements that eventually led to the accuracy of current soft-core protocols, the fact the VI approach achieves similar accuracy already in the first attempt suggests that future refinements, e.g., of the lambda dependency on the end states, will push the accuracy even further.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and data availability

The source code is available at https://www.mpibpc.mpg.de/gromacs-vi-extension or https://gitlab.gwdg.de/martin.reinhardt/gromacs-vi-extension. Documentation, topologies and input parameter files of the above test cases are also available on the website and the repository. In the gitlab repository, all changes with respect to the official underlying GROMACS code can be retraced.

As installation is identical to that of GROMACS 2020, refer to http://manual.gromacs.org/documentation/2020/install-guide/index.html for detailed instructions.

Acknowledgments

References

[1] J. Konc, S. Lešnik, D. Janežič, J. Cheminformatics 7 (1) (2015) 48, http://dx.doi.org/10.1186/s13321-015-0096-0, URL https://jcheminf.biomedcentral.com/articles/10.1186/s13321-015-0096-0.
[2] C.D. Christ, T. Fox, J. Chem. Inf. Model. 54 (1) (2014) 108–120, http://dx.doi.org/10.1021/ci4004199, URL http://pubs.acs.org/doi/10.1021/ci4004199.
[3] K.A. Armacost, S. Riniker, Z. Cournia, J. Chem. Inf. Model. 60 (1) (2020) 1–5, http://dx.doi.org/10.1021/acs.jcim.9b01174, URL https://pubs.acs.org/doi/10.1021/acs.jcim.9b01174.
[4] J.M. Rickman, R. LeSar, Annu. Rev. Mater. Res. 32 (1) (2002) 195–217, http://dx.doi.org/10.1146/annurev.matsci.32.111901.153708, URL http://www.annualreviews.org/doi/10.1146/annurev.matsci.32.111901.153708.
[5] G.G. Vogiatzis, L.C. van Breemen, D.N. Theodorou, M. Hütter, Comput. Phys. Comm. 249 (2020) 107008, http://dx.doi.org/10.1016/j.cpc.2019.107008, URL https://linkinghub.elsevier.com/retrieve/pii/S0010465519303431.
[6] T.D. Swinburne, M.-C. Marinica, Phys. Rev. Lett. 120 (13) (2018) 135503, http://dx.doi.org/10.1103/PhysRevLett.120.135503, URL https://doi.org/10.1103/PhysRevLett.120.135503https://link.aps.org/doi/10.1103/PhysRevLett.120.135503.
[7] R.W. Zwanzig, J. Chem. Phys. (ISSN: 0021-9606) 22 (8) (1954) 1420–1426, http://dx.doi.org/10.1063/1.1740409, URL http://aip.scitation.org/doi/10.1063/1.1740409.
[8] M.R. Shirts, J.D. Chodera, J. Chem. Phys. 129 (12) (2008) 124105, http://dx.doi.org/10.1063/1.2978177, URL http://aip.scitation.org/doi/10.1063/1.2978177.
[9] T.C. Beutler, A.E. Mark, R.C. van Schaik, P.R. Gerber, W.F. van Gunsteren, Chem. Phys. Lett. 222 (6) (1994) 529–539, http://dx.doi.org/10.1016/0009-2614(94)00397-1, URL https://linkinghub.elsevier.com/retrieve/pii/0009261494003971.
[10] M. Zacharias, T.P. Straatsma, J.A. McCammon, J. Chem. Phys. 100 (12) (1994) 9025–9031, http://dx.doi.org/10.1063/1.466707, URL http://aip.scitation.org/doi/10.1063/1.466707.
[11] T. Steinbrecher, D.L. Mobley, D.A. Case, J. Chem. Phys. 127 (21) (2007) 214108, http://dx.doi.org/10.1063/1.2799191, URL http://aip.scitation.org/doi/10.1063/1.2799191.
[12] J.G. Kirkwood, J. Chem. Phys. 3 (5) (1935) 300–313, http://dx.doi.org/10.1063/1.1749657, URL http://aip.scitation.org/doi/10.1063/1.1749657.
[13] M. Reinhardt, H. Grubmüller, J. Chem. Theory Comput. 16 (6) (2020) 3504–3512, http://dx.doi.org/10.1021/acs.jctc.0c00106, URL https://pubs.acs.org/doi/10.1021/acs.jctc.0c00106.
[14] M. Reinhardt, H. Grubmüller, Phys. Rev. E 102 (4) (2020) 043312, http://dx.doi.org/10.1103/PhysRevE.102.043312, URL https://link.aps.org/doi/10.1103/PhysRevE.102.043312.
[15] A. Gelman, X.-l. Meng, Statist. Sci. 13 (2) (1998) 163–185, http://dx.doi.org/10.1214/ss/1028905934, URL papers2://publication/uuid/34B8C3F7-A3E8-4BF9-AB3C-E8A1B883ED7D http://projecteuclid.org/euclid.ss/1028905934.
[16] A. Blondel, J. Comput. Chem. 25 (7) (2004) 985–993, http://dx.doi.org/10.1002/jcc.20025, URL http://aip.scitation.org/doi/10.1063/1.1678245http://doi.wiley.com/10.1002/jcc.20025.
[17] T.T. Pham, M.R. Shirts, J. Chem. Phys. 136 (12) (2012) 124120, http://dx.doi.org/10.1063/1.3697833, URL http://aip.scitation.org/doi/10.1063/1.3697833.
[18] C.D. Christ, W.F. van Gunsteren, J. Chem. Phys. 126 (18) (2007) 184110, http://dx.doi.org/10.1063/1.2730508, URL http://aip.scitation.org/doi/10.1063/1.2730508.
[19] C.D. Christ, W.F. Van Gunsteren, J. Chem. Phys. 128 (17) (2008) http://dx.doi.org/10.1063/1.2913050.
[20] J.W. Perthold, C. Oostenbrink, J. Phys. Chem. B 122 (19) (2018) 5030–5037, http://dx.doi.org/10.1021/acs.jpcb.8b02725, URL https://pubs.acs.org/doi/10.1021/acs.jpcb.8b02725.
[21] J.W. Perthold, D. Petrov, C. Oostenbrink, J. Chem. Inf. Model. (2020) http://dx.doi.org/10.1021/acs.jcim.0c00456, acs.jcim.0c00456, URL https://pubs.acs.org/doi/10.1021/acs.jcim.0c00456.
[22] M.J. Abraham, T. Murtola, R. Schulz, S. Páll, J.C. Smith, B. Hess, E. Lindahl, SoftwareX 1–2 (2015) 19–25, http://dx.doi.org/10.1016/j.softx.2015.06.001, URL https://linkinghub.elsevier.com/retrieve/pii/S2352711015000059.
[23] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M.R. Shirts, J.C. Smith, P.M. Kasson, D. van der Spoel, B. Hess, E. Lindahl, Bioinformatics 29 (7) (2013) 845–854, http://dx.doi.org/10.1093/bioinformatics/btt055, URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btt055.
[24] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A.E. Mark, H.J.C. Berendsen, J. Comput. Chem. 26 (16) (2005) 1701–1718, http://dx.doi.org/10.1002/jcc.20291, URL http://doi.wiley.com/10.1002/jcc.20291.
[25] C.C. Bannan, G. Calabró, D.Y. Kyu, D.L. Mobley, J. Chem. Theory Comput. 12 (8) (2016) 4015–4024, http://dx.doi.org/10.1021/acs.jctc.6b00449.
[26] J. Wang, R.M. Wolf, J.W. Caldwell, P.A. Kollman, D.A. Case, J. Comput. Chem. 25 (9) (2004) 1157–1174, http://dx.doi.org/10.1002/jcc.20035, URL http://doi.wiley.com/10.1002/jcc.20035.
[27] W.L. Jorgensen, J. Chandrasekhar, J.D. Madura, R.W. Impey, M.L. Klein, J. Chem. Phys. 79 (2) (1983) 926, http://dx.doi.org/10.1063/1.445869, URL http://scitation.aip.org/content/aip/journal/jcp/79/2/10.1063/1.445869.
[28] M. Parrinello, A. Rahman, Phys. Rev. Lett. 45 (14) (1980) 1196–1199, http://dx.doi.org/10.1103/PhysRevLett.45.1196, URL https://link.aps.org/doi/10.1103/PhysRevLett.45.1196.
[29] T. Darden, D. York, L. Pedersen, J. Chem. Phys. 98 (12) (1993) 10089–10092, http://dx.doi.org/10.1063/1.464397, URL http://scitation.aip.org/content/aip/journal/jcp/98/12/10.1063/1.464397http://aip.scitation.org/doi/10.1063/1.464397.
[30] B. Hess, H. Bekker, H.J.C. Berendsen, J.G.E.M. Fraaije, J. Comput. Chem. 18 (12) (1997) 1463–1472, http://dx.doi.org/10.1002/(SICI)1096-987X(199709)18:12<1463::AID-JCC4>3.0.CO;2-H, URL https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1096-987X(199709)18:12%3C1463::AID-JCC4%3E3.0.CO;2-H.