JCTC Journal of Chemical Theory and Computation

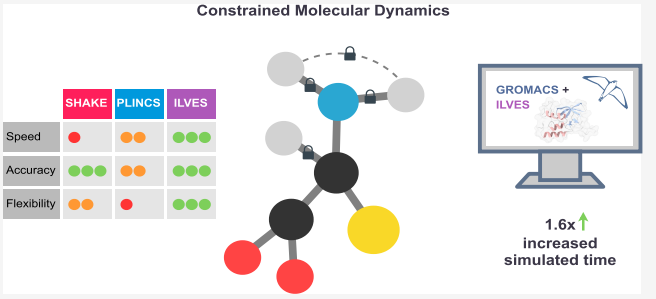
pubs.acs.org/JCTC Article

ILVES: Accurate and Efficient Bond Length and Angle Constraints in Molecular Dynamics

Lorién López-Villellas,* Carl Christian Kjelgaard Mikkelsen, Juan José Galano-Frutos, Santiago Marco-Sola, Jesús Alastruey-Benedé, Pablo Ibáñez, Pablo Echenique, Miquel Moretó, Maria Cristina De Rosa, and Pablo García-Risueño*

Cite This: https://doi.org/10.1021/acs.jctc.5c01376

ABSTRACT: All-atom, force field-based molecular dynamics simulations are essential tools in computational chemistry, enabling the prediction and analysis of biomolecular systems with atomic-level resolution. However, as system sizes and simulation time scales increase, so does the associated computational cost. To extend simulated time using the same resources, a common strategy is to constrain the fastest degrees of freedom, such as bond lengths, allowing for larger integration time steps without compromising accuracy. The de facto state-of-the-art algorithms for this purpose—SHAKE, LINCS, and P-LINCS—are integrated into most molecular dynamics packages and widely adopted across the field. Despite their impact, these methods exhibit limitations: all converge slowly when high numerical accuracy is required, and the LINCS and P-LINCS algorithms cannot handle general angular constraints, limiting further increases in time step. In this article, we introduce ILVES, a family of parallel algorithms that converge so rapidly that it is now practical to solve bond length and associated angular constraint equations as accurately as the hardware will allow. We have integrated ILVES into GROMACS, and our analysis demonstrates that it is superior to the state-of-the-art when constraining bond lengths. Due to its better convergence properties, we also show that if the time step is increased up to 3.5 fs by enforcing angular constraints, ILVES enables a 1.65× increase in simulated time using the same computational resources and wall-clock time, an outcome unattainable with current methods. This advance can significantly reduce the computational cost of most all-atom molecular dynamics simulations while improving their accuracy and extending access to larger systems and longer time scales.



INTRODUCTION

Molecular dynamics simulations (MD)[1,2] have greatly impacted a wide range of fields in science and technology.[3,4] They are of special importance in chemistry and medicine, with applications including the design of drugs and catalysts,[5−9] e.g., helping to understand interaction- or mutation-driven biological processes.[10−12] One of the biggest advantages of MD-based approaches is that they provide information on the simulated systems at the atomic level (positions, velocities, forces), which enables the study of phenomena whose analysis in a laboratory is often not feasible or affordable.[13,14] Thanks to recent advances, particularly in artificial intelligence, the impact of MD is set to grow. For example, the prediction of protein structures through AlphaFold[15] is already boosting massive-scale analysis of the behavior of proteins and their interactions for a variety of relevant areas.[16−22]

It is commonly accepted that, to be reliable, the discrete integration of the equations of motion in molecular dynamics must include at least five steps per vibration period of every degree of freedom (e.g., bond lengths, bond angles, or dihedral angles). This sets an upper limit for the value of the *time step*, i.e., for the separation between consecutive simulated times. Since the calculation at every time point requires a certain number of arithmetic operations, the size of the time step limits the total real-time that can be simulated using a given amount of computational resources. Due to this, in MD simulations it is customary to constrain some of the fastest internal degrees of freedom to fixed values. If the physical model and the resulting dynamics and thermodynamics are not distorted by doing so, the removal of the shortest vibrational periods allows an increase in the time step, thus reaching longer total times with the same computational effort.

Taking into account the well-known hierarchy that organizes vibrational periods in proteins and other biological mole-

Received: August 18, 2025
Revised: August 25, 2025
Accepted: August 26, 2025

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

cules,[23] the imposition of constraints begins with the fastest degrees of freedom and proceeds gradually to slower ones, thus allowing to increase the time step at each stage. It is very common in production MD simulations to constrain all bond lengths or those involving a hydrogen atom to increase the time step up to 2 fs. However, the situation around constraining bond angles is more heterogeneous in the literature. Although it is commonly mentioned that imposing constraints on the bond angles of hydrogen atoms is the reasonable next stage for further increasing the time step, it is difficult to find actual production simulations that do so. Instead, a variety of techniques are used, such as dummy hydrogens,[24] redistribution of mass to make hydrogen atoms heavier and their vibrations slower,[24] the artificial enlargement of the angular vibrational constants in the force field[23] or the use of united atoms, i.e., the assimilation of the mass and the charge of hydrogen atoms into the heavy atoms to which they are bonded, thus effectively removing them from the model.[25] All these techniques are useful for increasing the time step, but they introduce alterations to the model that are not justifiable a priori from physical or chemical considerations. The actual constraining of hydrogen bond angles—which is physically justifiable if we accept that this vibrational degree of freedom can be modeled as a quantum harmonic oscillator at its ground state—is performed only in a selected set of works[26−30] and not without difficulties. This has been done, for example, using internal coordinates to integrate the equations of motion instead of Cartesian coordinates, which introduces an important computational overhead.[27] Hydrogen bond angles have been constrained in production simulations using (P-)LINCS, but the bond lengths of heavy atoms had to remain unconstrained due to (P-)LINCS' convergence problems.[31] Finally, successful simulations with the GROMOS package have been reported using a modified version of the SHAKE algorithm to handle angles.[29] However, no in-depth assessment of the computational cost is provided, and convergence difficulties appear in the study when a small set of new constraints is added.[29]

In this work, we present ILVES-M and ILVES-F, two parallel algorithms that solve the same system of differential-algebraic equations as SHAKE, but the constraint equations are solved using either Newton's method or a quasi-Newton method rather than the nonlinear Gauss-Seidel method used by SHAKE. Our algorithms and software outperform the state-of-the-art algorithms, SHAKE[32] and (P-)LINCS.[33,34] A review of all relevant algorithms, SHAKE, (P)-LINCS, ILVES-M, and ILVES-F, can be found in Section 2 of the Supporting Information. In particular, we show that, in most tests involving bond length constraints, the ILVES algorithms converge so rapidly that solving the constraint equations with high accuracy is not only possible but eminently practical. Most importantly, by leveraging the existing GROMACS framework, we show that ILVES-M and ILVES-F can also constrain bond angles in parallel with low computational overhead, in contrast to SHAKE and (P-)LINCS. Our analysis shows that by constraining the bond angles of hydrogen atoms and increasing the time step to 3.5 fs, ILVES enables a 1.65× increase in simulated time using the same computational resources and wall-clock time as a simulation with the default 2 fs time step. These results establish, for the first time as far as we are aware, that constraining hydrogen bond angles enables a substantial increase in simulation throughput. The ILVES-M and ILVES-F code, integrated into GROMACS, is publicly available at https://github.com/LorienLV/_PAPER_ILVES.

LIMITATIONS OF THE STATE-OF-THE-ART CONSTRAINT SOLVERS

SHAKE and (P-)LINCS are decades-old algorithms. SHAKE[32] is nearly 50 years old, while LINCS[33] was presented in 1997 and P-LINCS[34] appeared in 2008. Though their contribution to science has been tremendous, they have specific limitations that we seek to address. The constraint solver in the original SHAKE algorithm converges slowly and is not considered a good candidate for parallelization.[35] Parallel versions of SHAKE[35,36] have not been widely used, and the implementation of SHAKE in GROMACS is sequential. The GROMACS library for molecular simulation is so widely used that we have chosen it to serve as a baseline for our analysis. The successful application of (P-)LINCS hinges on the convergence of a specific infinite series and this condition can be violated in the context of coupled angular constraints (see the Supporting Information and the original LINCS paper[33]), and the use of (P-)LINCS for this purpose is actively discouraged in the GROMACS manual itself.[37] An interesting example can be found in the paper[38] where the issues were so severe that LINCS had to be abandoned in favor of SHAKE.

In general, SHAKE, LINCS, and P-LINCS are rarely used to solve the constraint equations as accurately as the hardware will allow, as this goal can only be achieved using significant time and computational resources.[39,40] Superficially, this issue might appear insignificant, as there are many other sources of error in a simulation of molecular dynamics. However, there are cases where the error introduced in the constraints phase can result in severe distortions of the simulated system's physics.

NEED TO SOLVE THE CONSTRAINT EQUATIONS ACCURATELY

In a recent study,[40] we demonstrated that solving constraints inaccurately introduces distortions that can make the simulation unreliable. Insufficient accuracy when solving the constraints is equivalent to applying undesired, spurious, and random external forces,[40] which generates a non-negligible drift in the energy of the simulated system that consequently ruins the trustworthiness of simulations in the microcanonical (NVE) ensemble. This has led several studies to state that constraint equations must be solved *down to the limit of computational arithmetic/machine precision*.[41,42] Simulations with a thermostat (NVT, NPT ensembles) also present such undesired energetic drifts, which contribute to making the conserved quantity (also called *conserved energy*) of the thermostat (e.g., Nosé-Hoover, V-rescale) become non-conserved. Due to this, there is no guarantee that the equations of the thermostat are satisfactorily solved, hence there is no guarantee that the simulation corresponds to the sought ensemble, which makes its reliability drop.[43,44] Moreover, the drift introduced by the inaccurate solving of the constraints distorts the time (τ_T) for reaching the sought temperature (T). This can be observed in the V-rescale thermostat,[44] which calculates a rescaling factor for the velocities that, on average, is expected to make the temperature of the system approximately equal to the desired temperature T after a simulated time τ_T (being τ_T an input parameter of the simulation). However, due to the inaccurate constraint solving,

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

an additional amount of energy is injected into, or extracted from, the system, which makes the average time for reaching T deviate from τ_T in an unknown manner. In addition, imposing constraints inaccurately systematically misestimates bond lengths and makes them randomly change their values in an irregular manner. Moreover, artifactual regimes arise as periods where the averages of the lengths of the bonds differ from the values set by constraints, which alternate with periods where the bond lengths remain nearly unchanged.[40]

In GROMACS, the default SHAKE tolerance (`shake-tol`, defined as the maximum relative error allowed when solving constraints) is 10^{-4}. There exists no such demanded tolerance for P-LINCS, which has been said to cause unphysical dynamics and temperatures of thousands of Kelvin due to fast rotation of NH_3 groups.[45] Nevertheless, it is generally assumed that the average accuracy of P-LINCS with the default GROMACS parameters is typically similar to SHAKE's. Such default settings lead to the non-negligible distorting effects on energy drifts and bond lengths mentioned above; in contrast, solving the constraints more accurately strongly dampens these undesired effects.[40] Other research works have also found non-negligible distorting effects due to inaccurate constraint solving: ref 39. stressed that GROMACS' default parameters lead to nonconverged results and make temperatures of the simulated system unreliable, which is fixed if constraints are accurately solved. Other research indicates that inaccuracy in constraints can lead to wrong densities[46] or to collective motion artifacts, like spurious phase transitions from liquid to an icy state.[47] In Section 4 of the Supporting Information, we shall argue further in favor of solving the constraint equations as accurately as the hardware will allow.

The array of inconveniences due to inaccurate constraint solving can be largely mitigated if the constraint forces are calculated with the largest possible accuracy (for the chosen numerical precision) instead of the default values in GROMACS of 10^{-4}—for SHAKE—or undetermined—for LINCS—. Nevertheless, doing so has been precluded to date, most likely due to numerical complexity issues.

■ ILVES ALGORITHMS

ILVES is a family of algorithms for imposing constraints in the context of molecular dynamics. The ILVES algorithms compute discrete approximations of the solution to the same system of differential-algebraic equations as the SHAKE algorithm. However, whereas SHAKE relies on the nonlinear Gauss-Seidel method, which converges locally and linearly, the ILVES algorithms are based on Newton's method combined with direct solvers, resulting in drastically faster convergence rates. In general, applying direct solvers to linear systems requires $O(n^3)$ floating-point operations, being n the number of equations (which is equal to the number of constraints in our case[48]). Nonetheless, the particular structure of the linear systems that arise when applying constraints in MD is directly tied to the linear and sparse topology of molecular structures, so direct solvers can be applied in $O(n)$ time for general molecules.[48,49] The ILVES algorithms exploit this property to dramatically accelerate convergence relative to SHAKE.

In this paper, we present two algorithms, ILVES-M ("main") and ILVES-F ("fast"), both of which leverage distributed-memory parallelism, shared-memory parallelism, and SIMD vectorization. ILVES-M solves the same system of differential-algebraic equations as SHAKE but employs Newton's method and a direct solver. To exploit shared-memory parallelism, ILVES-M uses a custom thread-parallel LU factorization based on the Schur complement method.[50] For distributed-memory parallelism, it extends this thread-parallel LU factorization with the Overlapping Partitioning Method (OPM).[51] Consequently, when executed across multiple domains, ILVES-M behaves as a quasi-Newton method. However, its convergence remains extremely fast, typically requiring very few (usually zero) additional iterations compared to single-domain execution.

ILVES-F is a variant of ILVES-M that reduces the computational cost by using a fixed symmetric approximation of the coordinate matrix.[52] The symmetry of this matrix allows for replacing LU factorization with $LDLT$ factorization, which improves efficiency. Moreover, since the $LDLT$ factorization needs only be computed once per time step, the total computational cost is nearly halved. Due to its symmetric approximation, ILVES-F behaves as a quasi-Newton method even in shared-memory executions. Nonetheless, its convergence is exceptionally fast,[53,54] and it delivers better performance than ILVES-M in most scenarios.

In a previous article,[40] we introduced ILVES-PC, a proof-of-concept implementation applying direct solvers and Newton's method to calculate constraint forces in biological molecules, specifically peptides and proteins. For completeness, we include ILVES-PC in our performance analysis in this paper.

We present a detailed description of the mathematical foundations of ILVES, as well as implementation details of ILVES-M and ILVES-F, in Sections 2 and 3 of the Supporting Information.

■ RESULTS

We conducted an extensive set of simulations to assess the efficiency and reliability of ILVES, covering five representative systems: two solvated proteins (barnase, referred to as the BARN system, and the COVID-19 main protease, referred to as the COVID system), a solvated protein−DNA complex (the DNAP system), a system of 2000 benzene molecules (the BENZ system), and a tetrameric protein embedded in a lipid bilayer (the LIPID system). Full details of these systems, along with the procedures used for their preparation and simulation are provided in Sections 5 and 6 of the Supporting Information. Our reliability analysis—based on the calculation of observable quantities—as well as complementary performance results can also be found in the Supporting Information (Sections 7 and 8). Below, we summarize the outcome of our performance study, comparing ILVES-M, ILVES-F, and ILVES-PC with state-of-the-art constraint solvers. The time spent in the initialization of the solvers can be high in distributed memory simulations. For this reason, the execution times in our analysis include both initialization and processing for all solvers—except ILVES-PC, which was released as a proof of concept without distributed memory support and optimized initialization. Additionally, to ensure a fair comparison, we developed a modified version of P-LINCS that guarantees constraints are satisfied within a given tolerance, as SHAKE does; we refer to this variant as MP-LINCS. This was accomplished by repeating P-LINCS' correction phase, controlled by the `lincs-iter` parameter in the original implementation until the desired maximum relative error in solving the constraints is met. Although this modification introduces additional synchronization points in parallel executions, potentially affecting performance, it is

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

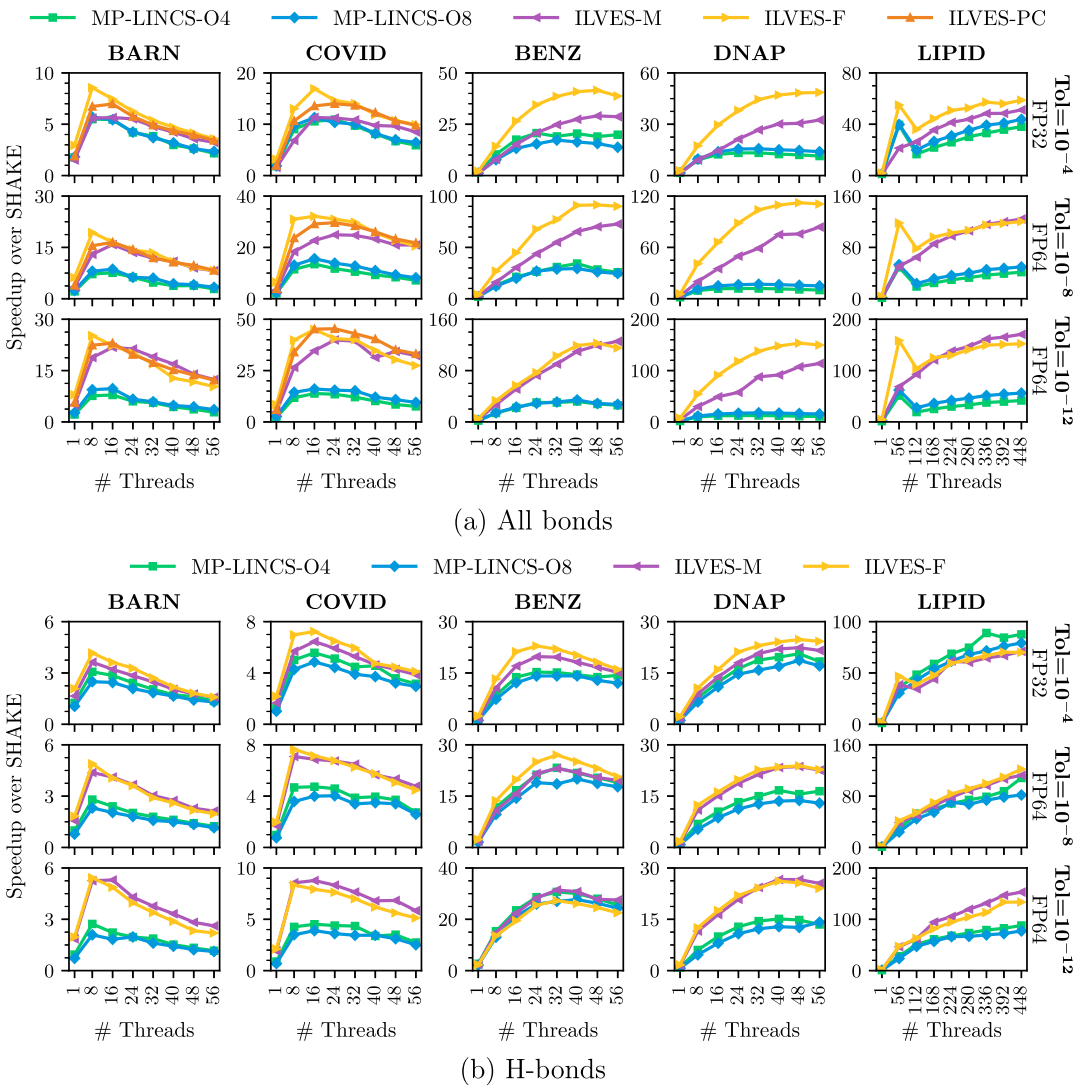


Figure 1. Multithread speedup over SHAKE of MP-LINCS, ILVES-M, ILVES-F, and ILVES-PC. The BARN, COVID, BENZ, and DNAP simulations are executed using a single task on a single chip of a computing node. The LIPID simulation is executed using up to 8 tasks (up to 4 nodes and 8 chips). MP-LINCS tested for `lincs-order = 4` and `lincs-order = 8`. (a) Constraints imposed on all bonds; (b) constraints imposed on H-bonds.

important to note that it is straightforward to modify ILVES to execute a fixed number of iterations without checking the tolerance, thereby eliminating the same synchronization points introduced in MP-LINCS. While GROMACS' implementation of SHAKE itself is not parallelized, it can still be used in parallel simulations without domain decomposition. In such cases, SHAKE runs on a single thread, while the rest of the simulation proceeds in parallel. This approach was used to obtain the results reported for SHAKE in parallel simulations.

The speed of the constraint solving is closely related to the minimum accuracy demanded. We thus considered three values of the referred tolerance (maximum allowed relative error for every constraint): Tol = 10^{-4} (which is the default in GROMACS), Tol = 10^{-8} and Tol = 10^{-12}. Simulations for Tol = 10^{-4} are performed using GROMACS compiled in single-precision mode (FP32), whereas simulations with stricter tolerances are conducted with GROMACS compiled in double-precision mode (FP64).

We have measured the performance of the algorithms in cases where constraints are imposed on either: (i) hydrogen bonds, (ii) all bonds, or (iii) all bonds together with certain angles of hydrogen atoms (specifically, H−X−H and X−O−H angles, where X represents a generic atomic species). The choice of constraint settings is generally determined—or at least recommended—by the force field used. CHARMM36[55] and CHARMM36m[56] support hydrogen bond constraints, while other force fields such as AMBER[57] and OPLS/AA[58] support constraints on hydrogen bonds and all bonds. Constraining angles remain uncommon, likely because none of the widely used constraint algorithms, SHAKE or (P-)LINCS, can satisfactorily handle coupled angle constraints.

Although solving constraints is sometimes assumed to require a relatively low fraction of the total execution time of simulations, some authors inform that it can be as high as 50 or 60%.[41,59] In our simulations constraining all bonds, SHAKE accounts for up to 92% of the total execution time, MP-LINCS up to 42%, and ILVES up to 16%. These percentages depend on the number of cores and are lower in H-bonds simulations, where SHAKE accounts for up to 60%, MP-LINCS up to 6%, and ILVES up to 5%. A detailed figure of the solvers' relative execution times is provided in Section 8 of the Supporting Information.

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

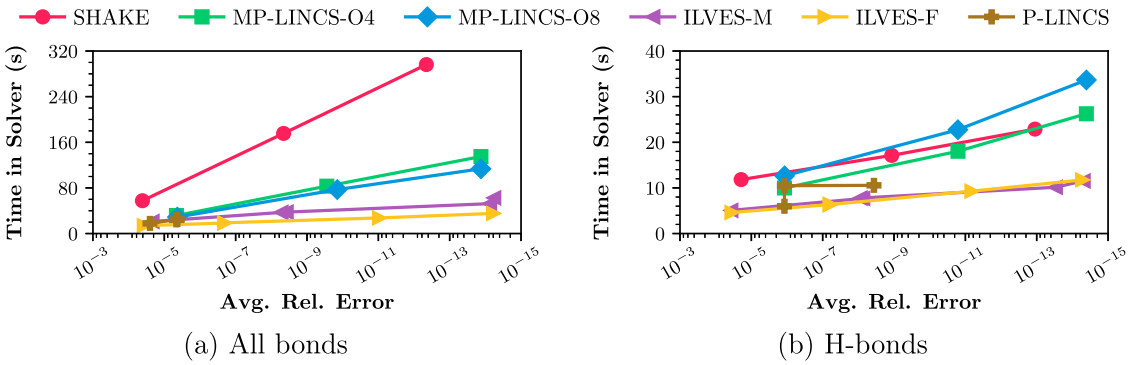


Figure 2. Execution time of the block for solving constraints as a function of the average relative error in satisfying them for different constraint solvers. (a) Constraints imposed on all bonds; (b) constraints imposed on H-bonds. Note that the *y*-axis is logarithmic and that the tolerance decreases from left to right.

Figure 1 presents the speedup over SHAKE for different numbers of threads and tasks when constraining all bonds and when constraining H-bonds. This metric is defined as the ratio of the execution time of SHAKE with a single thread to the execution time of the given solver using N threads. For the barnase (BARN), Covid-19 main protease (COVID), benzene (BENZ), and DNA−protein complex (DNAP), our simulations employed up to 56 threads and 1 task on a single Intel Xeon Platinum 8480+ chip. The larger (390 K atoms, 149 K ex water) lipid bilayer with proteins (LIPID) simulation was executed using up to 8 tasks and 56 threads per task, i.e., up to 4 nodes and 8 chips. The production stage of each of the simulations consisted of 50k steps of size 2 fs. In all-bonds simulations (Figure 1a), ILVES-M and ILVES-F achieve speedups over MP-LINCS across all simulations and tolerances, with a maximum of 158× over SHAKE and 14× over MP-LINCS. In H-bonds simulations (Figure 1b), MP-LINCS delivers better parallel performance than in all-bonds simulations, thus narrowing the performance gap between the solvers. In these simulations, MP-LINCS only surpasses ILVES-M and ILVES-F in the LIPID simulation at Tol = 10^{-4}. On the other hand, ILVES-F delivers better performance than MP-LINCS in the rest of the simulations for a maximum speedup of 134× over SHAKE and speedups over MP-LINCS up to 1.8×.

Even though the tolerance defines the maximum acceptable error, the rapid convergence of the ILVES algorithm often yields errors significantly below this threshold. This results in accuracy gains, providing a compelling reason to choose ILVES over MP-LINCS in simulations where their performance is similar. In Figure 2 we display the execution time required for imposing the constraints as a function of the average relative error, which is defined as the average for N_s steps and n constraints that follows: $\left(\frac{1}{2n\cdot N_s}\right)\cdot\left(\sum_{k=1}^{N_s}\sum_{i=1}^{n}|d_i^2-(\boldsymbol{q}_{a_i}(t_k)-\boldsymbol{q}_{b_i}(t_k))^2|/d_i^2\right)$, where $\boldsymbol{q}_{a_i}(t_k)$, $\boldsymbol{q}_{b_i}(t_k)$ are the positions of both atoms joined by the ith constraint after applying the constraint forces corresponding to the kth step, and d_i are the bond length constants. Every point of Figure 2 corresponds to a simulation performed with different parameters (values of the constraint tolerance for SHAKE; values of the constraint tolerance or number of iterations for ILVES-M and ILVES-F; values of the number of iterations—`lincs-iter`—and truncation of the Neumann series—`lincs-order`—for P-LINCS; values of the constraint tolerance and `lincs-order` for MP-LINCS). The results displayed in Figure 2 correspond to the Covid main protease (4697 constraints) simulated for 50K steps in a single core. If we compare algorithms that ensure that a minimum accuracy is satisfied, like MP-LINCS-O4 and ILVES-F, we observe that, for approximately the same execution time, the latter is far more accurate than the former. For example, for constraints on all bonds, Figure 2 displays a point for MP-LINCS-O4 whose execution time is 32 s and whose average relative error is $5\cdot10^{-6}$; it also displays a point for ILVES-F whose execution time is 27 s and whose average relative error is $9\cdot10^{-12}$. This feature also holds for constraints on hydrogen bonds: examples of points displayed in Figure 2 are (10 s, 10^{-6}) for MP-LINCS-O4, (9 s, $7\cdot10^{-12}$) for ILVES-F and (10 s, $3\cdot10^{-14}$) for ILVES-M. This example indicates that, for similar execution times, the ILVES algorithms are between 500,000 and 30,000,000 times more accurate than P-LINCS algorithms. Figure 2 indicates that the fast convergence of the ILVES methods makes much more accurate solutions possible requiring very low execution times, making it affordable to solve constraints near the limit of machine precision. We stress that increasing the accuracy of constraint solving is also desirable in simulations made with numerical single precision. In such cases, our tests indicate that the maximum enforceable tolerance is about Tol = 10^{-6} instead of the value Tol = 10^{-12} which corresponds to double precision.

The discussion presented in this section so far corresponds to the case of imposing constraints on just bond lengths, which limits the maximum time step to 2 fs. But, in addition to the possibility of achieving a higher degree of accuracy and computational savings that the ILVES family of algorithms provides for this very common set of constraints, its better convergence properties also allow us to cross a line unprecedented in the literature as far as we are aware. In what follows, we demonstrate that ILVES can be utilized to efficiently impose constraints on specific bond angles of hydrogen atoms, thereby enabling an increase in both the time step and the simulation throughput.

At present GROMACS offers time steps beyond 2 fs by applying several techniques such as *mass repartitioning* or *virtual sites*.[30,37] Mass repartitioning involves assigning hydrogen atom masses greater than 1 atomic mass unit, which is compensated by withdrawing part of the mass of heavy atoms. Virtual sites consist of determining the position of hydrogen atoms as a function of the position of three nearby heavy atoms, i.e., without applying forces to the hydrogen atoms. Although these approaches have been shown to produce

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

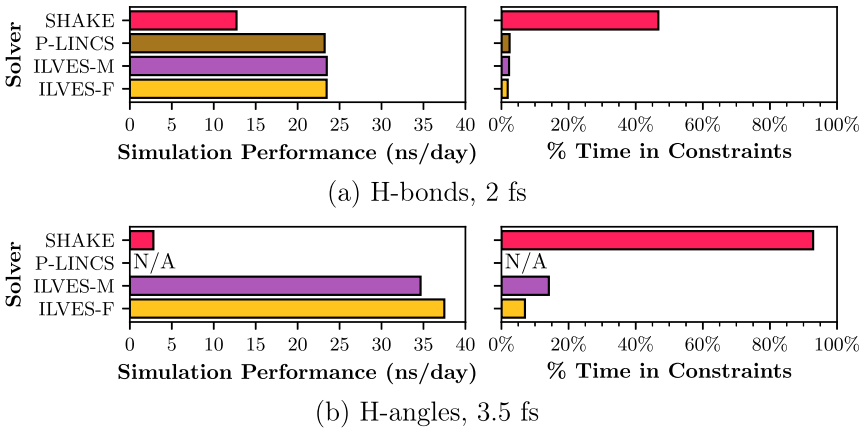


Figure 3. LIPID simulation performance (left) and percentage of the execution time spent on the constraint solver (right) using a 2 fs time step with `constraints = h-bonds` (a) and a 3.5 fs time step with `constraints = h-angles` (b), across four constraint solvers: SHAKE, P-LINCS, ILVES-M, and ILVES-F. The performance is reported in nanoseconds simulated per day, using all 56 cores of an Intel Xeon Platinum 8480+ chip. P-LINCS is not compatible with `constraints = h-angles` and is marked as N/A in the figure.

suitable results for some observables and thermodynamic quantities,[60] the additional assumptions that they introduce in the physical model can significantly alter some kinetic properties such as diffusion in lipid membranes or the typical time of protein−ligand binding.[61−63] By comparison, imposing constraints on H-angles is chemically and physically justifiable, as quantum harmonic oscillators resemble constraints more closely than classical harmonic oscillators. Literature indicates that the time step can be safely increased up to 4 fs by constraining all covalent bonds and the angles involving hydrogen atoms.[23,24,27,29] However, the option `constraints = h-angles` in Gromacs only imposes constraints on a subset of all the bond angles related to hydrogen atoms, namely those defined between two hydrogen atoms connected to the same heavy atom X in a H-X-H scheme and the angle between a hydrogen atom connected to an oxygen atom and the heavy atom X connected to the oxygen in a X−O-H scheme. This freezes some of the vibrations associated with the angular degrees of freedom of hydrogen atoms, but not all of them, and this is the reason why `constraints = h-angles` in Gromacs allows an increase of the time step to 3.5 fs but not to 4 fs. The implementation of full bond angle constraints will be a matter of future research.

Despite its availability, the `constraints = h-angles` option of Gromacs has not been successfully used in the literature to increase the time step, most likely due to the limitations of the state-of-the-art constraint solvers: (P-)LINCS is usually unable to impose constraints on coupled angles,[38] and SHAKE converges extremely slowly.[64] This is shown in Figure 3, in which we increase the time step of the LIPID simulation to 3.5 fs (this is the identified upper bound for stable simulations when using the `constraints = h-angles` setup). The referred figure reports the simulation performance in nanoseconds simulated per day and the percentage of execution time spent on the constraint solver using SHAKE, P-LINCS, ILVES-M, and ILVES-F, under two configurations: the default Gromacs settings (`constraints = h-bonds`, `ts = 2 fs`, `Tol` = 10^{-4}) and the new settings (`constraints = h-angles`, `ts = 3.5 fs`, `Tol` = 10^{-4}). The simulation was performed using all 56 cores of an Intel Xeon Platinum 8480+ processor and ran for 1.5 million steps. The results show that increasing the time step from 2 to 3.5 fs by introducing angle constraints causes SHAKE to dominate the simulation time, accounting for 93% of the total runtime and severely limiting overall performance. In addition, P-LINCS does not work with `constraints = h-angles` (marked as N/A in the figure). In contrast, ILVES enables the simulation to run significantly faster, increasing performance from 23 ns/day to 38 ns/day, which translates to a 1.65× improvement. Furthermore, ILVES accounts for a small fraction of the total runtime: 14% for ILVES-M and 7% for ILVES-F.

The previous results demonstrate that ILVES paves the way for a new approach to increasing the time step without relying on potentially unphysical approximations, such as mass repartitioning. Nevertheless, further research is required. Currently, increasing the time step beyond 3.5 fs is not possible within Gromacs, as it lacks support for constraining a larger set of angles involving hydrogen atoms. Additionally, existing force fields have not been parametrized for use with angle constraints. Further investigation is, therefore, necessary to establish how to correctly impose angle constraints—and potentially dihedral constraints—so that ILVES can enable time steps well above 4 fs and to evaluate how introducing such constraints would affect the physical accuracy and stability of molecular dynamics simulations.

CONCLUSIONS AND FUTURE WORK

In this work, we introduce novel parallel algorithms based on Newton's method and direct linear solvers, designed to impose constraints on molecular systems. These algorithms demonstrate significant improvements in both accuracy and efficiency compared to the current state-of-the-art methods when constraining bond lengths. Moreover, they enable the efficient constraining of additional degrees of freedom in parallel, establishing the foundation for increasing the time step of simulations. We show that when the time step is increased from 2 to 3.5 fs by constraining some hydrogen angles (leveraging Gromacs' existing framework for this task), P-LINCS does not work, and SHAKE dominates the total execution time of Gromacs, significantly degrading performance. In contrast, our solvers enable a 1.65× increase in simulated time using the same computational resources and wall-clock time.

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

In this article, we have shown that increasing the time step by applying angle constraints in combination with ILVES can yield substantial performance gains. Looking toward future research, we plan to investigate how to further and reliably extend the time step by constraining all hydrogen angles,[31] dihedral angles,[28] and other internal degrees of freedom. ILVES is expected to solve the equations associated with these additional constraints efficiently, thus enabling higher time steps in simulations with and without virtual sites,[30] which we also intend to explore. Further work will also focus on integrating these constraints into molecular dynamics packages and assessing their effects on simulation accuracy and stability. The current implementation of P-LINCS found in GROMACS supports GPU execution when synchronization between threads or nodes is not required, such as when constraining hydrogen bonds. We are developing GPU-accelerated versions of ILVES-M and ILVES-F for these same use cases, which we expect will enable higher accuracy without increasing execution time, consistent with the CPU-based results presented in this work. Further planned research will consist of optimizing ILVES for water molecules,[65] which—from preliminary tests—is expected to improve performance over the widely used SETTLE algorithm.[66]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.5c01376.

Overview of constraints solvers for molecular dynamics, mathematical background, description of the ILVES implementation, extended study of the importance of accuracy when solving constraints, experimental setup, reliability study, and extended performance analysis (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Lorién López-Villellas − *Departamento de Informática e Ingeniería de Sistemas/Aragón Institute for Engineering Research (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain;* orcid.org/0000-0002-1891-4359; Email: lorien.lopez@unizar.es

Pablo García-Risueño − *Instituto de Biocomputación y Física de Sistemas Complejos, Zaragoza 50018, Spain;* orcid.org/0000-0002-8142-9196; Email: risueno@unizar.es

Authors

Carl Christian Kjelgaard Mikkelsen − *Department of Computing Science, Umeå University, Umeå SE-90187, Sweden*

Juan José Galano-Frutos − *Instituto de Biocomputación y Física de Sistemas Complejos, Zaragoza 50018, Spain;* Present Address: Certest Biotec S.L., 50840, San Mateo de Gállego, Zaragoza, Spain; orcid.org/0000-0002-1896-7805

Santiago Marco-Sola − *Barcelona Supercomputing Center, Barcelona 08034, Spain; Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona 08034, Spain*

Jesús Alastruey-Benedé − *Departamento de Informática e Ingeniería de Sistemas/Aragón Institute for Engineering Research (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain;* orcid.org/0000-0003-4164-5078

Pablo Ibáñez − *Departamento de Informática e Ingeniería de Sistemas/Aragón Institute for Engineering Research (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain*

Pablo Echenique − *Instituto de Química Física Blas Cabrera (CSIC), Madrid 28006, Spain*

Miquel Moretó − *Barcelona Supercomputing Center, Barcelona 08034, Spain; Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona 08034, Spain*

Maria Cristina De Rosa − *Istituto di Scienze e Tecnologie Chimiche "Giulio Natta" (SCITEC) − National Research Council (CNR), Milan 20133, Italy;* orcid.org/0000-0002-9611-2490

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.5c01376

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank José Alejandre and Edgar Núñez (UAM-Iztapalapa, Mexico) for providing input files of organic solvents and Prof. Berk Hess for useful discussion. This work has been partially supported by the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 (contracts PID2022-136454NB-C22, PID2023-146193OB-I00, and PID2023-146511NB-I00), by the Generalitat de Catalunya (contract 2021-SGR-00763 and 2021-SGR-00574), by the Gobierno de Aragón (E45_20R T58_23R research groups), and by Lenovo-BSC Contract-Framework Contract (2020). Carl Christian Kjelgaard Mikkelsen is supported by eSSENCE, a collaborative e-Science programme funded by the Swedish Research Council within the framework of the strategic research areas designated by the Swedish Government. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

■ REFERENCES

(1) Leimkuhler, B.; Reich, S. Simulating Hamiltonian dynamics, 1st ed.; Monographs on Applied and Computational Mathematics Cambridge University Press: Cambridge, 2004.

(2) Karplus, M. Dynamics of Folded Proteins. *Abstracts of papers of the American Chemical Society* **1978**, *175*, 70.

(3) Karplus, M.; McCammon, J. A. Molecular dynamics simulations of biomolecules. *Nat. Struct. Biol.* **2002**, *9*, 646−652.

(4) Hospital, A.; Goñi, J. R.; Orozco, M.; Gelpí, J. L. Molecular dynamics simulations: advances and applications. *Advances and Applications in Bioinformatics and Chemistry* **2015**, *8*, 37−47.

(5) Shan, X.; Cai, Y.; Zhu, B.; Zhou, L.; Sun, X.; Xu, X.; Yin, Q.; Wang, D.; Li, Y. Rational strategies for improving the efficiency of design and discovery of nanomedicines. *Nat. Commun.* **2024**, *15*, 9990.

(6) Andreano, E. SARS-CoV-2 Escape a Highly Neutralizing COVID-19 Convalescent Plasma. *Proc. Natl. Acad. Sci. U. S. A.* **2021**, *118*, No. e2103154118.

(7) Bhardwaj, V. K.; Singh, R.; Sharma, J.; Rajendran, V.; Purohit, R.; Kumar, S. Identification of bioactive molecules from tea plant as SARS-CoV-2 main protease inhibitors. *J. Biomol. Struct. Dyn.* **2021**, *39*, 3449−3458.

(8) Yu, W.; Kumar, S.; Zhao, M.; Weber, D. J.; MacKerell, A. D., Jr High-Throughput Ligand Dissociation Kinetics Predictions Using Site

Identification by Ligand Competitive Saturation. *J. Chem. Theory Comput.* **2025**, *21*, 4964−4978.
(9) Lu, D.; Luo, D.; Zhang, Y.; Wang, B. A Robust Induced Fit Docking Approach with the Combination of the Hybrid All-Atom/United-Atom/Coarse-Grained Model and Simulated Annealing. *J. Chem. Theory Comput.* **2024**, *20*, 6414−6423.
(10) Galano-Frutos, J. J.; García-Cebollada, H.; Sancho, J. Molecular dynamics simulations for genetic interpretation in protein coding regions: where we are, where to go and when. *Briefings in Bioinformatics* **2021**, *22*, 3−19.
(11) Hollingsworth, S. A.; Dror, R. O. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99*, 1129−1143.
(12) Kim, C. N.; Shin, D.; Wang, A.; Nowakowski, T. J. Spatiotemporal molecular dynamics of the developing human thalamus. *Science* **2023**, *382*, No. eadf9941.
(13) Alonso, J. L.; Chass, G. A.; Csizmadia, I. G.; Echenique, P.; Tarancón, A. *Meeting on Fundamental Physics 'Alberto Galindo'*; Aula Documental de Investigación: Madrid, 2004.
(14) Galano-Frutos, J. J.; Sancho, J. Energy, water, and protein folding: A molecular dynamics-based quantitative inventory of molecular interactions and forces that make proteins stable. *Protein Sci.* **2024**, *33*, No. e4905.
(15) Jumper, J.; Evans, R.; Pritzel, A.; Green, T. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.
(16) Li, Y.; Arghittu, S. M.; Dietz, M. S.; Hella, G. J.; Haße, D.; Ferraris, D. M.; Freund, P.; Barth, H.-D.; Iamele, L.; de Jonge, H. Single-molecule imaging and molecular dynamics simulations reveal early activation of the MET receptor in cells. *Nat. Commun.* **2024**, *15*, 9486.
(17) Lehrer, S.; Rheinstein, P. H. Predicted Configuration and Stability of the ATAD2/SOX10 Complex Using Molecular Dynamics Simulations. *Cancer Diagnosis & Prognosis* **2023**, *3*, 398.
(18) Zheng, N.; Long, M.; Zhang, Z.; Zan, Q.; Osire, T.; Zhou, H.; Xia, X. Protein-Glutaminase engineering based on isothermal compressibility perturbation for enhanced modification of soy protein isolate. *J. Agric. Food Chem.* **2022**, *70*, 13969−13978.
(19) Osifalujo, E. A.; Rutkowski, B. N.; Satterwhite, L. R.; Betts, P. C.; Nkosi, A. K.; Froese, J. T. Production of novel Rieske dioxygenase metabolites enabled by enzyme engineering. *Catalysis Science & Technology* **2023**, *13*, 3784−3790.
(20) Meng, B.; Abdullahi, A.; Ferreira, I. A.; Goonawardane, N.; Saito, A.; Kimura, I.; Yamasoba, D.; Gerber, P. P.; Fatihi, S.; Rathore, S. Altered TMPRSS2 usage by SARS-CoV-2 Omicron impacts infectivity and fusogenicity. *Nature* **2022**, *603*, 706−714.
(21) Mosalaganti, S.; Obarska-Kosinska, A.; Siggel, M.; Taniguchi, R.; Turoňová, B.; Zimmerli, C. E.; Buczak, K.; Schmidt, F. H.; Margiotta, E.; Mackmull, M.-T AI-based structure prediction empowers integrative structural analysis of human nuclear pores. *Science* **2022**, *376*, No. eabm9506.
(22) Min, J.; Rong, X.; Zhang, J.; Su, R.; Wang, Y.; Qi, W. Computational design of peptide assemblies. *J. Chem. Theory Comput.* **2024**, *20*, 532−550.
(23) Stocker, U.; Juchli, D.; van Gunsteren, W. F. Increasing the time step and efficiency of molecular dynamics simulations: optimal solutions for equilibrium simulations or structure refinement of large biomolecules. *Mol. Simul.* **2003**, *29*, 123−138.
(24) Feenstra, K. A.; Hess, B.; Berendsen, H. J. C. Improving efficiency of large time-scale molecular dynamics simulations of hydrogen-rich Systems. *J. Comput. Chem.* **1999**, *20*, 786−798.
(25) Van Gunsteren, W. F.; Karplus, M. Effects of constraints on the dynamics of macromolecules. *Macromolecules* **1982**, *15*, 1528−1544.
(26) Gonnet, P.; Walther, J. H.; Koumoutsakos, P. Theta SHAKE: An extension to SHAKE for the explicit treatment of angular constraints. *Comput. Phys. Commun.* **2009**, *180*, 360−364.
(27) Mazur, A. Hierarchy of Fast Motions in Protein Dynamics. *J. Phys. Chem. B* **1998**, *102*, 473−479.
(28) Pechlaner, M.; Dorta, A. P.; Lin, Z.; Rusu, V. H.; van Gunsteren, W. F. A method to apply bond-angle constraints in molecular dynamics simulations. *J. Comput. Chem.* **2021**, *42*, 418−434.
(29) Pechlaner, M.; van Gunsteren, W. F. On the use of intramolecular distance and angle constraints to lengthen the time step in molecular and stochastic dynamics simulations of proteins. *Proteins: Struct., Funct., Bioinf.* **2022**, *90*, 543−559.
(30) Fábián, B.; Thallmair, S.; Hummer, G. Optimal bond constraint topology for molecular dynamics simulations of cholesterol. *J. Chem. Theory Comput.* **2023**, *19*, 1592−1601.
(31) Krieger, E.; Vriend, G. New ways to boost molecular dynamics simulations. *J. Comput. Chem.* **2015**, *36*, 996−1007.
(32) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comput. Phys.* **1977**, *23*, 327−341.
(33) Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. LINCS: A Linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18*, 1463−1472.
(34) Hess, B. P-LINCS: A parallel linear constraint solver for molecular simulation. *J. Chem. Theory Comput.* **2008**, *4*, 116−122.
(35) Elber, R.; Ruymgaart, A.; Hess, B. SHAKE parallelization. *European Physical Journal Special Topics* **2011**, *200*, 211−223.
(36) Weinbach, Y.; Elber, R. Revisiting and parallelizing SHAKE. *J. Comput. Phys.* **2005**, *209*, 193−206.
(37) Abraham, M. et al. *GROMACS 2024.1 Manual*, 2024,.
(38) Broadbent, R. J.; Spencer, J. S.; Mostofi, A. A.; Sutton, A. P. Accelerated simulations of aromatic polymers: application to polyether ether ketone (PEEK). *Mol. Phys.* **2014**, *112*, 2672−2680.
(39) Thallmair, S.; Javanainen, M.; Fábián, B.; Martínez-Seara, H.; Marrink, S. J. Nonconverged constraints cause artificial temperature gradients in lipid bilayer simulations. *J. Phys. Chem. B* **2021**, *125*, 9537−9546.
(40) López-Villellas, L.; Mikkelsen, C.; Galano, J.; Marco-Sola, S.; Alastruey, J.; Ibáñez Marín, P.; Moretó, M.; Sancho, J.; García-Risueño, P. Accurate and efficient constrained molecular dynamics of polymers using Newton's method and special purpose code. *Comput. Phys. Commun.* **2023**, *288*, No. 108742.
(41) Hammonds, K.; Heyes, D. Shadow Hamiltonian in classical NVE molecular dynamics simulations: A path to long time stability. *J. Chem. Phys.* **2020**, *152*, No. 024114.
(42) Toxvaerd, S.; Heilmann, O. J.; Ingebrigtsen, T.; Schrøder, T. B.; Dyre, J. C. Time-reversible molecular dynamics algorithms with bond constraints. *J. Chem. Phys.* **2009**, *131*, No. 064102.
(43) Manifold correction method for the Nosé−Hoover and Nosé−Poincare Molecular Dynamics simulations. Okumura, H.; Itoh, S. G.; Ito, A. M.; Nakamura, H.; Fukushima, T. *J. Phys. Soc. Jpn.* **2014**, *83*, No. 024003.
(44) Bussi, G.; Zykova-Timan, T.; Parrinello, M. Isothermal-isobaric Molecular Dynamics using stochastic velocity rescaling. *J. Chem. Phys.* **2009**, *130*, No. 074101.
(45) van Gunsteren, W. F.; Daura, X.; Fuchs, P. F.; Hansen, N.; Horta, B. A.; Hünenberger, P. H.; Mark, A. E.; Pechlaner, M.; Riniker, S.; Oostenbrink, C. On the effect of the various assumptions and approximations used in molecular simulations on the properties of bio-molecular systems: overview and perspective on issues. *ChemPhysChem* **2021**, *22*, 264−282.
(46) Hess, B.; van der Spoel, D.; Abraham, M.; Lindahl, E. *On The Importance of Accurate Algorithms for Reliable Molecular Dynamics Simulations*; Cambridge Open Engage: Cambridge, 2019.
(47) Chiu, S.-W.; Clark, M.; Subramaniam, S.; Jakobsson, E. Collective motion artifacts arising in long-duration Molecular Dynamics simulations. *J. Comput. Chem.* **2000**, *21*, 121−131.
(48) García-Risueño, P.; Echenique, P.; Alonso, J. L. Exact and efficient calculation of Lagrange multipliers in constrained biological polymers: Proteins and nucleic acids as example cases. *J. Comput. Chem.* **2011**, *32*, 3039−3046.
(49) García-Risueño, P.; Echenique, P. Linearly scaling direct method for accurately inverting sparse banded matrices. *Journal of Physics A: Mathematical and Theoretical* **2012**, *45*, No. 065204.

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(50) Golub, G. H.; Van Loan, C. F., Eds. *Matrix Computations*, 2nd ed.; The Johns Hopkins University Press: Baltimore and London, 1993.
(51) Lou, G. *Parallel methods for solving linear systems via overlapping decomposition. M.Sc. thesis*, University of Illinois at Urbana-Champaign, 1989.
(52) Eastman, P.; Pande, V. S. Constant Constraint Matrix Approximation: A robust, parallelizable constraint method for molecular simulations. *J. Chem. Theory Comput.* **2010**, *6*, 434−437.
(53) Kjelgaard Mikkelsen, C. C.; López-Villellas, L.; García-Risueño, P.; *How accurate does Newton have to be?* In Proceedings of the 14th International Conference on Parallel Processing and Applied Mathematics (PPAM), 2022; Part I, pp 3−15.
(54) Kjelgaard Mikkelsen, C. C.; López-Villellas, L.; García-Risueño, P. Newton's method revisited: How accurate do we have to be? *Concurr. Comput.: Pract. Exp.* **2024**, *36*, No. e7853.
(55) Huang, J.; MacKerell, A. D., Jr CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *J. Comput. Chem.* **2013**, *34*, 2135−2145.
(56) Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B. L.; Grubmüller, H.; MacKerell, A. D. J. CHARMM36m: an improved force field for folded and intrinsically disordered proteins. *Nat. Methods* **2017**, *14*, 71−73.
(57) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins: Struct., Funct., Bioinf.* **2006**, *65*, 712−725.
(58) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225−11236.
(59) Hammonds, K. D.; Ryckaert, J.-P. On the convergence of the SHAKE algorithm. *Comput. Phys. Commun.* **1991**, *62*, 336−351.
(60) Hopkins, C. W.; Le Grand, S.; Walker, R. C.; Roitberg, A. E. Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *J. Chem. Theory Comput.* **2015**, *11*, 1864−1874.
(61) Pal, S.; Jahanmahin, O.; Zhang, S.; Lee, T.-S. Impact of Timestep on Energy Drift and Accuracy of Alchemical Free Energy Calculations. *ChemRxiv* **2025**, This content is a preprint and has not been peer-reviewed.
(62) Balusek, C.; Hwang, H.; Lau, C. H.; Lundquist, K.; Hazel, A.; Pavlova, A.; Lynch, D. L.; Reggio, P. H.; Wang, Y.; Gumbart, J. C. Accelerating Membrane Simulations with Hydrogen Mass Repartitioning. *J. Chem. Theory Comput.* **2019**, *15*, 4673−4686.
(63) Sahil, M.; Sarkar, S.; Mondal, J. Long-time-step molecular dynamics can retard simulation of protein-ligand recognition process. *Biophys. J.* **2023**, *122*, 802−816.
(64) Wacha, A.; Varga, Z.; Beke-Somfai, T. Comparative Study of Molecular Mechanics Force Fields for β-Peptidic Foldamers: Folding and Self-Association. *J. Chem. Inf. Model.* **2023**, *63*, 3799−3813.
(65) Mikkelsen, C. C. K.; Alastruey-Benedé, J.; Ibáñez Marín, P.; García-Risueño, P.; *Accelerating sparse arithmetic in the context of Newton's method for small molecules with bond constraints*. In Proceedings of the 11th International Conference on Parallel Processing and Applied Mathematics (PPAM), 2016; Part I, pp 160−171.
(66) Miyamoto, S.; Kollman, P. A. Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models. *J. Comput. Chem.* **1992**, *13*, 952−962.

https://doi.org/10.1021/acs.jctc.5c01376
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX