NVU dynamics. III. Simulating molecules at constant potential energy

Ingebrigtsen, Trond; Dyre, J. C.

NVU dynamics. III. Simulating molecules at constant potential energy

Trond S. Ingebrigtsen[a] and Jeppe C. Dyre
DNRF Centre "Glass and Time," IMFUFA, Department of Sciences, Roskilde University, Postbox 260, DK-4000 Roskilde, Denmark

This is the final paper in a series that introduces geodesic molecular dynamics at constant potential energy. This dynamics is entitled *NVU* dynamics in analogy to standard energy-conserving Newtonian *NVE* dynamics. In the first two papers [T. S. Ingebrigtsen, S. Toxvaerd, O. J. Heilmann, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **135**, 104101 (2011); T. S. Ingebrigtsen, S. Toxvaerd, T. B. Schrøder, and J. C. Dyre, *ibid.* **135**, 104102 (2011)], a numerical algorithm for simulating geodesic motion of atomic systems was developed and tested against standard algorithms. The conclusion was that the *NVU* algorithm has the same desirable properties as the Verlet algorithm for Newtonian *NVE* dynamics, i.e., it is time-reversible and symplectic. Additionally, it was concluded that *NVU* dynamics becomes equivalent to *NVE* dynamics in the thermodynamic limit. In this paper, the *NVU* algorithm for atomic systems is extended to be able to simulate the geodesic motion of molecules at constant potential energy. We derive an algorithm for simulating rigid bonds and test this algorithm on three different systems: an asymmetric dumbbell model, Lewis-Wahnström o-terphenyl (OTP) and rigid SPC/E water. The rigid bonds introduce additional constraints beyond that of constant potential energy for atomic systems. The rigid-bond *NVU* algorithm conserves potential energy, bond lengths, and step length for indefinitely long runs. The quantities probed in simulations give results identical to those of Nosé-Hoover *NVT* dynamics. Since Nosé-Hoover *NVT* dynamics is known to give results equivalent to those of *NVE* dynamics, the latter results show that *NVU* dynamics becomes equivalent to *NVE* dynamics in the thermodynamic limit also for molecular systems. © *2012 American Institute of Physics*. [http://dx.doi.org/10.1063/1.4768957]

I. INTRODUCTION

In two recent papers[1,2] (henceforth, Papers I and II), molecular dynamics at constant potential energy was introduced, tested, and compared to well-known molecular dynamics algorithms. This new molecular dynamics is entitled *NVU* dynamics in analogy to standard energy-conserving Newtonian *NVE* dynamics. The conclusion was that *NVU* dynamics is a fully valid molecular dynamics, which for sufficiently large systems can be used interchangeably with *NVE* dynamics for calculating most quantities of interest. *NVU* dynamics is not faster than standard *NVE* or *NVT* dynamics, but introduces a new way of thinking about molecular dynamics. Molecular dynamics at constant potential energy was previously considered by Cotterill and co-workers,[3–6] by Scala *et al.*,[7] and most recently by Stratt and co-workers,[8–11] who, however, allowed also lower potential energy values. Our motivation for studying *NVU* dynamics derive from recent work on strongly correlating liquids and their isomorphs[12–19] (see the Introduction of Paper I).

NVU dynamics is defined by geodesic motion on the constant-potential-energy hypersurface Ω defined by

$$\Omega = \{\mathbf{R} \in R^{3N} \mid U(\mathbf{R}) = U_0\}. \quad (1)$$

Here, $\mathbf{R} \equiv \{\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(N)}\}$ in which $\mathbf{r}^{(k)}$ is the position vector of the k'te particle (we follow the notation of the Appendix of Paper II), and U is the potential-energy function of an N-particle classical system. A geodesic on Ω is a curve that satisfies the condition of stationary length for fixed endpoints $\mathbf{R}_A$ and $\mathbf{R}_B$, i.e.,

$$\delta \int_{\mathbf{R}_A}^{\mathbf{R}_B} dl \, \bigg|_{\Omega} = 0, \quad (2)$$

where dl is the line element of the metric. The shortest path between any two points is a geodesic. On a sphere, geodesics are great circles, the "straightest lines" of the surface. Traversing a geodesic at constant velocity thus corresponds to a generalization of Newton's first law to a curved space (the surface itself).

In Paper I, the *NVU* algorithm was developed via a discretization of Eq. (2), subsequently carrying out the variation. This technique, which is known as variational integration,[20–23] resulted in a "basic" *NVU* algorithm that is similar to the well-known Verlet algorithm $\mathbf{R}_{i+1} = 2\mathbf{R}_i - \mathbf{R}_{i-1} + (\Delta t)^2 \mathbf{F}_i/m$ for Newtonian (*NVE*) dynamics (m is the particle mass, which is here the same for all particles, and $\mathbf{F}_i \equiv -\nabla_{\mathbf{R}_i} U$ is the 3N-dimensional force vector); the index i refers to step i of the integration sequence. In the Verlet algorithm, Δt is a fixed time step length. In comparison, the basic *NVU* algorithm is given by (Paper I)

$$\mathbf{R}_{i+1} = 2\mathbf{R}_i - \mathbf{R}_{i-1} - \frac{2\mathbf{F}_i \cdot (\mathbf{R}_i - \mathbf{R}_{i-1})}{\mathbf{F}_i^2} \mathbf{F}_i. \quad (3)$$

If the number of particles N increases, the relative variation of the term $-2\mathbf{F}_i \cdot (\mathbf{R}_i - \mathbf{R}_{i-1})/\mathbf{F}_i^2$ decreases, and this is why equivalence with Newtonian *NVE* dynamics is established in

[a]trond@ruc.dk.

137, 244101-1

the thermodynamic limit. This equivalence should be understood in the sense that the relative deviations between, for instance, *NVE* and *NVU* time auto-correlation functions go to zero as $N \to \infty$.

Paper I additionally developed a "stabilized" version of the basic *NVU* algorithm to prevent the accumulation of numerical errors. This version of the algorithm is given by (defining the position changes $\mathbf{\Delta}_{i+1/2} \equiv \mathbf{R}_{i+1} - \mathbf{R}_i$)

$$\mathbf{\Delta}_{i+1/2} = l_0 \frac{\mathbf{A}_{i+1/2}}{\|\mathbf{A}_{i+1/2}\|}, \tag{4}$$

$$\mathbf{R}_{i+1} = \mathbf{R}_i + \mathbf{\Delta}_{i+1/2}, \tag{5}$$

where l_0 is the step length and

$$\mathbf{A}_{i+1/2} = \frac{\mathbf{\Delta}_{i-1/2} + (-2\mathbf{F}_i \cdot \mathbf{\Delta}_{i-1/2} + U_{i-1} - U_0)}{\mathbf{F}_i^2} \mathbf{F}_i. \tag{6}$$

All simulations in Papers I and II were performed with the stabilized algorithm. The basic algorithm was used, however, for theoretical considerations. Note that the basic *NVU* algorithm has the same excellent stability as the Verlet algorithm, and the accumulation of numerical errors is no more serious.

In this article, we extend the stabilized *NVU* algorithm to deal with simulations of molecular systems. Molecular systems are simulated by introducing rigid and/or flexible bonds between the atoms in the modelling. Flexible bonds introduce merely an additional contribution to *U*, for instance, harmonic spring potentials. The *NVU* algorithm conserves the total potential energy and can readily simulate flexible bonds. The focus in this paper is thus on implementing rigid bonds in the framework of *NVU* dynamics.

Section II considers *NVU* dynamics with rigid bonds. Introducing rigid bonds in the simulations lead to Lagrangian multipliers in addition to those introduced in order to keep the potential energy constant (Paper I). Section II is fairly technical and easiest to read after reading Paper I. Section III gives simulation and model details. Section IV tests the rigid-bond *NVU* algorithm, and Sec. V investigates the *NVU* sampling properties by comparing the *NVU* results to Nosé-Hoover *NVT* results[24, 25] on three different systems: the asymmetric dumbbell model,[26] Lewis-Wahnström OTP,[27] and rigid SPC/E water.[28] Nosé-Hoover *NVT* dynamics is known to give results equivalent to *NVE* dynamics in the thermodynamic limit,[29] and we refer to these dynamics interchangeably in the forthcoming sections. Finally, Sec. VI concludes.

II. RIGID-BOND *NVU* ALGORITHM

The rigid bonds[30, 31] introduce constraints among the particle coordinates of the system. Each constraint $\alpha = 1, \ldots, G$ is of the form

$$\sigma_\alpha(\mathbf{R}) \equiv (\mathbf{r}^{(k_\alpha)} - \mathbf{r}^{(l_\alpha)})^2 \equiv (\mathbf{r}^\alpha)^2 = C_\alpha^2; \tag{7}$$

it expresses that the distance between particles k_α and l_α is a constant, C_α. In Papers I and II, the integral of Eq. (2) was merely restricted to the constant-potential-energy hypersurface Ω. Each rigid bond constraint introduces a function σ_α to be kept constant, and thus the integral of Eq. (2) is now further restricted to the sub-manifold ω of Ω where the bond

constraints are satisfied,

$$\omega = \{\mathbf{R} \in \Omega \mid \sigma_\alpha(\mathbf{R}) = C_\alpha^2, \ \alpha = 1, \ldots, G\}. \tag{8}$$

If the bond constraints are independent, as assumed throughout the paper, ω is a (3N - G - 1)-dimensional compact Riemannian manifold. The variational principle defining *NVU* dynamics with rigid bonds is given by

$$\delta \int_{\mathbf{R}_A}^{\mathbf{R}_B} dl = 0 \bigg|_\omega. \tag{9}$$

Most of Papers I and II dealt with the case of identical particle masses, but we wish here to develop a completely general molecular *NVU* algorithm. The line element dl is defined by

$$dl^2 \equiv \sum_k \tilde{m}_k (d\mathbf{r}^{(k)})^2, \tag{10}$$

where $\tilde{m}_k = m_k / \langle m \rangle$ is the "reduced" mass of particle *k*. Equation (10) is not the standard Euclidean line element, but a mass-weighted line element that goes back to Hertz.[32, 33] We shall refer to this metric as the "Hertzian" metric. This metric ensures equivalence between *NVU* and *NVE* dynamics for systems of atoms and molecules of varying mass. In the Appendix, we derive the variable-mass atomic *NVU* algorithm applying the Hertzian metric (correcting also a typo of the Appendix of Paper II).

Applying the variational integration technique to Eq. (9) gives

$$\delta \left(\sum_i \sqrt{\sum_k \tilde{m}_k \left(\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)} \right)^2} - \sum_i \lambda_i U(\mathbf{R}_i) \right.$$
$$\left. + \sum_{i,\alpha} \Lambda_{\alpha i} \sigma_\alpha(\mathbf{R}_i) \right) = 0. \tag{11}$$

In Eq. (11), the path is divided into a number of discrete points and one Lagrangian multiplier $\Lambda_{\alpha i}$ is introduced for each constraint α at every point *i*. Following standard notation for constraint molecular dynamics,[30, 31] the Lagrangian multipliers of the bond constraints are chosen with a positive sign. As in Papers I and II, we now make the Ansatz of constant step length l_0, i.e.,

$$\sum_k \tilde{m}_k \left(\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)} \right)^2 \equiv l_0^2. \tag{12}$$

Carrying out the variation of Eq. (11) using Eq. (12) leads to (compare the derivation in Paper I)

$$\mathbf{r}_{i+1}^{(k)} = 2\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)} + \frac{l_0}{\tilde{m}_k} \lambda_i \mathbf{f}_i^{(k)} + \frac{l_0}{\tilde{m}_k} \nabla_{\mathbf{r}_i^{(k)}} \sum_\alpha \Lambda_{\alpha i} \sigma_\alpha, \tag{13}$$

where $\mathbf{f}_i^{(k)} = -\nabla_{\mathbf{r}_i^{(k)}} U$ is the force on particle *k* at step *i*. This equation constitutes the *NVU* algorithm with rigid bonds. It has a close resemblance to the Lagrangian equations of motion with holonomic constraints,[31] i.e., rigid-bond *NVE* dynamics.[30] Equation (13) contains $G + 1$ Lagrangian multipliers for each integration step, which must be determined to complete the algorithm.

TABLE I. Definitions and nomenclature of the text.

Symbol	Definition
$\sigma_\alpha(\mathbf{R})$	The α'th bond constraint between particles k_α and l_α with $\alpha = 1, \ldots, G$. ($\sigma_\alpha = (\mathbf{r}^\alpha)^2 = C_\alpha^2$).
$\tilde{m}_k$	The mass of particle k divided by the average mass of the system. ($\tilde{m}_k = m_k/\langle m \rangle$).
	3-dimensional vectors
$\mathbf{r}_i^{(k)}$	Position of particle k at step i.
$\boldsymbol{\delta}_{i+1/2}^{(k)}$	Displacement of the position of particle k between step i and $i+1$. ($\boldsymbol{\delta}_{i+1/2}^{(k)} = \mathbf{r}_{i+1}^{(k)} - \mathbf{r}_i^{(k)}$).
$\mathbf{f}_i^{(k)}$	Force on particle k at step i. ($\mathbf{f}_i^{(k)} = -\nabla_{\mathbf{r}_i^{(k)}} U$).
$\mathbf{g}_i^{(k)}$	Constraint force on particle k at step i. ($\mathbf{g}_i^{(k)} = \nabla_{\mathbf{r}_i^{(k)}} \sum_\alpha \Lambda_{\alpha i} \sigma_\alpha$).
$\mathbf{r}_i^\alpha$	Displacement of the positions of particles k_α and l_α at step i. ($\mathbf{r}_i^\alpha = \mathbf{r}_i^{(k_\alpha)} - \mathbf{r}_i^{(l_\alpha)}$).
$\boldsymbol{\delta}_{i-1/2}^\alpha$	Displacement of the velocities of particles k_α and l_α at step $i - 1/2$. ($\boldsymbol{\delta}_{i-1/2}^\alpha = \boldsymbol{\delta}_{i-1/2}^{(k_\alpha)} - \boldsymbol{\delta}_{i-1/2}^{(l_\alpha)}$).
$\mathbf{s}_i^\alpha$	Sum of displacements of positions and velocities of particles k_α and l_α at, respectively, step i and $i - 1/2$. ($\mathbf{s}_i^\alpha = \mathbf{r}_i^\alpha + \boldsymbol{\delta}_{i-1/2}^\alpha$).
$\tilde{\mathbf{f}}_i^\alpha$	Displacement of the forces on particles k_α and l_α at step i divided by their reduced particle mass. ($\tilde{\mathbf{f}}_i^\alpha = \mathbf{f}_i^{(k_\alpha)}/\tilde{m}_{k_\alpha} - \mathbf{f}_i^{(l_\alpha)}/\tilde{m}_{l_\alpha}$).
$\tilde{\mathbf{g}}_i^\alpha$	Displacement of the constraint forces on particles k_α and l_α at step i divided by their reduced particle mass. ($\tilde{\mathbf{g}}_i^\alpha = \mathbf{g}_i^{(k_\alpha)}/\tilde{m}_{k_\alpha} - \mathbf{g}_i^{(l_\alpha)}/\tilde{m}_{l_\alpha}$).
	3N-dimensional vectors
$\mathbf{R}_i$	Position of all particles at step i. ($\mathbf{R}_i = \{\mathbf{r}_i^{(1)}, \ldots, \mathbf{r}_i^{(N)}\}$).
$\boldsymbol{\Delta}_{i+1/2}$	Displacement of the positions between step i and $i+1$. ($\boldsymbol{\Delta}_{i+1/2} = \mathbf{R}_{i+1} - \mathbf{R}_i$).
$\mathbf{F}_i$	Force on all particles at step i. ($\mathbf{F}_i = -\nabla_{\mathbf{R}_i} U$).
$\tilde{\mathbf{F}}_i$	Force on all particles at step i divided by the reduced particle mass. ($\tilde{\mathbf{F}}_i = \{\mathbf{f}_i^{(1)}/\tilde{m}_1, \ldots, \mathbf{f}_i^{(N)}/\tilde{m}_N\}$).
$\tilde{\mathbf{G}}_i$	Constraint force on all particles at step i divided by the reduced particle mass. ($\tilde{\mathbf{G}}_i = \{\mathbf{g}_i^{(1)}/\tilde{m}_1, \ldots, \mathbf{g}_i^{(N)}/\tilde{m}_N\}$).

A. Determining the *NVU* Lagrangian multipliers

This section shows how to calculate the Lagrangian multipliers. Since the algorithm is to be implemented on a computer (with finite-precision), we shall proceed directly to a "stabilized" algorithm conserving for indefinitely long runs potential energy, bond lengths, and step length (in 3N-dimensions). The resulting algorithm reduces to the stabilized atomic *NVU* algorithm of Eqs. (4)–(6) in the case of no bonds constraints.

Some notation used in the following derivation is now introduced (the nomenclature of text is summarized in Table I). Defining $\boldsymbol{\delta}_{i+1/2}^{(k)} \equiv \mathbf{r}_{i+1}^{(k)} - \mathbf{r}_i^{(k)}$ and $\mathbf{g}_i^{(k)} \equiv \nabla_{\mathbf{r}_i^{(k)}} \sum_\alpha \Lambda_{\alpha i} \sigma_\alpha$ the "Leap-frog"[34] version of the rigid-bond *NVU* algorithm Eq. (13) reads

$$\boldsymbol{\delta}_{i+1/2}^{(k)} = \boldsymbol{\delta}_{i-1/2}^{(k)} + \frac{l_0}{\tilde{m}_k}\lambda_i \mathbf{f}_i^{(k)} + \frac{l_0}{\tilde{m}_k}\mathbf{g}_i^{(k)}, \qquad (14)$$

$$\mathbf{r}_{i+1}^{(k)} = \mathbf{r}_i^{(k)} + \boldsymbol{\delta}_{i+1/2}^{(k)}. \qquad (15)$$

In analogy to rigid-bond *NVE* dynamics we call $\mathbf{g}_i^{(k)}$ the "constraint force" on particle k at step i. Introducing the notation $\tilde{\mathbf{F}}_i \equiv \{\mathbf{f}_i^{(1)}/\tilde{m}_1, \ldots, \mathbf{f}_i^{(N)}/\tilde{m}_N\}$ and $\tilde{\mathbf{G}}_i \equiv \{\mathbf{g}_i^{(1)}/\tilde{m}_1, \ldots, \mathbf{g}_i^{(N)}/\tilde{m}_N\}$, the *NVU* algorithm in the full 3N-dimensional coordinate space reads

$$\boldsymbol{\Delta}_{i+1/2} = \boldsymbol{\Delta}_{i-1/2} + l_0\lambda_i \tilde{\mathbf{F}}_i + l_0\tilde{\mathbf{G}}_i, \qquad (16)$$

$$\mathbf{R}_{i+1} = \mathbf{R}_i + \boldsymbol{\Delta}_{i+1/2}, \qquad (17)$$

The Lagrangian multipliers are calculated by combining a result derived in Paper I with the method applied in the *SHAKE* algorithm[30] for rigid bonds in *NVE* dynamics.[30,35,36] The *SHAKE* algorithm calculates the Lagrangian multipliers from the equations $(\mathbf{r}_{i+1}^\alpha)^2 = C_\alpha^2$. In doing so, the target value of the constraints C_α appears explicitly in the algorithm, making the bond lengths insensitive to numerical error. The ex-

pression for $\mathbf{r}_{i+1}^\alpha$ is supplied by the integration algorithm containing herein the Lagrangian multipliers. In our case, this gives G equations with $G + 1$ unknowns. The missing equation is supplied by an expression derived in Paper I, namely that $U_{i+1} = U_{i-1} - \mathbf{F}_i \cdot (\mathbf{R}_{i+1} - \mathbf{R}_{i-1})$ to third order in the step length. In the discrete sequence of points, U_{i+1} is set equal to U_0 (the constant defining Ω), making the constraint of constant potential energy also insensitive to numerical errors. We thus have the following $G + 1$ equations for calculating the Lagrangian multipliers

$$U_{i-1} - \mathbf{F}_i \cdot (\mathbf{R}_{i+1} - \mathbf{R}_{i-1}) - U_0 = 0, \qquad (18)$$

$$\left(\mathbf{r}_{i+1}^\alpha\right)^2 - C_\alpha^2 = 0, \quad (\alpha = 1, \ldots, G). \qquad (19)$$

By Eqs. (16) and (17); $\mathbf{R}_{i+1} - \mathbf{R}_{i-1} = \boldsymbol{\Delta}_{i+1/2} + \boldsymbol{\Delta}_{i-1/2} = 2\boldsymbol{\Delta}_{i-1/2} + l_0\lambda_i \tilde{\mathbf{F}}_i + l_0\tilde{\mathbf{G}}_i$. Defining $\boldsymbol{\delta}_{i-1/2}^\alpha \equiv \boldsymbol{\delta}_{i-1/2}^{(k_\alpha)} - \boldsymbol{\delta}_{i-1/2}^{(l_\alpha)}$, $\tilde{\mathbf{f}}_i^\alpha \equiv \mathbf{f}_i^{(k_\alpha)}/\tilde{m}_{k_\alpha} - \mathbf{f}_i^{(l_\alpha)}/\tilde{m}_{l_\alpha}$, and $\tilde{\mathbf{g}}_i^\alpha \equiv \mathbf{g}_i^{(k_\alpha)}/\tilde{m}_{k_\alpha} - \mathbf{g}_i^{(l_\alpha)}/\tilde{m}_{l_\alpha}$, since by Eqs. (14) and (15); $\mathbf{r}_{i+1}^\alpha = \mathbf{r}_{i+1}^{(k_\alpha)} - \mathbf{r}_{i+1}^{(l_\alpha)} = \mathbf{r}_i^{(k_\alpha)} - \mathbf{r}_i^{(l_\alpha)} + \boldsymbol{\delta}_{i+1/2}^{(k_\alpha)} - \boldsymbol{\delta}_{i+1/2}^{(l_\alpha)} = \mathbf{r}_i^\alpha + \boldsymbol{\delta}_{i-1/2}^\alpha + l_0\lambda_i \tilde{\mathbf{f}}_i^\alpha + l_0\tilde{\mathbf{g}}_i^\alpha$, it follows that

$$U_{i-1} - \mathbf{F}_i \cdot [2\boldsymbol{\Delta}_{i-1/2} + l_0\lambda_i \tilde{\mathbf{F}}_i + l_0\tilde{\mathbf{G}}_i] - U_0 = 0, \qquad (20)$$

$$\left[\mathbf{r}_i^\alpha + \boldsymbol{\delta}_{i-1/2}^\alpha + l_0\lambda_i \tilde{\mathbf{f}}_i^\alpha + l_0\tilde{\mathbf{g}}_i^\alpha\right]^2 - C_\alpha^2 = 0, \quad (\alpha = 1, \ldots, G). \qquad (21)$$

The above coupled quadratic equations for the Lagrangian multipliers are now solved following the produce of the *MILC-SHAKE* algorithm,[37] which starts by neglecting the second order terms in the Lagrangian multipliers and solving the resulting linear equations. Afterwards, the second order terms are taken into account in an iterative manner—the details of which are described below.

For each integration step i, the linearized equations are given as

$$\mathbf{A}_i \boldsymbol{\lambda}_i = \mathbf{b}_i, \qquad (22)$$

where $\mathbf{A}_i$ is a $(G + 1) \times (G + 1)$ matrix, $\boldsymbol{\lambda}_i \equiv \{\lambda_i, \Lambda_{1i}, \ldots, \Lambda_{Gi}\}$, and $\mathbf{b}_i$ a $G + 1$ column vector. We start by calculating explicitly the first few elements of the matrix $\mathbf{A}_i$. A_{11} consists merely of the factor in front of λ_i in Eq. (20), i.e., $A_{11} = -l_0 \tilde{\mathbf{F}}_i \cdot \mathbf{F}_i$. The second element A_{12} appears after expansion of the dot product $\mathbf{F}_i \cdot \tilde{\mathbf{G}}_i$. Noting that $\nabla_{\mathbf{r}_i^{(k_\alpha)}} \sigma_\alpha = 2\mathbf{r}_i^\alpha$, we have $\mathbf{F}_i \cdot \tilde{\mathbf{G}}_i = \mathbf{f}_i^{(1)} \cdot \mathbf{g}_i^{(1)}/\tilde{m}_1 + \ldots + \mathbf{f}_i^{(N)} \cdot \mathbf{g}_i^{(N)}/\tilde{m}_N = 2\Lambda_{1i}(\tilde{\mathbf{f}}_i^1 \cdot \mathbf{r}_i^1) + \ldots + 2\Lambda_{Gi}(\tilde{\mathbf{f}}_i^G \cdot \mathbf{r}_i^G)$. The last equation follows as the Lagrangian multipliers appear in pairs, differing only by the sign from $\nabla_{\mathbf{r}_i^{(k_\alpha)}} \sigma_\alpha$ and the term $\mathbf{f}_i^{(k_\alpha)}/\tilde{m}_{k_\alpha}$. We thus

find $A_{12} = -2l_0 \tilde{\mathbf{f}}_i^1 \cdot \mathbf{r}_i^1$, $A_{13} = -2l_0 \tilde{\mathbf{f}}_i^2 \cdot \mathbf{r}_i^2$, etc. In the second row of $\mathbf{A}_i$, the short-hand notation $\mathbf{s}_i^\alpha \equiv \mathbf{r}_i^\alpha + \delta_{i-1/2}^\alpha$ is introduced, making $A_{21} = 2l_0(\mathbf{s}_i^1 \cdot \tilde{\mathbf{f}}_i^1)$, i.e., the factor in front of λ_i after squaring of the parentheses. The next element A_{22} appears after expanding $\mathbf{s}_i^1 \cdot \tilde{\mathbf{g}}_i^1 = \mathbf{s}_i^1 \cdot \sum_\beta \Lambda_{\beta i}(\frac{1}{\tilde{m}_{k_1}} \nabla_{\mathbf{r}_i^{(k_1)}} \sigma_\beta - \frac{1}{\tilde{m}_{l_1}} \nabla_{\mathbf{r}_i^{(l_1)}} \sigma_\beta)$. In this sum, we identify the factor in front of Λ_{1i}, giving $A_{22} = 2l_0 \mathbf{s}_i^1 \cdot (\frac{1}{\tilde{m}_{k_1}} \nabla_{\mathbf{r}_i^{(k_1)}} \sigma_1 - \frac{1}{\tilde{m}_{l_1}} \nabla_{\mathbf{r}_i^{(l_1)}} \sigma_1)$, and similarly for the remaining elements of the second row.

Altogether, the elements of $\mathbf{A}_i$ are thus given by

$$\mathbf{A}_i = 2l_0 \begin{pmatrix} -\tilde{\mathbf{F}}_i \cdot \mathbf{F}_i/2 & -\tilde{\mathbf{f}}_i^1 \cdot \mathbf{r}_i^1 & \cdots & -\tilde{\mathbf{f}}_i^G \cdot \mathbf{r}_i^G \\ \mathbf{s}_i^1 \cdot \tilde{\mathbf{f}}_i^1 & \mathbf{s}_i^1 \cdot \left(\frac{1}{\tilde{m}_{k_1}} \nabla_{\mathbf{r}_i^{(k_1)}} \sigma_1 - \frac{1}{\tilde{m}_{l_1}} \nabla_{\mathbf{r}_i^{(l_1)}} \sigma_1\right) & \cdots & \mathbf{s}_i^1 \cdot \left(\frac{1}{\tilde{m}_{k_1}} \nabla_{\mathbf{r}_i^{(k_1)}} \sigma_G - \frac{1}{\tilde{m}_{l_1}} \nabla_{\mathbf{r}_i^{(l_1)}} \sigma_G\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_i^G \cdot \tilde{\mathbf{f}}_i^G & \mathbf{s}_i^G \cdot \left(\frac{1}{\tilde{m}_{k_G}} \nabla_{\mathbf{r}_i^{(k_G)}} \sigma_1 - \frac{1}{\tilde{m}_{l_G}} \nabla_{\mathbf{r}_i^{(l_G)}} \sigma_1\right) & \cdots & \mathbf{s}_i^G \cdot \left(\frac{1}{\tilde{m}_{k_G}} \nabla_{\mathbf{r}_i^{(k_G)}} \sigma_G - \frac{1}{\tilde{m}_{l_G}} \nabla_{\mathbf{r}_i^{(l_G)}} \sigma_G\right) \end{pmatrix}. \quad (23)$$

The column vector $\mathbf{b}_i$ consists of all zeroth-order terms in Eqs. (20) and (21)

$$\mathbf{b}_i = \begin{pmatrix} U_0 - U_{i-1} + 2\mathbf{F}_i \cdot \boldsymbol{\Delta}_{i-1/2} \\ C_1^2 - (\mathbf{s}_i^1)^2 \\ \vdots \\ C_G^2 - (\mathbf{s}_i^G)^2 \end{pmatrix}. \quad (24)$$

Turning now to the iteration procedure, the second-order terms in the Lagrangian multipliers (Eq. (21)) are taken into account by iterating the right-hand side of Eq. (22) via the scheme ($\alpha = 1, \ldots, G$)

$$b_\alpha^{j+1} = b_\alpha^j + \left[C_\alpha^2 - \left((\mathbf{r}_{i+1}^\alpha)^2\right)^j\right]. \quad (25)$$

The superscript j refers here to iteration j, and $((\mathbf{r}_{i+1}^\alpha)^2)^j$ are the positions associated with iteration j. The element b_0 is not updated as it derives from the constraint of constant potential energy. For each iteration j, the term $C_\alpha^2 - ((\mathbf{r}_{i+1}^\alpha)^2)^j$ is expected to become smaller as the bonds are satisfied better and better, and indeed, convergence was achieved within a few iterations.[37]

For each integration step i, the algorithm for determining the *NVU* Lagrangian multipliers thus proceeds as follows:

1. The Lagrangian multipliers of iteration j, $(\boldsymbol{\lambda}_i)^j$, are calculated from Eq. (22).
2. $((\mathbf{r}_{i+1}^\alpha)^2)^j$ is calculated via Eqs. (14) and (15) using $(\boldsymbol{\lambda}_i)^j$.
3. $\mathbf{b}_i$ is updated via Eq. (25) from $((\mathbf{r}_{i+1}^\alpha)^2)^j$.
4. The above steps are repeated until convergence is established (we used a preset number of iterations, typically 3–5).

How is constant step length l_0 ensured numerically after determining the Lagrangian multipliers? Generalizing the ap-

proach of Paper I, we introduce a normalizing factor such that

$$\delta_{i+1/2}^{(k)} = l_0 \frac{\boldsymbol{\chi}_{i+1/2}^{(k)}}{\sqrt{\sum_k \tilde{m}_k \left(\boldsymbol{\chi}_{i+1/2}^{(k)}\right)^2}}, \quad (26)$$

$$\mathbf{r}_{i+1}^{(k)} = \mathbf{r}_i^{(k)} + \delta_{i+1/2}^{(k)}, \quad (27)$$

where

$$\boldsymbol{\chi}_{i+1/2}^{(k)} \equiv \delta_{i-1/2}^{(k)} + \frac{l_0}{\tilde{m}_k} \lambda_i \mathbf{f}_i^{(k)} + \frac{l_0}{\tilde{m}_k} \mathbf{g}_i^{(k)}. \quad (28)$$

The normalizing factor is close to unity[1] and ensures trivially $\sum_k \tilde{m}_k (\delta_{i+1/2}^{(k)})^2 = l_0^2$, i.e., that the step length is conserved. The algorithm is now absolutely stable, conserving potential energy, bond lengths, and step length for indefinitely long runs. The stability of the *NVU* algorithm is tested numerically in Sec. IV.

B. Alternative determination of the *NVU* Lagrangian multipliers

Section II A followed the traditional way of calculating the Lagrangian multipliers. The *NVU* Lagrangian multipliers may also be calculated by Taylor expanding the constraints σ_α in analogy to the method sketched above for the potential energy. In this way, the constraints of constant potential energy and constant bond lengths are treated on equal footing. The set of equations to be solved is the following (recall that $\mathbf{R}_{i+1} - \mathbf{R}_{i-1} = 2\boldsymbol{\Delta}_{i-1/2} + l_0 \lambda_i \tilde{\mathbf{F}}_i + l_0 \tilde{\mathbf{G}}_i$),

$$U_{i-1} - \mathbf{F}_i \cdot (\mathbf{R}_{i+1} - \mathbf{R}_{i-1}) - U_0 = 0, \quad (29)$$

$$\sigma_{\alpha(i-1)} + \nabla_{\mathbf{R}_i} \sigma_{\alpha i} \cdot (\mathbf{R}_{i+1} - \mathbf{R}_{i-1}) - C_\alpha^2 = 0, \quad (\alpha = 1, \ldots, G). \quad (30)$$

The equations for the Lagrangian multipliers are now linear and thus no iterations are needed. The bond constraints σ_α are obeyed to the same order $O(l_0^3)$ as the constraint of constant potential energy. The sampling properties of this alternative determination method is tested briefly in Sec. V. It appears to be a promising new way of determining the Lagrangian multipliers in connection with rigid bonds, which might also be useful for standard bond-constraint *NVE* or *NVT* simulations.

III. SIMULATION DETAILS AND MODEL SYSTEMS

We investigated three systems: the asymmetric dumbbell model, the Lewis-Wahnström OTP model, and rigid SPC/E water. For all simulated pair potentials, the shifted-force truncation scheme was applied at a cut-off radius r_c. If the pair potential is $v(r)$ and the pair force is $f(r) = -v'(r)$, the shifted force is given by[34, 38]

$$f_{\text{SF}}(r) = \begin{cases} f(r) - f(r_c) & \text{if} \quad r < r_c\,, \\ 0 & \text{if} \quad r > r_c\,. \end{cases} \tag{31}$$

This corresponds to using the following pair potential below r_c: $v_{\text{SF}}(r) = v(r) - v'(r_c)(r - r_c) - v(r_c)$. All simulations were performed with the *NVT* and *NVU* algorithms. Recall that *NVE* and *NVT* dynamics give equivalent results;[29] for this reason, no simulations are presented for *NVE* dynamics. The Roskilde University Molecular Dynamics (RUMD) code[39] was used for molecular dynamics simulations (an optimized open-source GPU code). The *NVT* ensemble is generated via the Nosé-Hoover algorithm,[24, 25, 40] and the bonds held fixed using the time-reversible constraint algorithm of Refs. 35 and 36. The *NVU* algorithm is described in Sec. II. The starting files for *NVU* dynamics were taken from an equilibrated *NVT* simulation. The positions and velocities of the *NVT* configuration do not correspond perfectly to motion on ω, since the potential energy and step length are not those of $U_0 = \langle U \rangle$ and l_0, respectively. As all the constraints are to be satisfied simultaneously, this results in numerical problems when starting the simulation from the particular *NVT* configuration. A more gentle procedure is thus applied, where the atomic *NVU* algorithm is used for a couple of integration steps to ensure the correct values of U_0 and l_0. Afterwards, the rigid-bond *NVU* algorithm is used.

A. *NVU* iteration procedure

The quadratic equations (Eq. (25)) were iterated with a fixed number of iterations (between 3 and 5). The linear systems were solved utilizing Cusp,[41] a library for solving systems of linear equations on the *GPU*. More specifically, the stabilized biconjugate gradient algorithm with a Jacobi preconditioner[42] was used with the initial value $\lambda_i = \lambda_{i-1}$ ($\lambda_0 = 0$ for the start of the simulation). The relative tolerance τ of the solver for the asymmetric dumbbell and Lewis-Wahnström OTP models was chosen as $\tau = 10^{-7}$ and for rigid SPC/E water as $\tau = 3 \times 10^{-7}$. A larger tolerance was chosen for rigid SPC/E water due to convergence issues in connection with the shifted-force Coulomb interactions (see below).

The maximum number of allowed iterations was 50. A restart scheme was applied when the solver did not converge within the chosen tolerance. In this case, the solver (and quadratic iteration) was restarted from the partially estimated "solution" adding 2×10^{-7} to the tolerance. It should be noted that the stabilized biconjugate gradient algorithm may get trapped, resulting in a break-down of the Cusp linear solver. If this happens, it is detected by our program, and the solver and quadratic iteration are restarted with a smaller number (10) of maximum allowed iterations for the solver.

B. The asymmetric dumbbell

The asymmetric dumbbell model[26] consists of a large (*A*) and a small (*B*) Lennard-Jones (LJ) particle, rigidly bonded with bond distance of $r_{AB} = 0.29/0.4963$ (here and henceforth units are given in LJ units referring to the *A* particle such that $\sigma_{AA} = 1$, $\epsilon_{AA} = 1$, and $m_A = 1$). The asymmetric dumbbell model has $\sigma_{BB} = 0.3910/0.4963$, $\epsilon_{BB} = 0.66944/5.726$, and $m_B = 15.035/77.106$. The *AB* interaction between different molecules is determined by the Lorentz-Berthelot mixing rule.[34] $n = 500$ molecules (here and henceforth n denotes the number of molecules and N the number of atoms) were used in the simulations with a pair-potential cut-off of $r_c = 2.5$. The step length l_0 was fixed in the range 0.125–0.138 depending on the state point.

Simulations were also performed where the rigid bonds were replaced by stiff harmonic springs. The spring constant was $k = 3000$, while all other model parameters remained unchanged.

C. Lewis-Wahnström OTP

The Lewis-Wahnström OTP model[27] consists of three identical LJ particles rigidly bonded in an isosceles triangle with sides of $r_{AA} = 1$ and top angle of $75°$. All parameters (including the masses) are unity for the OTP model. $n = 320$ molecules were simulated and a pair-potential cut-off of $r_c = 2.5$ was used. The step length was 0.100.

D. SPC/E water

The SPC/E water model[28] is an isosceles triangle with sides $r_{OH} = 1/3.166$ and top angle $109.47°$. The OO intermolecular interactions are given by the LJ pair potential ($\epsilon_{OO} = 1$, $\sigma_{OO} = 1$, and $m_O = 15.9994/1.00794$). The three

TABLE II. Potential energy, deviation of bond lengths and step length as functions of integration step number in the *NVU* algorithm for Lewis-Wahnström OTP ($\rho = 0.329$, T = 0.700). Single-precision floating-point arithmetic was used for the simulations.

Integration steps	U/N	$(1/G \sum_\alpha (r^\alpha - C_\alpha)^2)^{1/2}$	$\sum_k \bar{m}_k (\delta_{i+1/2}^{(k)})^2$
10^1	−4.42550	2.81207×10^{-7}	0.0999999
10^2	−4.42552	3.03535×10^{-7}	0.1000000
10^3	−4.42552	2.81128×10^{-7}	0.1000000
10^4	−4.42552	2.95078×10^{-7}	0.1000000
10^5	−4.42550	3.08793×10^{-7}	0.1000000
10^6	−4.42551	2.90477×10^{-7}	0.1000000

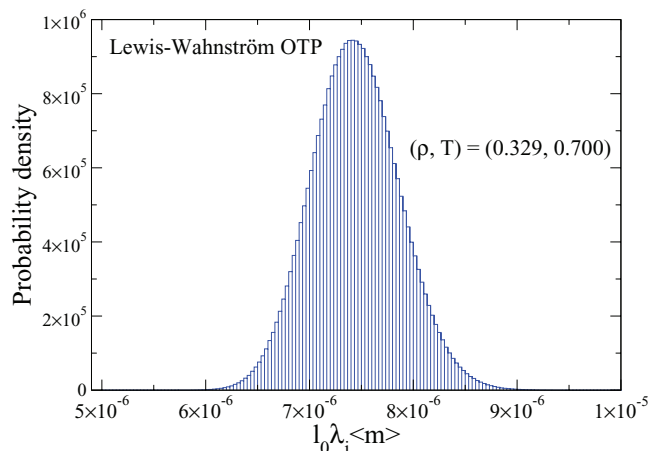FIG. 1. The probability density of the "time step" length $(\Delta t_{i,NVU})^2$ $\equiv l_0\lambda_i\langle m\rangle$ of the rigid-bond *NVU* algorithm for Lewis-Wahnström OTP at $\rho = 0.329$ and T = 0.700. $n = 320$ molecules were simulated.

particles are charged with $q_O = -22.0$ and $q_H = |q_O|/2$. $n = 2000$ molecules were simulated and a pair-potential cut-off of $r_c = 6.28$ for both LJ and Coulomb interactions was applied.[43,44] The step length was fixed in the range 0.06–0.07 depending on the state point. For this system, the numerical stability is sensitive to the cut-off used in the Coulomb interactions, but a larger shifted-force cut-off improves this behavior.[44]

IV. TESTING THE STABILITY OF THE RIGID-BOND *NVU* ALGORITHM

This section tests the conservation properties of the rigid-bond *NVU* algorithm. Table II shows the potential energy, the deviation of bond lengths, and step length as functions of integration step number for Lewis-Wahnström OTP at $\rho = 0.329$ and T = 0.700. It is clear that these quantities are conserved by the algorithm and that no drift occurs. The step length is conserved to the highest accuracy since it is not prone to numerical error in determining the Lagrangian multipliers.
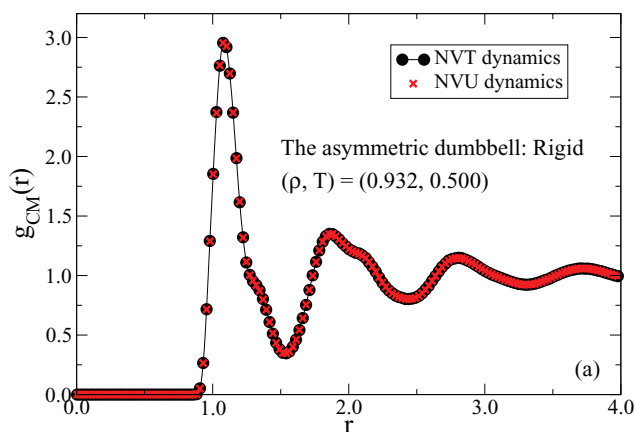
Figure 1 shows the distribution of the term $l_0\lambda_i\langle m\rangle$ in Eq. (13) ($\tilde{m}_k = m_k/\langle m\rangle$). In *NVU* dynamics there is, as such, no notation of time; a geodesic on the manifold can be traversed with any velocity. Comparing the *NVU* algorithm of Eq. (13) to the rigid-bond Verlet algorithm[30] $\mathbf{r}_{i+1}^{(k)}$ $= 2\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)} + ((\Delta t)^2/m_k)[\mathbf{f}_i^{(k)} + \mathbf{g}_i^{(k)}]$, we can define the term $l_0\lambda_i\langle m\rangle$ as a varying "time step" length of the *NVU* algorithm (see also Paper II), i.e.,

$$(\Delta t_{i,NVU})^2 \equiv l_0\lambda_i\langle m\rangle. \tag{32}$$

The integration steps of the *NVU* algorithm are thus henceforth referred to as "time steps." The average of Eq. (32) is used in Sec. V when comparing to *NVT* dynamics. As was the case for the atomic *NVU* algorithm (Paper I), $l_0\lambda_i\langle m\rangle$ is Gaussian distributed for large systems and its relative variation decreases as the number of particles increases. It thus becomes a better and better approximation to treat this term as constant, implying equivalent sampling properties of *NVU* and *NVE* dynamics also when rigid bonds are included in the simulations.

V. SAMPLING PROPERTIES OF THE RIGID-BOND *NVU* ALGORITHM

The *NVU* algorithm is now compared to *NVT* dynamics for the three different models. First, we consider the asymmetric dumbbell model,[26] both rigid and flexible. Afterwards, the Lewis-Wahnström OTP model,[27] and finally the rigid SPC/E water model.[28]

A. The asymmetric dumbbell model

In Figs. 2(a) and 2(b) are shown, respectively, the molecular center-of-mass (CM) radial distribution functions and the CM incoherent intermediate scattering functions for the rigid asymmetric dumbbell model[26] for different temperatures at $\rho = 0.932$. The black circles and curves give *NVT* simulation results while the red crosses give the *NVU* simulation results. The two radial distribution functions in Fig. 2(a) agree very well, and this is also the case for the dynamics in Fig. 2(b).
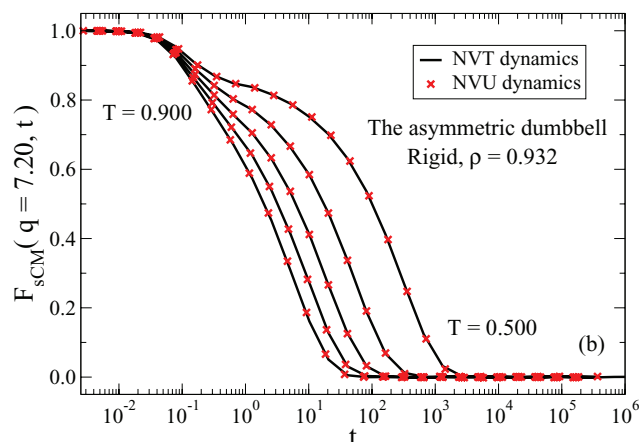

FIG. 2. Comparison of structure and dynamics in *NVU* and *NVT* simulations of the rigid asymmetric dumbbell model. The black circles and curves give *NVT*, the red crosses *NVU* simulation results. (a) The molecular CM radial distribution functions at $\rho = 0.932$ and T = 0.500. (b) The molecular CM incoherent intermediate scattering functions at $\rho = 0.932$ and T = 0.500, 0.600, 0.700, 0.800, 0.900.
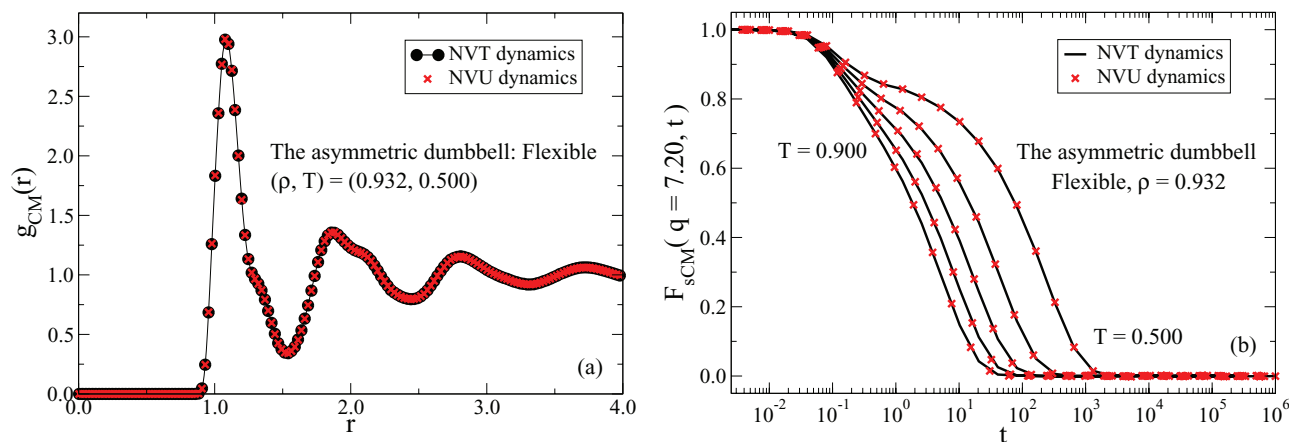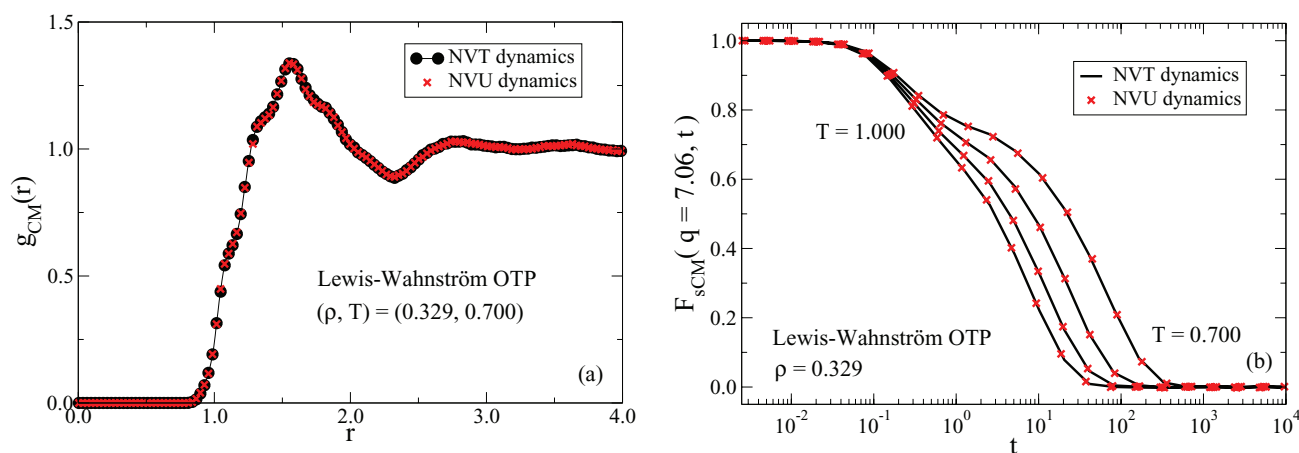
FIG. 3. Comparison of structure and dynamics in *NVU* and *NVT* simulations of the flexible-bond asymmetric dumbbell model. The black circles and curves give *NVT*, the red crosses *NVU* simulation results. The same state points as in Fig. 2 were simulated. (a) The molecular CM radial distribution functions at $\rho = 0.932$ and T = 0.500. (b) The molecular CM incoherent intermediate scattering functions at $\rho = 0.932$ and T = 0.500, 0.600, 0.700, 0.800, 0.900.

FIG. 4. Comparison of center-of-mass structure and dynamics in *NVU* and *NVT* simulations of the Lewis-Wahnström OTP model. The black circles and curves give *NVT*, the red crosses *NVU* simulation results. (a) The molecular CM radial distribution functions at $\rho = 0.329$ and T = 0.700. (b) The molecular CM incoherent intermediate scattering functions at $\rho = 0.329$ and T = 0.700, 0.800, 0.900, 1.000.

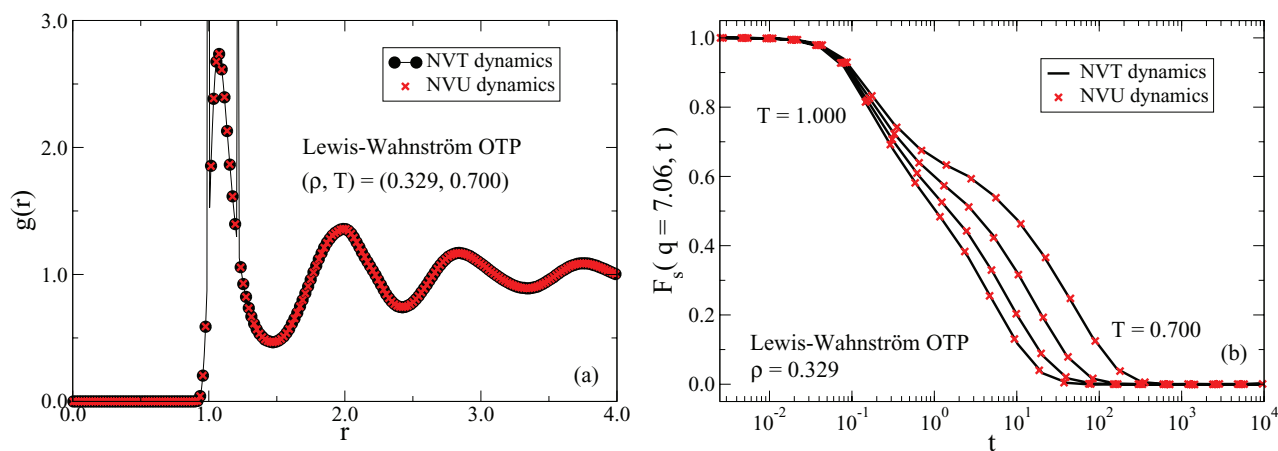FIG. 5. Comparison of particle structure and dynamics in *NVU* and *NVT* simulations for the Lewis-Wahnström OTP model. The black circles and curves give *NVT*, the red crosses *NVU* simulation results. (a) The particle radial distribution functions at $\rho = 0.329$ and T = 0.700. (b) The particle incoherent intermediate scattering functions at $\rho = 0.329$ and T = 0.700, 0.800, 0.900, 1.000.
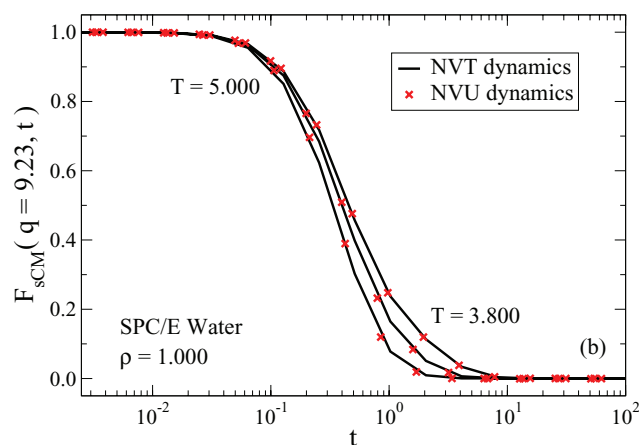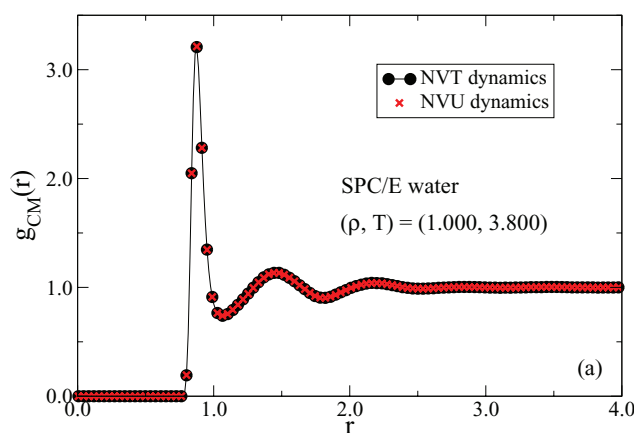
FIG. 6. Comparison of structure and dynamics in *NVU* and *NVT* simulations of rigid SPC/E water. The black circles and curves give *NVT*, the red crosses *NVU* simulation results. (a) The molecular CM radial distribution functions at $\rho = 1.000$ and T = 3.800. (b) The molecular CM incoherent intermediate scattering functions at $\rho = 1.000$ and T = 3.800, 4.200, 5.000.

For reference, we also simulated (Fig. 3) the corresponding quantities for the flexible-bond asymmetric dumbbell model at the state points of Fig. 2. Again, there is a very good agreement between *NVU* and *NVT* dynamics.

B. Lewis-Wahnström OTP

We show in Figs. 4(a) and 4(b), respectively, the molecular CM radial distribution functions and CM incoherent intermediate scattering functions for the Lewis-Wahnström OTP model.[27] The same symbols and meanings as in the preceding section are used. Again, the *NVU* and *NVT* simulations agree very well for both structure and dynamics.

For comparison, we also show in Fig. 5 the corresponding particle quantities for the OTP model.

C. SPC/E water

Finally, we consider in Fig. 6 the same quantities as above for the (not strongly correlating) rigid SPC/E water model.[28]
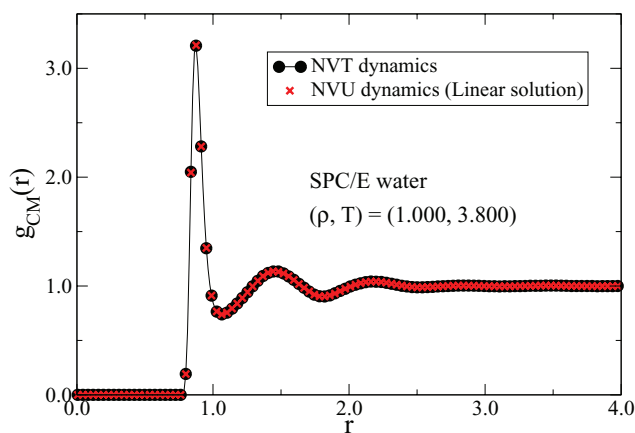

FIG. 7. Comparison of structure in *NVU* and *NVT* simulations of rigid SPC/E water at $\rho = 1.000$ and T = 3.800 applying the linear method to determine the Lagrangian multipliers (Eqs. (29) and (30)). The bond lengths are here conserved to order 10^{-6} in the standard deviation of the bonds (using single-precision).

Again, full equivalence between *NVU* and *NVT* dynamics is found.

The linear algorithm for determining the Lagrangian multipliers presented in Sec. II B (Eqs. (29) and (30)) is tested in Fig. 7 by probing the molecular CM radial distribution functions. *NVU* and *NVT* dynamics also here give identical results.

We conclude from the presented results that for sufficiently large molecular systems with flexible and/or rigid bonds, *NVU* dynamics is equivalent to Nosé-Hoover *NVT* dynamics (and, by implication, to Newtonian *NVE* dynamics).

VI. SUMMARY

NVU dynamics is molecular dynamics at constant potential energy realized by tracing out a geodesic on the constant-potential-energy hypersurface Ω (Eq. (1)). In Papers I and II,[1,2] a "basic" and a "stabilized" atomic *NVU* algorithm for simulating geodesics on Ω were developed. The basic *NVU* algorithm has excellent stability and it is time-reversible and symplectic; the stabilized algorithm was developed only to prevent accumulation of numerical error as also happens for *NVE* dynamics. It was found that atomic *NVU* dynamics becomes equivalent to atomic *NVE* dynamics in the thermodynamic limit.

In this paper, the stabilized *NVU* algorithm has been extended to simulate molecules at constant potential energy. Molecules are generally simulated by introducing rigid and/or flexible bonds in the models. The atomic *NVU* algorithm keeps the potential energy constant and can thus right away simulate flexible bonds. The focus here was on incorporating rigid bonds in the framework of *NVU* dynamics, which leads to the introduction of additional Lagrangian multipliers beyond those of the constraint of constant potential energy. This is completely analogous to the approach for simulating rigid bonds in standard Newtonian *NVE* dynamics.[30,35,36] In the *NVU* algorithm, a set of coupled quadratic equations was constructed for calculating the Lagrangian multipliers and solved in an iterative manner as a linear system, a procedure developed for rigid-bond *NVE* dynamics in the *MILC-SHAKE*

algorithm.[37] In addition, a set of linear equations was presented for calculating the Lagrangian multipliers, which appears to be a promising new way of simulating rigid bonds.

The rigid-bond *NVU* algorithm reduces to the atomic *NVU* algorithm when there are no rigid bonds. The algorithm was tested on three different model systems: the asymmetric dumbbell model, Lewis-Wahnström OTP, and rigid SPC/E water. The probed quantities in the simulation gave identical results to those of Nosé-Hoover *NVT* dynamics. We conclude that also for molecular systems, *NVU* dynamics becomes equivalent to *NVE* dynamics in the thermodynamic limit (since *NVE* and *NVT* dynamics are known to give equivalent results[29]).

ACKNOWLEDGMENTS

APPENDIX: DERIVATION OF THE ATOMIC *NVU* ALGORITHM FOR THE HERTZIAN METRIC

According to Newtonian dynamics, heavy particles move slower than light particles in thermal equilibrium. The standard Euclidean metric does not involve the particle masses, and thus applying this metric to geodesic motion for systems of varying masses will not produce dynamics equivalent to Newtonian dynamics in a thermal system. The mass-weighted metric of Hertz,[32] however, ensures that *NVU* dynamics becomes equivalent to *NVE* dynamics in the thermodynamic limit, as is clear from the derivation below. The Hertzian metric is given by (where $\tilde{m}_k = m_k/\langle m \rangle$)

$$dl^2 \equiv \sum_k \tilde{m}_k (d\mathbf{r}^{(k)})^2. \tag{A1}$$

We here derive the discrete *NVU* algorithm applying this metric (this appendix also corrects a typo in Eq. (A5) of Paper II). The discretized variational condition for geodesic motion on Ω is

$$\delta \left(\sum_i \sqrt{\sum_k \tilde{m}_k (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)})^2} - \sum_i \lambda_i U(\mathbf{R}_i) \right) = 0. \tag{A2}$$

Assuming a constant step length l_0, i.e.,

$$\sum_k \tilde{m}_k (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)})^2 \equiv l_0^2, \tag{A3}$$

it follows by differentiation with respect to $\mathbf{r}_i^{(k)}$ from Eq. (A2) that

$$\tilde{m}_k (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)}) + \tilde{m}_k (\mathbf{r}_i^{(k)} - \mathbf{r}_{i+1}^{(k)}) + l_0 \lambda_i \mathbf{f}_i^{(k)} = 0. \tag{A4}$$

Defining $\mathbf{a}_i^{(k)} \equiv (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)})$ and $\mathbf{b}_i^{(k)} \equiv (\mathbf{r}_i^{(k)} - \mathbf{r}_{i+1}^{(k)})$, Eq. (A3) expresses that $\sum_k \tilde{m}_k ((\mathbf{a}_i^{(k)})^2 - (\mathbf{b}_i^{(k)})^2) = \sum_k \tilde{m}_k (\mathbf{a}_i^{(k)} + \mathbf{b}_i^{(k)}) \cdot (\mathbf{a}_i^{(k)} - \mathbf{b}_i^{(k)}) = 0$, and thus via

Eq. (A4)

$$\sum_k \tilde{m}_k \left(-l_0/\tilde{m}_k \lambda_i \mathbf{f}_i^{(k)} \right) \cdot \left(\mathbf{r}_{i+1}^{(k)} - \mathbf{r}_{i-1}^{(k)} \right) = 0. \tag{A5}$$

Equivalently,

$$\sum_k \mathbf{f}_i^{(k)} \cdot \mathbf{r}_{i+1}^{(k)} = \sum_k \mathbf{f}_i^{(k)} \cdot \mathbf{r}_{i-1}^{(k)}. \tag{A6}$$

Combining Eq. (A6) with the discrete *NVU* algorithm (Eq. (A4)) gives the following result

$$l_0 \lambda_i = \frac{-2 \sum_k \mathbf{f}_i^{(k)} \cdot (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)})}{\sum_k \frac{(\mathbf{f}_i^{(k)})^2}{\tilde{m}_k}}. \tag{A7}$$

The atomic *NVU* algorithm with varying masses is thus given by

$$\mathbf{r}_{i+1}^{(k)} = 2\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)} + \frac{l_0}{\tilde{m}_k} \lambda_i \mathbf{f}_i^{(k)}, \tag{A8}$$

$$l_0 \lambda_i = \frac{-2 \sum_k \mathbf{f}_i^{(k)} \cdot (\mathbf{r}_i^{(k)} - \mathbf{r}_{i-1}^{(k)})}{\sum_k \frac{(\mathbf{f}_i^{(k)})^2}{\tilde{m}_k}}. \tag{A9}$$

Equation (A9) fluctuates relatively less and less as the number of particles increases, and equivalence with *NVE* dynamics is established in the thermodynamic limit.

[1] T. S. Ingebrigtsen, S. Toxvaerd, O. J. Heilmann, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **135**, 104101 (2011) (Paper I).
[2] T. S. Ingebrigtsen, S. Toxvaerd, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **135**, 104102 (2011) (Paper II).
[3] R. M. J. Cotterill, Phys. Rev. B **33**, 262 (1986).
[4] R. M. J. Cotterill and J. U. Madsen, *Characterizing Complex Systems*, edited by H. Bohr (World Scientific, Singapore, 1990), p. 177.
[5] J. L. E. Platt, B. Waszkowycz, R. Cotterill, and B. Robson, Biophys. Chem. **43**, 221 (1992).
[6] R. M. J. Cotterill and J. U. Madsen, J. Phys.: Condens. Matter **18**, 6507 (2006).
[7] A. Scala, L. Angelani, R. D. Leonardo, G. Ruocco, and F. Sciortino, Philos. Mag. B **82**, 151 (2002).
[8] C. Wang and R. M. Stratt, J. Chem. Phys. **127**, 224503 (2007).
[9] C. Wang and R. M. Stratt, J. Chem. Phys. **127**, 224504 (2007).
[10] C. N. Nguyen and R. M. Stratt, J. Chem. Phys. **133**, 124503 (2010).
[11] C. N. Nguyen, J. I. Isaacson, K. B. Shimmyo, A. Chen, and R. M. Stratt, J. Chem. Phys. **136**, 184504 (2012).
[12] N. P. Bailey, U. R. Pedersen, N. Gnan, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **129**, 184507 (2008).
[13] N. P. Bailey, U. R. Pedersen, N. Gnan, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **129**, 184508 (2008).
[14] T. B. Schrøder, N. P. Bailey, U. R. Pedersen, N. Gnan, and J. C. Dyre, J. Chem. Phys. **131**, 234503 (2009).
[15] N. Gnan, T. B. Schrøder, U. R. Pedersen, N. P. Bailey, and J. C. Dyre, J. Chem. Phys. **131**, 234504 (2009).
[16] T. B. Schrøder, N. Gnan, U. R. Pedersen, N. P. Bailey, and J. C. Dyre, J. Chem. Phys. **134**, 164505 (2011).
[17] T. S. Ingebrigtsen, T. B. Schrøder, and J. C. Dyre, J. Phys. Chem. B **116**, 1018 (2012).
[18] T. S. Ingebrigtsen, L. Bøhling, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **136**, 061102 (2012).
[19] T. S. Ingebrigtsen, T. B. Schrøder, and J. C. Dyre, Phys. Rev. X **2**, 011011 (2012).
[20] J. E. Marsden and M. West, Acta Numerica **10**, 357 (2001).
[21] R. Elber, A. Cardenas, A. Ghosh, and H. Stern, Adv. Chem. Phys. **126**, 93 (2003).
[22] A. Lew, "Variational time integrators in computational solid mechanics," Ph.D. dissertation (California Institute of Technology, 2003).
[23] M. West, "Variational integrators," Ph.D. dissertation (California Institute of Technology, 2004).

[24]S. Nosé, J. Chem. Phys. **81**, 511 (1984).

[25]W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).

[26]T. B. Schrøder, U. R. Pedersen, N. P. Bailey, S. Toxvaerd, and J. C. Dyre, Phys. Rev. E **80**, 041502 (2009).

[27]L. J. Lewis and G. Wahnström, Phys. Rev. E **50**, 3865 (1994).

[28]H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma, J. Phys. Chem. **91**, 6269 (1987).

[29]D. J. Evans and B. L. Holian, J. Chem. Phys. **83**, 4069 (1985).

[30]J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[31]H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*, 3rd ed. (Addison-Wesley, 2002).

[32]H. Hertz, "Die Prinzipien der Mechanik," in *Neuem Zusammenhange Dargestellt* (Johann Ambrosius Barth, Leipzig, 1894).

[33]J. Lützen, *Mechanistic Images in Geometric Form: Heinrich Hertz's "Principles of Mechanics"* (Oxford University Press, Oxford, 2005).

[34]M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Science, 1987).

[35]S. Toxvaerd, O. J. Heilmann, T. Ingebrigtsen, T. B. Schrøder, and J. C. Dyre, J. Chem. Phys. **131**, 064102 (2009).

[36]T. Ingebrigtsen, O. J. Heilmann, S. Toxvaerd, and J. C. Dyre, J. Chem. Phys. **132**, 154106 (2010).

[37]A. G. Bailey, C. P. Lowe, and A. P. Sutton, J. Comput. Phys. **227**, 8949 (2008).

[38]S. Toxvaerd and J. C. Dyre, J. Chem. Phys. **134**, 081102 (2011).

[39]All simulations were performed using a molecular dynamics code optimized for *NVIDIA* graphics cards, which is available as open source code at http://rumd.org.

[40]S. Toxvaerd, Mol. Phys. **72**, 159 (1991).

[41]N. Bell and M. Garland, "Cusp: Generic parallel algorithms for sparse matrix and graph computations," (2012), version 0.3.0; available also at http://cusp-library.googlecode.com.

[42]W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd ed. (Cambridge University Press, 2007).

[43]C. J. Fennell and J. D. Gezelter, J. Chem. Phys. **124**, 234104 (2006).

[44]J. S. Hansen, T. B. Schrøder, and J. C. Dyre, J. Phys. Chem. B **116**, 5738 (2012).