Universal integration scheme for MD using Gear formalism

J. Janek

September 14, 2023

Contents

1 Introduction **1**

2 Gear integrators **2**
2.1 From Verlet to Gear . . . 2
2.2 Gear methods . . . 3

3 TRVP in Gear formalism **4**
3.1 TRVP – Time-Reversible Velocity Predictor . . . 4
3.2 Gear formalism . . . 6

4 Constrained dynamics **8**
4.1 SHAKE algorithm . . . 8
4.2 Method of Lagrange multipliers . . . 8
4.3 SHAKE in Gear formalism . . . 10
4.4 SHAKE with higher order Gear integrators . . . 11

5 Universal MD integration scheme **12**

6 Applications and preliminary results **15**
6.1 NVE ensemble . . . 15
6.1.1 Dump-bell . . . 16
6.1.2 Nitrogen . . . 16
6.1.3 SPC/E water . . . 17
6.2 NVT ensemble – Nosé–Hoover thermostat . . . 18
6.2.1 Nitrogen . . . 21
6.3 NPT ensemble – Nosé–Hoover (MTK) barostat . . . 21
6.3.1 Nitrogen . . . 23

7 Conclusion **24**

1 Introduction

The aim of this work is to provide a universal integration scheme for molecular dynamics (MD) simulations in any statistical ensemble. The standard integration method for MD is the Verlet method [1] in different flavors (see [2, 1, 3, 4]). Despite its simplicity, it has many advantages over the other options. Mainly, it is time-reversible, even symplectic (area-preserving) [5, 6],

and thus conserved quantities (namely energy) are really conserved during numerical integration (within the integration error). Being widely used, the Verlet method can be extended to perform simulations in other ensembles than in the microcanonical (NVE). E.g., solving Nosé–Hoover equations of motion, a canonical (NVT) ensemble can be simulated [7, 8]. Similarly, solving the set of equations of Martyna et al. [9, 10], an isothermal-isobaric ensemble can be reproduced. Nevertheless, the extension of the Verlet method for these ensembles is not straightforward and usually involves quite complicated Trotter decomposition of the Liouville operator of the system evolution or iterations to self-consistency. The standard algorithms based on the Trotter decomposition were devised by Martyna et al. [10].

Despite being widely accepted as *the integration method* for MD, the Verlet algorithm is not the only possible. In the early times of MD, another family of numerical integration methods was quite popular (see [3]). In this work, a universal design of MD integration will be proposed, suited both to the Verlet method and the higher-order predictor-corrector (multistep) methods, avoiding the need for two distinct procedures.

2 Gear integrators

2.1 From Verlet to Gear

The Verlet integration method can be written in the form usually called the Velocity Verlet as follows [4, 6]

$$\begin{aligned} \mathbf{r}_i(t+h) &= \mathbf{r}_i(t) + h\dot{\mathbf{r}}_i(t) + \frac{h^2}{2m_i}\mathbf{f}_i(t) \\ \dot{\mathbf{r}}_i(t+h) &= \dot{\mathbf{r}}_i(t) + \frac{h}{2m_i}\left[\mathbf{f}_i(t+h) + \mathbf{f}_i(t)\right] \end{aligned} \tag{1}$$

The second equation can be split into a predictor

$$\dot{\mathbf{r}}_i^{\mathrm{P}}(t+h) = \dot{\mathbf{r}}_i(t) + \frac{h}{m_i}\mathbf{f}_i(t) \tag{2}$$

where the superscript P denotes the *predicted* value, and corrector

$$\dot{\mathbf{r}}_i^{\mathrm{C}}(t+h) = \dot{\mathbf{r}}_i^{\mathrm{P}}(t+h) + \frac{h}{2m_i}\left[\mathbf{f}_i(t+h) - \mathbf{f}_i(t)\right] \tag{3}$$

where, similarly, the superscript C denotes the *corrected* value. Adopting $\ddot{\mathbf{r}}_i^{\mathrm{P}}(t+h) = \ddot{\mathbf{r}}_i^{\mathrm{P}}(t)$ and

$$\mathbf{R}_i(t) = \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \\ h\dot{x}_i(t) & h\dot{y}_i(t) & h\dot{z}_i(t) \\ \frac{h^2}{2}\ddot{x}_i(t) & \frac{h^2}{2}\ddot{y}_i(t) & \frac{h^2}{2}\ddot{z}_i(t) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i^{\mathrm{T}}(t) \\ h\dot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \frac{h^2}{2}\ddot{\mathbf{r}}_i^{\mathrm{T}}(t) \end{bmatrix} \tag{4}$$

the inaccuracy of prediction can be evaluated as the difference between the predicted value of acceleration and the value determined by forces at time $t+h$

$$\mathbf{g}(\mathbf{R}_i^{\mathrm{P}}) = \frac{h^2}{2m_i}\mathbf{f}_i(t+h) - \mathbf{R}_i^{\mathrm{P}}(t+h)^{\mathrm{T}}\delta_2 \tag{5}$$

where forces $\mathbf{f}_i(t+h)$ are computed using positions stored in $\mathbf{R}_i^{\mathrm{P}}(t+h)$ and $\delta_2 = [0,0,1]^{\mathrm{T}}$ (superscript T denotes *transposition*). Adding this difference to the predicted value of acceleration, the corrected acceleration is obtained which is equal to $\mathbf{f}_i(t+h)/m_i$. Note that the difference (5) can also be used to correct the predicted velocity (see (2) and (3)).

Defining the *predictor matrix*

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and the vector of *corrector* coefficients

$$\mathbf{c} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \tag{7}$$

we can write the Velocity Verlet method in the predictor-corrector formalism as follows[a]

$$\begin{aligned} &\text{1. prediction:} && \mathbf{R}_i^{\mathrm{P}}(t+h) &&= \mathbf{A}\mathbf{R}_i^{\mathrm{C}}(t) \\ &\text{2. evaluation of forces:} && \mathbf{g}_i(t+h) &&= \frac{h^2}{2}\frac{\mathbf{f}_i(t+h)}{m_i} - \mathbf{R}_i^{\mathrm{P}}(t+h)^{\mathrm{T}}\delta_2 \\ &\text{3. correction:} && \mathbf{R}_i^{\mathrm{C}}(t+h) &&= \mathbf{R}_i^{\mathrm{P}}(t+h) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t+h) \end{aligned} \tag{8}$$

The described algorithm is completely equivalent to the Velocity Verlet method and can be readily used for MD simulations of systems without constraints and/or velocity-dependent forces in the NVE ensemble.

2.2 Gear methods

In the previous paragraph, we have shown that the Verlet integration method can be reformulated in terms of predictor-corrector integration methods (PEC – prediction, evaluation, correction). Generally, the main idea of these methods is to make use of the previously computed history in order to predict future positions, velocities etc. Then, the predicted coordinates should be corrected using the forces acting on the system.

The prediction can be easily derived from the Taylor expansion to the desired order. The future positions, velocities etc. are given as a linear combination of the previously computed values.

The corrector step has two main purposes:

1. to maintain or improve the order of the method
2. to ensure the stability of the method.

Roughly, the stability of the method means that any error made during the numerical integration does not grow exponentially as the integration proceeds. The derivation of the corrector coefficients for higher-order methods is not straightforward and is described elsewhere (see [11–14]). Essentially, the first two coefficients are responsible for the order of the method and can be modified when, e.g., better energy conservation is needed, the others are chosen in order to achieve the best stability possible.

Each PEC method is thus uniquely defined by the predictor matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & \binom{k-1}{0} \\ & 1 & 2 & 3 & \cdots & \binom{k-1}{1} \\ & & 1 & 3 & \cdots & \binom{k-1}{2} \\ & & & 1 & \cdots & \binom{k-1}{3} \\ & & & & \ddots & \vdots \\ \Large 0 & & & & & \binom{k-1}{k-1} \end{bmatrix} \tag{9}$$

[a] The multiplication $\mathbf{x}\mathbf{y}^{\mathrm{T}}$ is considered as a *matrix-matrix* multiplication (outer product) and the result is thus a matrix.

and the vector of corrector coefficients.

Three classes of methods can be derived. In the original work [14, 13], the coefficients ensuring the highest possible order are given. These methods will be referred to as *the original Gear methods*.

When the forces (the right-hand side of the differential equation) depend explicitly on velocities, the highest order is unachievable. It is therefore convenient to use methods giving better estimates of velocities. This can be, at least theoretically, achieved by using *the velocity versions* of the PEC methods. Formulated originally by Gear [13], they are suggested to use for MD simulations by Allen and Tildesley [6] when forces depend on velocities. They differ from the original methods only in the first corrector coefficient and their order is decreased by one.

The third class was developed recently by us [11, 12]. Modifying the first or first two corrector coefficients, better time-reversibility, and thus energy conservation can be accomplished. Despite being derived only numerically from the solution of the linear harmonic oscillator, the new methods seem to provide better energy conservation than the original ones even for much more complex systems. Moreover, the two versions of the Verlet method formally fit this class of *improved* methods.

The corrector coefficients are given in Table 1. The names of the integrators are determined according to the following key:

1. k denotes the *size* of the method, i.e., the predictor matrix is of size $k \times k$ and the matrix $\mathbf{R}$ stores positions and the first $k-1$ scaled derivatives

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{r}_i^{\mathrm{T}}(t) \\ h\dot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \frac{h^2}{2}\ddot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \vdots \\ \frac{h^{k-1}}{(k-1)!}\mathbf{r}_i^{(k-1)\ \mathrm{T}}(t) \end{bmatrix} \tag{10}$$

2. m is the error order of the method (for the definition see [12])
3. the *original* (nothing), *velocity* (`v`) and *improved* (`e`) methods are distinguished by the last letter.

The overall algorithm for the NVE simulation was described above (see equation (8)).

3 TRVP in Gear formalism

3.1 TRVP – Time-Reversible Velocity Predictor

In its original formulation, the Verlet algorithm does not provide the value or velocity at time $t+h$ when forces at $t+h$ are to be computed. The second equation in (1) becomes implicit when forces depend on velocities. Therefore, special methods must be used, e.g., in the case of the Nosé–Hoover thermostat or barostat.

One and the most common remedy for this is to use the Trotter expansion of the Liouville operator. This is the case of the standard MTTK integration schemes [10]. Their main disadvantage is their complexity and the need for a specially tailored integration scheme for each ensemble.

The second option is to use iterations until self-consistency. Having considered that the position is not changed during the corrector part of the Verlet method (for higher-order methods this is not true), the correction can be performed repeatedly without the need of recalculating

Table 1: Corrector coefficients for the most stable methods of size k, global error order m and time-reversibility order n.

Method	m	n	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
`k3m3`	3	3	$\frac{1}{6}$	1	1						
`k3m2v`	2	3	$\frac{1}{3}$	1	1						
`k3m2e`*	2	∞	0	1	1						
`k4m4`	4	5	$\frac{1}{6}$	$\frac{5}{6}$	1	$\frac{1}{3}$					
`k4m3v`	3	3	$\frac{1}{4}$	$\frac{5}{6}$	1	$\frac{1}{3}$					
`k4m2e`†	2	∞	0	$\frac{2}{3}$	1	$\frac{1}{3}$					
`k5m5`	5	5	$\frac{19}{120}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$				
`k5m4v`	4	5	$\frac{19}{90}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$				
`k5m4e`	4	7	$\frac{1}{12}$	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$				
`k5m4s`	4	5	0	$\frac{3}{4}$	1	$\frac{1}{2}$	$\frac{1}{12}$				
`k6m6`	6	7	$\frac{3}{20}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$			
`k6m5v`	5	5	$\frac{3}{16}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$			
`k6m4e`	4	9	$\frac{1}{30}$	$\frac{23}{36}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$			
`k6m4s`	4	7	0	$\frac{56}{90}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$			
`k7m7`	7	7	$\frac{863}{6048}$	$\frac{95}{144}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$		
`k7m6v`	6	7	$\frac{863}{5040}$	$\frac{95}{144}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$		
`k7m6e`	6	9	$\frac{7}{72}$	$\frac{95}{144}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$		
`k7m6s`	6	7	0	$\frac{95}{144}$	1	$\frac{25}{36}$	$\frac{35}{144}$	$\frac{1}{24}$	$\frac{1}{360}$		
`k8m8`	8	9	$\frac{275}{2016}$	$\frac{19087}{30240}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$	
`k8m7v`	7	7	$\frac{275}{1728}$	$\frac{19087}{30240}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$	
`k8m6e`	6	11	$\frac{43}{840}$	$\frac{217}{360}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$	
`k8m6s`	6	9	0	$\frac{123}{210}$	1	$\frac{137}{180}$	$\frac{5}{16}$	$\frac{17}{240}$	$\frac{1}{120}$	$\frac{1}{2520}$	
`k9m9`	9	9	$\frac{33953}{259200}$	$\frac{5257}{8640}$	1	$\frac{49}{60}$	$\frac{203}{540}$	$\frac{49}{480}$	$\frac{7}{432}$	$\frac{1}{720}$	$\frac{1}{20160}$
`k9m8v`	8	9	$\frac{33953}{226800}$	$\frac{5257}{8640}$	1	$\frac{49}{60}$	$\frac{203}{540}$	$\frac{49}{480}$	$\frac{7}{432}$	$\frac{1}{720}$	$\frac{1}{20160}$
`k9m8e`	8	11	$\frac{859}{8640}$	$\frac{5257}{8640}$	1	$\frac{49}{60}$	$\frac{203}{540}$	$\frac{49}{480}$	$\frac{7}{432}$	$\frac{1}{720}$	$\frac{1}{20160}$
`k9m8s`	8	9	0	$\frac{5257}{8640}$	1	$\frac{49}{60}$	$\frac{203}{540}$	$\frac{49}{480}$	$\frac{7}{432}$	$\frac{1}{720}$	$\frac{1}{20160}$

* the Velocity Verlet method [1]

† Beeman's version of the Verlet method [3]

the physical forces depending on positions. However, for the higher-order methods, the iterations would require this recalculation.

Another approach was developed by Kolafa and Lísal [15]. Instead of gradually changing the temperature in half- and quarter-steps, as in the MTTK scheme, or iterating to self-consistency, the future value is predicted by a time-reversible velocity predictor (TRVP). If done appropriately, this leads to a much simpler algorithm that provides results comparable to the standard methods.

The velocity in time $t + h$ is predicted using the following formula

$$h\dot{\mathbf{r}}_i^{\mathrm{TRVP}}(t+h) = \sum_{j=0}^{k} B_j \left[\mathbf{r}_i(t-(j-1)h) - \mathbf{r}_i(t-jh)\right] \tag{11}$$

where k is the *predictor length* and the coefficients are defined recursively as

$$\begin{aligned} B_0 &= \frac{2k+1}{k+1} \\ B_j &= -B_{j-1}\frac{k+1-j}{k+1+j} \quad \text{for } j > 0 \end{aligned} \tag{12}$$

It was shown that for practical purposes $k = 2$ is sufficient [15, 2].

3.2 Gear formalism

The idea of velocity prediction is inherently present in the PEC methods. Hence, it is not difficult to incorporate the TRVP algorithm. Two main modifications must be made.

1. Matrix $\mathbf{R}_i$ holding the coordinates of atom i must be enlarged to store also the previous positions, or better their differences.
2. Predictor matrix $\mathbf{A}$ must be extended by the TRVP coefficients.

Let the new matrix $\mathbf{R}_i$ be

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{r}_i^{\mathrm{T}}(t) \\ h\dot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \frac{h^2}{2}\ddot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \vdots \\ \frac{h^{k-1}}{(k-1)!}\mathbf{r}_i^{(k-1)\ \mathrm{T}}(t) \\ (h\dot{\mathbf{r}}_i^{\mathrm{TRVP}}(t))^{\mathrm{T}} \\ \mathbf{r}_i^{\mathrm{T}}(t) - \mathbf{r}_i^{\mathrm{T}}(t-h) \\ \mathbf{r}_i^{\mathrm{T}}(t-h) - \mathbf{r}_i^{\mathrm{T}}(t-2h) \\ \vdots \\ \mathbf{r}_i^{\mathrm{T}}(t-jh) - \mathbf{r}_i^{\mathrm{T}}(t-(j+1)h) \end{bmatrix} \tag{13}$$

Then, the extended predictor matrix must be

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & \cdots & \binom{k-1}{0} & & & & & & \\ & 1 & 2 & \cdots & \binom{k-1}{1} & & & & \mathbf{0} & & \\ & & 1 & \cdots & \binom{k-1}{2} & & & & & & \\ & & & \cdots & \binom{k-1}{3} & & & & & & \\ & \mathbf{0} & & \ddots & \vdots & & & & & & \\ & & & & \binom{k-1}{k-1} & & & & & & \\ 0 & B_0 & B_0 & \cdots & B_0 & 0 & B_1 & B_2 & B_3 & \cdots & B_j \\ 0 & 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ & & & & & 0 & 1 & 0 & 0 & \cdots & 0 \\ & & & & & 0 & 0 & 1 & 0 & \cdots & 0 \\ & \mathbf{0} & & & & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ & & & & & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \tag{14}$$

where k is the original size of the PEC method and j is the length of the velocity predictor.

For clarity, we will write down these matrices for the particular case of the Verlet method with TRVP of length 2. The above general formulae (13) and (14) become

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_i^{\mathrm{T}}(t) \\ h\dot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ \frac{h^2}{2}\ddot{\mathbf{r}}_i^{\mathrm{T}}(t) \\ (h\dot{\mathbf{r}}_i^{\mathrm{TRVP}}(t))^{\mathrm{T}} \\ \mathbf{r}_i^{\mathrm{T}}(t) - \mathbf{r}_i^{\mathrm{T}}(t-h) \\ \mathbf{r}_i^{\mathrm{T}}(t-h) - \mathbf{r}_i^{\mathrm{T}}(t-2h) \end{bmatrix} \tag{15}$$

and

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & B_0 & B_0 & 0 & B_1 & B_2 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{16}$$

Finally, the vector of corrector coefficients must be augmented to

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}^{\mathrm{orig}} \\ \hline 0 \\ c_0 \\ 0 \\ \vdots \end{bmatrix} \tag{17}$$

Particularly, for the Verlet method with TRVP of length 2

$$\mathbf{c} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

The resulting matrices are larger and their multiplications and additions can take longer time even though many of their elements are zero. Nevertheless, the prediction is still much

faster than the calculation of forces which is always the bottleneck of any MD simulation. The main advantage of the proposed approach is the simplicity of implementation. Once filled with correct values, matrices $\mathbf{A}$, $\mathbf{R}$ and vector $\mathbf{c}$ can be handled exactly the same way disregarding the particular method. This offers the possibility of having only one general integrator for simulations in whatever ensemble. However, one step is still missing to reach the goal.

4 Constrained dynamics

4.1 SHAKE algorithm

Treating systems with constraints, mainly with constrained bonds, an extension to the integration method is needed that introduces evaluation of some fictitious (in the sense of non-physical, not derived from any potential) *constraining forces*. The main objective of such forces is to keep the length of constrained bonds constant during the whole simulation. In order to get physical results, they should be calculated in such a way that the total momentum is conserved.

The most widely spread method of handling constrained bonds in MD was introduced in 1977 by Ryckaert et al. [16]. Their SHAKE algorithm, originally intended for the leap-frog version of the Verlet algorithm, was later extended by Andersen [17] to fit the Velocity Verlet integration scheme (this version is also known as RATTLE). The constraining forces ($\mathbf{f}_{\mathrm{c},i}$) are calculated by

$$\begin{aligned} \mathbf{r}_{ij}(t) &= \mathbf{r}_j(t) - \mathbf{r}_i(t) \\ \mathbf{r}_{ij}^{\mathrm{Verlet}}(t+h) &= \mathbf{r}_j^{\mathrm{Verlet}}(t+h) - \mathbf{r}_i^{\mathrm{Verlet}}(t+h) \\ \lambda &= \frac{\left|\mathbf{r}_{ij}^{\mathrm{Verlet}}(t+h)\right|^2 - l^2}{2l^2} \\ \mathbf{f}_{\mathrm{c},ij}(t) &= \lambda \frac{\mu_{ij}}{h^2} \mathbf{r}_{ij}(t) \end{aligned} \tag{19}$$

where l is the length of the constrained bond between atoms i and j and μ_{ij} is the their reduced mass. The constraining forces are then added to the physical forces at time t. Note, that forces correcting the positions at time $t+h$ are calculated in the direction of bonds at time t and are effectively added to forces acting at time t. The result is the same as if the positions at time $t+h$ calculated by the normal Verlet algorithm ($\mathbf{r}_i^{\mathrm{Verlet}}(t+h)$) were corrected in the following way [18]

$$\begin{aligned} \mathbf{r}_i(t+h) &= \mathbf{r}_i^{\mathrm{Verlet}}(t+h) + \lambda \frac{m_j}{m_i+m_j} \mathbf{r}_{ij}(t) \\ \mathbf{r}_j(t+h) &= \mathbf{r}_j^{\mathrm{Verlet}}(t+h) - \lambda \frac{m_i}{m_i+m_j} \mathbf{r}_{ij}(t) \end{aligned} \tag{20}$$

When multiple constrained bonds are present, the SHAKE algorithm must be applied on each bond successively and iterated to self-consistency to get correct bond lengths and achieve good time-reversibility.

4.2 Method of Lagrange multipliers

The SHAKE method was developed exclusively for the Verlet integrator. In case we would like to use another integration method, another way of computing constraining forces must be used. Starting from the first principles, we can proceed from the definition of the system's Lagrangian to the set of differential equations of motion then describe the evolution of the system with constrained bonds [19].

Let us define the set of constraints (see [18, 2])

$$c_\alpha(\mathbf{r}^N) = \left(\mathbf{r}_{j(\alpha)} - \mathbf{r}_{i(\alpha)}\right)^2 - l_\alpha^2 \tag{21}$$

then, the Lagrangian of the system is

$$\mathcal{L} = \sum_{i=1}^{N} \frac{m_i}{2} \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \sum_{\alpha=1}^{N_{\text{constr}}} \lambda_\alpha c_\alpha \tag{22}$$

where λ_α are the hitherto unknown Lagrangian multipliers. From the Lagrangian, we can derive the equations of motion in the form

$$\ddot{\mathbf{r}}_i = \frac{\mathbf{f}_i}{m_i} + \frac{1}{m_i} \sum_{\alpha=1}^{N_{\text{constr}}} \lambda_\alpha \frac{\partial c_\alpha}{\partial \mathbf{r}_i} \tag{23}$$

Again, we see that the size of constraining forces is given by λ_α for each bond. To evaluate their value, we can first multiply the equation (23) by $\frac{\partial c_\beta}{\partial \mathbf{r}_i}$ and sum over all atoms (over i)

$$\sum_{i=1}^{N} \ddot{\mathbf{r}}_i \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} = \sum_{i=1}^{N} \frac{\mathbf{f}_i}{m_i} \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} + \frac{1}{m_i} \sum_{\alpha=1}^{N_{\text{constr}}} \sum_{i=1}^{N} \lambda_\alpha \frac{\partial c_\alpha}{\partial \mathbf{r}_i} \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} \tag{24}$$

Now, let us differentiate c_β with respect to time to get

$$\frac{\mathrm{d}c_\beta}{\mathrm{d}t} = \sum_{i=1}^{N} \dot{\mathbf{r}}_i \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} = 0 \tag{25}$$

$$\frac{\mathrm{d}^2 c_\beta}{\mathrm{d}t^2} = \sum_{i=1}^{N} \ddot{\mathbf{r}}_i \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} + \sum_{i=1}^{N} \sum_{j=1}^{N} \dot{\mathbf{r}}_i \cdot \frac{\partial^2 c_\beta}{\partial \mathbf{r}_i \partial \mathbf{r}_j} \cdot \dot{\mathbf{r}}_j = 0 \tag{26}$$

Both these derivatives must be zero since all the constraints are conformed at any time. Comparing equations (24) and (26) and defining matrix $\mathbf{M}$ and vector $\mathbf{s}$ with the following entries

$$\begin{aligned} M_{\beta\alpha} &= \sum_{i=1}^{N} \frac{1}{m_i} \frac{\partial c_\beta}{\partial \mathbf{r}_i} \cdot \frac{\partial c_\alpha}{\partial \mathbf{r}_i} \\ s_\beta &= \sum_{i=1}^{N} \frac{\mathbf{f}_i}{m_i} \cdot \frac{\partial c_\beta}{\partial \mathbf{r}_i} + \sum_{i=1}^{N} \sum_{j=1}^{N} \dot{\mathbf{r}}_i \cdot \frac{\partial^2 c_\beta}{\partial \mathbf{r}_i \partial \mathbf{r}_j} \cdot \dot{\mathbf{r}}_j \end{aligned} \tag{27}$$

we finally arrive at the set of linear equations for λ_α

$$\mathbf{M}\lambda + \mathbf{s} = 0 \tag{28}$$

Matrix $\mathbf{M}$ is sparse and the solution can be obtained using, e.g., the conjugate gradient method. It is advantageous to predict the solution from the previous steps to minimise the number of iterations needed.

The method described above determines the forces needed to keep the bonds fixed. However, the integration is not precise and errors tend to accumulate. Therefore, corrections must be made to make the integration stable. The corrections can be written in the form

$$\begin{aligned} \mathbf{r}_{i,\text{corr}} &= \mathbf{r}_i + \Delta\mathbf{r}_i \\ \Delta\mathbf{r}_i &= \sum_{\alpha=1}^{N_{\text{constr}}} \frac{1}{m_i} \frac{\partial c_\alpha}{\partial \mathbf{r}_i} e_\alpha \end{aligned} \tag{29}$$

where e_α are the unknown coefficients. Since the corrections are supposed to be small, we can write the Taylor expansion of constraints to the first order

$$c_\beta\left(\mathbf{r}_{\text{corr}}^N\right) = c_\beta\left(\mathbf{r}^N\right) + \sum_{i=1}^{N} \frac{\partial c_\beta}{\partial \mathbf{r}_i} \cdot \Delta\mathbf{r}_i = 0 \tag{30}$$

Substituting $\Delta\mathbf{r}_i$ from (29) we get the set of linear equations for e_α

$$\mathbf{M}\mathbf{e} + \mathbf{w} = 0 \tag{31}$$

where $w_\alpha = c_\alpha \left(\mathbf{r}^N_{\text{corr}}\right)$. Corrections to velocities are made similarly.

This method, as well as the SHAKE method, is constructed in such a way that the momenta of molecules are conserved. However, due to the additional corrections, both to positions and velocities, which do not propagate to higher derivatives stored in $\mathbf{R}$, this algorithm breaks the consistency of the Gear methods. Moreover, the constraining forces depend on velocities (see (27)), thus the improved Gear methods, being derived under an assumption of velocity independent forces [11], provide no better time-reversibility than the original methods.

The first remedy that comes to mind is to calculate physical forces in the unconstrained system first. Then, the constraining forces can be made as big (their direction is given by the direction of bonds) as needed to ensure correct bond lengths. Except for their origin, the constraining forces should be treated like the physical forces, i.e. they should contribute to $\mathbf{g}(\mathbf{R}^{\mathrm{P}}_i)$. Nevertheless, this approach is unusable for `k3m2e` (and also for `k4m2e`) method, because the corrector coefficient c_0 is zero, so the correction step cannot affect positions. Furthermore, it is of no use even for the improved Gear methods, because the corresponding c_0 are generally quite small. Accordingly, the constraining forces must be large and the drafted method is highly unstable.

Fortunately, a better method can be derived by customizing the SHAKE method to use along with the PEC methods.

4.3 SHAKE in Gear formalism

In subsection 2.1 we have shown, how the traditional Verlet method can be rewritten to the PEC integration scheme. We should also be able to fit the SHAKE procedure into this formalism. Note that in the SHAKE method, the constraining forces acting at time t are parallel with rigid bonds at time t, but their size is actually calculated from predicted positions at time $t+h$. Hence, the positions both at time t and $t+h$ are needed at once. Having considered that in the Verlet-equivalent `k3m2e` method, positions at time $t+h$ are easily calculated from $\mathbf{R}_i(t)$, we can formulate the Verlet+SHAKE method in the PEC formalism (note the time-shift in comparison with (8), blue parts apply in case of SHAKE iterations)

$$\begin{aligned}
&\text{1. prediction:} && \mathbf{R}^{\mathrm{P}}_i(t) && = \mathbf{A}\mathbf{R}^{\mathrm{C}}_i(t-h) \\
&\text{2. evaluation:} && \mathbf{g}_i(t) && = \frac{h^2}{2}\frac{\mathbf{f}_i(t)}{m_i} - \mathbf{R}^{\mathrm{P}}_i(t)^{\mathrm{T}}\delta_2 \\
&\text{3. } \textit{normal} \text{ correction:} && \mathbf{R}^{\mathrm{I}}_i(t) && = \mathbf{R}^{\mathrm{P}}_i(t) + \mathbf{c}\mathbf{g}^{\mathrm{T}}_i(t) + \mathbf{c}\mathbf{h}^{\mathrm{T}}_i(t) \\
&\text{4. SHAKE:} && \mathbf{R}^{\mathrm{PI}}_i(t+h) && = \mathbf{A}\mathbf{R}^{\mathrm{I}}_i(t) \\
& && \lambda && = \frac{\left|\mathbf{r}^{\mathrm{PI}}_{ij}(t+h)\right|^2 - l^2}{2l^2} \\
& && \mathbf{h}_i(t) && \mathrel{+}= \frac{1}{\sum_i c_i}\lambda\frac{m_j}{m_i+m_j}\mathbf{r}^{\mathrm{P}}_{ij}(t) \\
&\text{5. final correction:} && \mathbf{R}^{\mathrm{C}}_i(t) && = \mathbf{R}^{\mathrm{P}}_i(t) + \mathbf{c}\mathbf{g}^{\mathrm{T}}_i(t) + \mathbf{c}\mathbf{h}^{\mathrm{T}}_i(t)
\end{aligned} \tag{32}$$

Here, $\mathbf{r}^{\mathrm{P}}_{ij}$ is the interatomic distance between atoms i and j calculated from positions stored in $\mathbf{R}^{\mathrm{P}}_i$ (predicted value) and $\mathbf{r}^{\mathrm{PI}}_{ij}(t+h)$ is the interatomic distance calculated from positions stored in $\mathbf{R}^{\mathrm{PI}}_i(t+h)$, i.e. *predicted* from $\mathbf{R}^{\mathrm{I}}_i(t)$ (values after *normal* correction).[b] The constrained

[b] In fact, there is no need to perform the whole predictor step with $\mathbf{R}^{\mathrm{I}}_i$ to get $\mathbf{R}^{\mathrm{PI}}_i$. It can be noticed, that the prediction of the position means only sum one column of $\mathbf{R}^{\mathrm{I}}_i$. Moreover, not even $\mathbf{R}^{\mathrm{I}}_i$ need to be computed

error $\mathbf{h}_i$ is very similar to the correction term in (20) but it is divided by the sum of corrector coefficients because the correction should be distributed to all terms in $\mathbf{R}_i$ to be gathered again in the predictor step.

The above algorithm described verbally:

1. Calculate predictor.
2. Calculate physical forces from predicted values.
3. Calculate constraining forces so that the bond lengths will be satisfied after the next prediction.
4. Correct the predicted values using *both types of* forces.

It may seem that handling point 3 is quite complicated, but it can be done quite easily with a little care.

If more interconnected bonds are present, this algorithm requires iterations like the original SHAKE. The repeating steps are the recalculation of $\mathbf{R}_i^{\mathrm{I}}(t)$ and $\mathbf{r}_{ij}^{\mathrm{PI}}(t+h)$, evaluation of λ and $\mathbf{h}_i$ in which the corrections are cumulated, see (32), steps 3 & 4. Comparing (32) with (19) and (20), the equivalence of the above method with the `k3m2e` corrector and the Verlet+SHAKE algorithm can be seen.

To calculate pressure, the contribution of constraining forces to the atomic virial is needed. Comparing $\mathbf{g}_i$ and $\mathbf{h}_i$ in (32), the following formula can be derived (the sum is over all atom pairs connected by a rigid bond).

$$\begin{aligned} W_{\mathrm{constr}} &= \sum_{\alpha}^{N_\alpha} \mathbf{f}_\alpha(t) \cdot \mathbf{r}_{ij}^{\mathrm{P}}(t) \\ &= \sum_{\alpha}^{N_\alpha} \frac{2\lambda_\alpha}{h^2 \sum_i c_i} \frac{m_i m_j}{m_i + m_j} \mathbf{r}_{ij}^{\mathrm{P}}(t) \cdot \mathbf{r}_{ij}^{\mathrm{P}}(t) \end{aligned} \tag{33}$$

4.4 SHAKE with higher order Gear integrators

The main difference between the Verlet method rewritten in the PEC formalism (`k3m2e` and `k4m2e`) and the other Gear methods is that while for the former it holds $c_0 = 0$ and the predicted position $\mathbf{r}_i^{\mathrm{P}}$ is thus not further corrected, for the latter the position is slightly modified during the *correction* step. The key point here is that the correction is small and presumably does not cause instability of the method. The correction of position is small for three reasons:

1. The prediction is the more precise, the higher the order of the method.
2. The first corrector coefficient is small and for the improved methods it is even smaller than for the original.
3. The constrained error $\mathbf{h}_i$ is distributed among all terms stored in $\mathbf{R}_i$.

Consequently, the constraints are not precisely fulfilled after the correction, but the error is sufficiently small and does not accumulate during the integration. The resulting algorithm is the same as the algorithm defined above (see (32)).

---

explicitly. The prediction can be made from $\mathbf{R}_i^{\mathrm{P}}$ (summing its columns) and prediction error $\mathbf{g}_i$ added multiplied by the sum of corrector coefficients like

$$\mathbf{r}_i^{\mathrm{PI}}(t+h) = \underbrace{[1, 1, \ldots, 1]}_{k} \mathbf{R}_i^{\mathrm{P}}(t) + \sum_j c_j \mathbf{g}_i(t)$$

5 Universal MD integration scheme

In the previous sections, we have shown how to overcome all major obstacles when designing the universal-purpose integrator for MD in whatever ensemble. Now, only the final step, i.e. the formulation of such an integration scheme, is missing. Let us once more summarise the ingredients of the MD integrator.

1. The *core* integrator, i.e. integrator for the NVE ensemble
2. The SHAKE or SHAKE-like algorithm for correcting constraints (if present)
3. The integrator of extended degrees of freedom (if needed)
4. The velocity predictor (in case of velocity-dependent RHS)

Not all of them are always needed, but the universal scheme should contain them all. If not needed, they can be easily skipped.

It can be convenient to choose the same integrator for both the physical and the extended degrees of freedom, but it is not strictly necessary. Note, that, e.g. in the MTTK algorithm, the extended degrees of freedom are integrated with smaller time steps and can easily be integrated also by a higher order integrator [10, 20]. Nevertheless, we will assume here that we have chosen the same method for all degrees of freedom. The extension is obvious.

Hence, the integration scheme is (see [15, 2])

1. **PREDICTION.** The prediction of both the physical and the extended degrees of freedom. This step includes TRVP for all degrees of freedom (if needed).

$$\begin{aligned} \mathbf{R}_i^{\mathrm{P}}(t) &= \mathbf{A}\mathbf{R}_i^{\mathrm{C}}(t-h) \\ \boldsymbol{\xi}^{\mathrm{P}}(t) &= \mathbf{A}\boldsymbol{\xi}^{\mathrm{C}}(t-h) \\ \boldsymbol{\Lambda}^{\mathrm{P}}(t) &= \mathbf{A}\boldsymbol{\Lambda}^{\mathrm{C}}(t-h) \end{aligned} \tag{34}$$

   where ξ is the extended degree of freedom coupled with the thermostat, e.g. in the Nosé–Hoover NVT integration scheme, and $\Lambda = \ln L$ where L is the size of the simulation box entering the equations of motion, e.g., in Nosé–Hoover [21] (or MTK [10]) NPT ensemble. In vectors $\boldsymbol{\xi}$ and $\boldsymbol{\Lambda}$, the scaled derivatives of ξ or Λ are stored similarly to $\mathbf{R}_i$ containing the scaled derivatives of $\mathbf{r}_i$. In the most complicated case (considered) when the simulation box is not cubic, $\boldsymbol{\Lambda}$ is a matrix with entries ($\Lambda_x = \ln L_x$, etc.). Here we consider a simple cubic box and define (the extension should be obvious)

$$\boldsymbol{\Lambda} = \begin{bmatrix} \Lambda \\ h\dot{\Lambda} \\ \frac{h^2}{2}\ddot{\Lambda} \\ \vdots \\ \frac{h^{k-1}}{(k-1)!}\Lambda^{(k-1)} \end{bmatrix} \tag{35}$$

   When the volume of the system fluctuates (e.g. in the previously mentioned Nosé–Hoover NPT ensemble), the time-derivative of position $\mathbf{r}_i$ is not exactly equal to the velocity $\mathbf{v}_i$ which is used to measure the kinetic temperature of the system. Typically, the differential equations for $\mathbf{r}_i$ and $\mathbf{v}_i$ looks like

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{v}_i + \dot{\Lambda}\mathbf{r}_i \\ \dot{\mathbf{v}}_i &= \frac{\mathbf{f}_i}{m_i} + \cdots \end{aligned} \tag{36}$$

However, Gear's formulation forces us to formulate the equations of motion as second-order differential equations, not as the system of two differential equations of the first order. Two different approaches are possible. The first is to take the second derivative of $\mathbf{r}_i$ and substitute the equation for $\mathbf{v}_i$ into it. The integration would be straightforward but it would be difficult to extract the velocity $\mathbf{v}_i$ needed to calculate the temperature. The second, and in our opinion better, solution is to solve the equation as if $\dot{\mathbf{r}}_i = \mathbf{v}_i$ and substitute the second part of the equation by direct rescaling of the simulation box. Here, we will follow this way and rescale the box by

$$\exp\left(\Lambda^{\mathrm{P}}(t) - \Lambda^{\mathrm{C}}(t-h)\right) \tag{37}$$

This scaling is applied to all atomic positions and the size of the simulated periodic box. The velocities stored in $\mathbf{R}_i$ thus remain intact.

2. **EVALUATION.** The calculation of the right-hand side (RHS) of the second-order differential equation for physical degrees of freedom (forces, accelerations) and of the error of prediction. The potential and kinetic energy of the physical (not extended) degrees of freedom are calculated during this step.

   In case the RHS depends on velocities, the correct (*predicted*) velocities must be used. Thus, if the Verlet integrator with TRVP is to be used, the TRVP velocities have to be used instead of the velocities stored in the second row of $\mathbf{R}_i$. Usually, there is no reason to use TRVP with a higher-order integrator, but it is possible if desired.

   This is the *evaluation* part of the PEC scheme. However, note that only the physical accelerations are computed and not the second derivatives of the extended degrees of freedom.

$$\mathbf{g}_i(t) = \frac{h^2}{2}\mathrm{RHS}\left(\mathbf{R}_i^{\mathrm{P}}(t)\right) - \mathbf{R}_i^{\mathrm{P}}(t)^{\mathrm{T}}\delta_2 \tag{38}$$

   E.g., for the NVE ensemble the right-hand side is

$$\mathrm{RHS}\left(\mathbf{R}_i^{\mathrm{P}}(t)\right) = \frac{\mathbf{f}_i(t)}{m_i} \tag{39}$$

   where $\mathbf{f}_i(t)$ is the force acting on atom i at time t calculated from positions in $\mathbf{R}_i^{\mathrm{P}}(t)$. As mentioned previously, the equation for the derivative of $\mathbf{v}_i$ should be used in the case of a fluctuating box, and not the equation for $\ddot{\mathbf{r}}_i$.

3. **CORRECTION 1** (optional). The intermediate correction is needed only if any constraints are to be treated by the SHAKE method. In all cases, only positions must be corrected, the correction of the derivatives can be left to the final correction step.

$$\mathbf{R}_i^{\mathrm{I}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) \tag{40}$$

   As already mentioned, $\mathbf{R}_i^{\mathrm{I}}$ need not be computed explicitly. Prediction of position $\mathbf{r}_i^{\mathrm{PI}}(t+h)$ (needed for the next step) can be made from $\mathbf{R}_i^{\mathrm{P}}$ and prediction error $\mathbf{g}_i$ added together and multiplied by the sum of corrector coefficients.

4. **SHAKE** (optional). In this place, the iterative SHAKE procedure should be used to correct all constrained bonds. The algorithm is described in subsection 4.3 in detail.

   If the simulation box is changing, it is necessary to know its size both at time t and $t+h$ (scaling factor σ). If the latter is not known in advance (it can be known, e.g. in the direct simulation of expansion [22]) further prediction of Λ may be needed. In fact,

this prediction from predicted values is conceptually quite similar to the MTK barostat algorithm implementation in MACSIMUS.

$$\begin{aligned}\left(\mathbf{R}_i^{\mathrm{PI}}(t+h) &= \mathbf{A}\mathbf{R}_i^{\mathrm{I}}(t)\right)\\ \mathbf{\Lambda}^{\mathrm{PI}}(t+h) &= \mathbf{A}\mathbf{\Lambda}^{\mathrm{P}}(t)\\ \sigma &= \exp(\Lambda^{\mathrm{PI}}(t+h) - \Lambda^{\mathrm{P}}(t))\\ \lambda &= \frac{\left|\sigma\mathbf{r}_{ij}^{\mathrm{PI}}(t+h)\right|^2 - l^2}{2l^2}\\ \mathbf{h}_i(t) &= \frac{1}{\sum_i c_i}\lambda\frac{m_j}{m_i+m_j}\mathbf{r}_{ij}^{\mathrm{P}}(t)\end{aligned} \tag{41}$$

The predicted scaling only approximates the true scaling if the volume does not change rapidly. The errors in the lengths of constrained bonds are bigger if the correlation time of barostat τ_P is smaller.

5. **CORRECTION 2.** The physical degrees of freedom can eventually be corrected using the error of prediction $\mathbf{g}_i(t)$ from the *evaluation* step and optionally $\mathbf{h}_i(t)$ from SHAKE. In case of the absence of extended degrees of freedom, this is the final part of the integration step.

$$\mathbf{R}_i^{\mathrm{C}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) + \mathbf{c}\mathbf{h}_i^{\mathrm{T}}(t) \tag{42}$$

6. **EVALUATION & CORRECTION** for the **EXTENDED DOFs**. Except for the *prediction* phase, we have so far omitted the extended degrees of freedom from the integration and used their predicted values if needed. In the final step, the error of prediction must be computed from the differential equations of motion for the extended degrees of freedom and then used to correct their predicted values.

   A choice must be made about which extended degree should be corrected first. If the differential equations are interconnected, the choice can be ambiguous. Here, we suggest calculating the correction for $\boldsymbol{\xi}$ (the thermostat variable) before $\mathbf{\Lambda}$ (the size of the simulation box) as in [2], but further discussion may be useful. However, it is definitely reasonable to correct one before evaluating the error of prediction of the other.

   If the volume of the simulated system is not fixed (Λ changes during the simulation), we should now again rescale the simulation box to the corrected value. Nevertheless, if we use the Verlet-equivalent methods (`k3m2e` or `k4m2e`) the corrected value is the same as the predicted one and this rescaling step can be omitted. Actually, it is strictly needed only before measurement as we can postpone the rescaling to the prediction step if we do not mind the inaccuracy yielding from the fact that after prediction we are scaling also the values of derivatives...

7. **MEASUREMENT**. Optional, usually once per a fixed number of steps (e.g. 10). All measured quantities can be recalculated (kinetic energy, the energy of the extended DOFs, etc.). Nevertheless, potential energy and related quantities (e.g. pressure) can be too expensive to be recalculated. In such cases, values obtained during the CORRECTION 1 part can be used. Their error is usually small enough and should converge to zero with the order of predictor.

We are convinced that virtually any set of equations of motion can be integrated by this universal scheme and a large number of existing integration algorithms can be formally fitted into it. In the following section, application examples are given together with the preliminary results comparing the new implementation with the traditional approach.

6 Applications and preliminary results

6.1 NVE ensemble

The simplest MD simulation is carried out in the microcanonical (NVE) ensemble. Here, the new method is exactly equivalent to the traditional Verlet+SHAKE, when one of the integrators `k3m2e` or `k4m2e` is used. These two versions differ only in the velocity calculation, where the former is equivalent to the standard Velocity Verlet [4] and the latter to the Beeman's version of Verlet [3].

The improved Gear methods were proven to yield better energy conservation than the original ones when no rigid bonds were present in the simulated system [11]. The proposed SHAKE method for Gear results in better energy conservation even with rigid bonds. Consequently, the main restriction of their use is overcome.

The integration algorithm in the NVE ensemble in the terms of the universal scheme listed above is the following.

1. **PREDICTION**. As no additional degrees of freedom are needed for simulations in the NVE ensemble, the prediction step is as simple as

$$\mathbf{R}_i^{\mathrm{P}}(t) = \mathbf{A}\mathbf{R}_i^{\mathrm{C}}(t-h) \tag{43}$$

2. **EVALUATION**. As mentioned previously the RHS of the differential equation of motion in the NVE ensemble follows Newton's second law, thus

$$\begin{aligned} \mathbf{f}_i(t) &= -\frac{\partial E_{\mathrm{pot}}(\mathbf{R}^{\mathrm{P}}(t))}{\partial \mathbf{r}_i} \\ \mathbf{g}_i(t) &= \frac{h^2 \mathbf{f}_i(t)}{2m_i} - \mathbf{R}_i^{\mathrm{P}}(t)^{\mathrm{T}} \delta_2 \end{aligned} \tag{44}$$

   where the forces are calculated from the positions stored in $\mathbf{R}_1^{\mathrm{P}}, \mathbf{R}_2^{\mathrm{P}}, \ldots, \mathbf{R}_N^{\mathrm{P}}$.

3. **CORRECTION 2**. Needed only if SHAKE needs to be performed.

$$\mathbf{R}_i^{\mathrm{PI}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) \tag{45}$$

4. **SHAKE**. This step is needed only for the systems with rigid bonds. The algorithm was described above, the key equations are written again here.

$$\begin{aligned} \lambda &= \frac{\left|\mathbf{r}_{ij}^{\mathrm{PI}}(t+h)\right|^2 - l^2}{2l^2} \\ \mathbf{h}_i(t) &= \frac{1}{\sum_i c_i} \lambda \frac{m_j}{m_i + m_j} \mathbf{r}_{ij}^{\mathrm{P}}(t) \end{aligned} \tag{46}$$

5. **CORRECTION 2**. Final correction is always required to take the differential equation into account.

$$\mathbf{R}_i^{\mathrm{C}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) + \mathbf{c}\mathbf{h}_i^{\mathrm{T}}(t) \tag{47}$$

6. **EVALUATION & CORRECTION** for the **EXTENDED DOFs**. This step can be *skipped* because there are no extended degrees of freedom in the NVE ensemble.

7. **MEASUREMENT**. Performed traditionally.

To test the new SHAKE method, two simple systems were chosen. The results are summarized below.

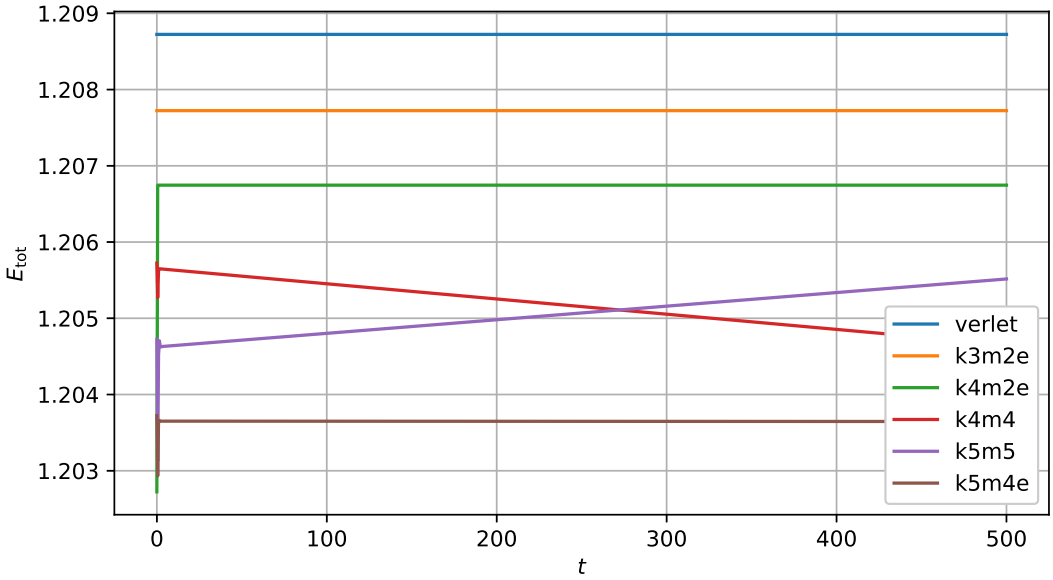


Figure 1: Total energy (in $k_{\mathrm{B}}\,\mathrm{K}$) of a dump-bell integrated by different methods (time step $h = 0.1\,\mathrm{ps}$). To see all methods results, multiples of $0.001\,k_{\mathrm{B}}\,\mathrm{K}$ are added.

6.1.1 Dump-bell

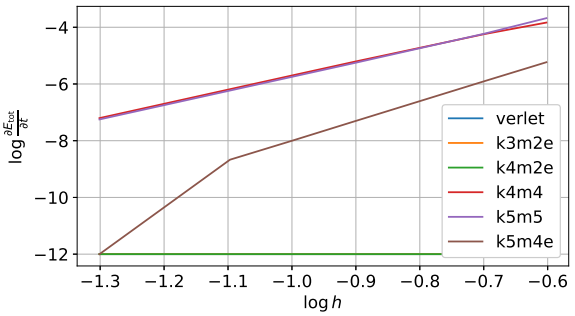
The elemental example of a system with a rigid bond could be called a *dump-bell*, i.e. a system consisting of two point particles connected by a rigid bond in free (vacuum) boundary conditions. In this trivial example, no real forces are acting on the system. Two point particles of mass $1\,\mathrm{g\,mol^{-1}}$ connected by a rigid bond of length $2\,\text{Å}$ originally laying on the x-axis are given velocities $1\,\text{Å}\,\mathrm{ps^{-1}}$ in opposite directions along the y-axis. Thus, the total energy should be $10\,\mathrm{J\,mol^{-1}} = 1.2027\,k_{\mathrm{B}}\,\mathrm{K}$. The results of numerical integration are shown on Figure 1. The main difference is the oscillations at the beginning caused by a gradual stabilization of higher derivatives terms in $\mathbf{R}_i$.

All methods perform well because the SHAKE method is constructed to conserve momentum, which means in this case also the total energy. However, despite the fact that the force calculated by SHAKE conserves the total momentum, not all the methods conserve energy exactly. The energy drift obtained by higher-order Gear methods is qualitatively the same as if they were applied to the system without constraints. The order of energy conservation can be roughly estimated from Figure 2a which shows the dependence of the common logarithm of the energy drift on the common logarithm of the time step for various methods. Theoretical values of time-reversibility order are 7 for `k5m4e` method and 5 for `k4m4` and `k5m5` methods.

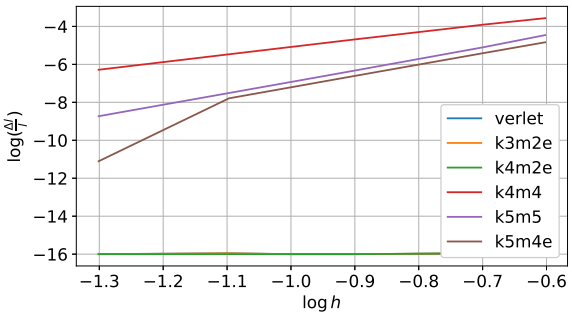
In section 4.4 SHAKE with higher order Gear integrators we have argued that the error in the length of the rigid bond is sufficiently small for the higher order methods, though it is always present. Figure 2b shows the highest relative error as a function of the time step (again expressed by common logarithms). It can be noticed that for the Verlet method and its equivalent the error is at most 10^{-16} which can be considered as zero but for the higher order methods the relative errors are of order 10^{-11}–10^{-4}, strongly dependent on the time step. The improved method again performs better than the original.

6.1.2 Nitrogen

The second more realistic system consists of 64 molecules of nitrogen. Two nitrogen atoms are connected by a rigid bond of length $1.09\,\text{Å}$ and interact with the other atoms by a standard 12-6 Lennard-Jones potential with $\epsilon = 99.8\,k_{\mathrm{B}}\mathrm{K}$ and $\sigma = 3.667\,\text{Å}$ [23]. Simulation outcomes are



(a) Energy slope (common logarithm) as a function of time step (common logarithm).



(b) Maximum relative error of the length of the rigid bond as a function of time step (both as a common logarithm).

Figure 2: Dump-bell integrated by various integrators and time steps.

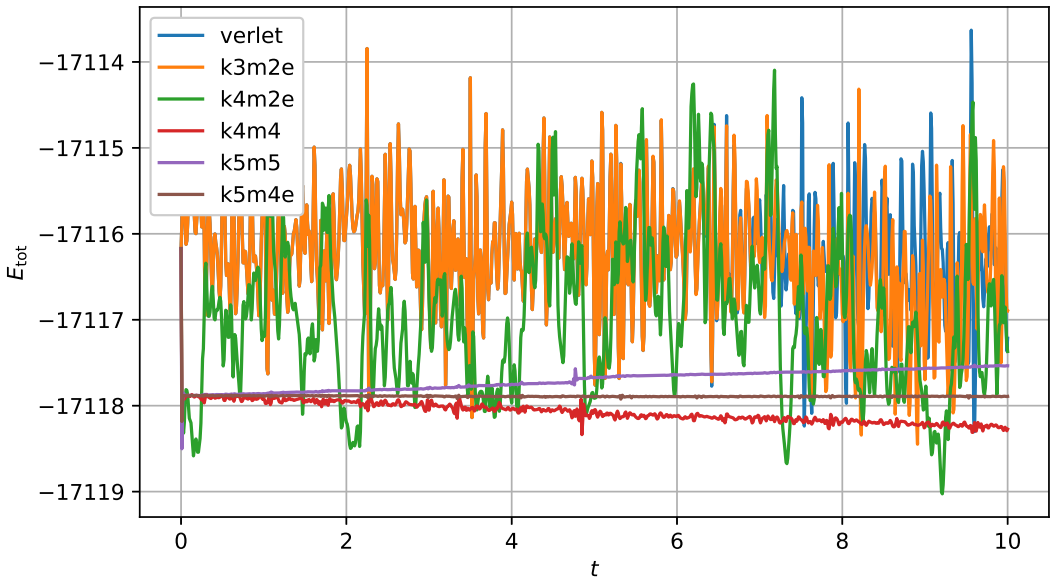


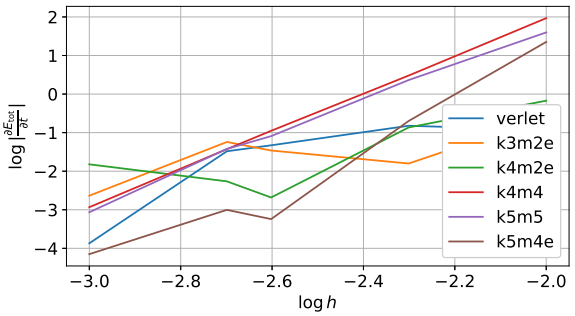
Figure 3: Total energy (in k_B K) of the nitrogen system integrated by different methods with the time step $h = 0.002$ ps.

consistent with the results for a dump-bell. In Figure 3, the total energy is plotted as a function of time. The two velocity Verlet integrators give the same numbers until round-off errors cause the separation of trajectories. The last Verlet equivalent method (`k4m2e`) does not trace exactly the same trajectory due to the slightly different initial conditions (as the third derivative of $\mathbf{r}_i$ is initially set to 0). The higher-order integrators obey expected trends. It can be immediately noticed, that the fluctuations are strikingly smaller than in the case of the Verlet method. The energy drift is evident for the original methods (`k4m4` and `k5m5`), but it is almost negligible for the improved method (`k5m4e`).

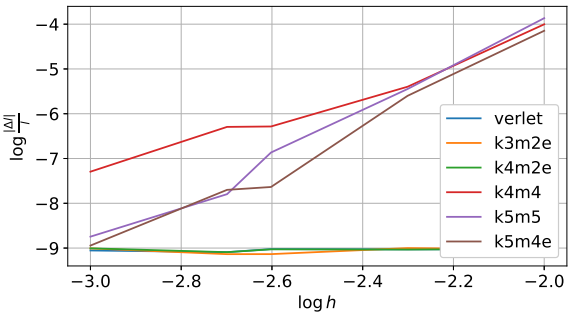
Figure 4 depicts the dependence of the energy slope and maximum relative error of rigid bond length on the time step. The overall picture mostly supports theoretical predictions. Despite the conceptual difficulty of the method time-reversibility order estimation, the presented results at least do not contradict the assumptions made in our article [11].

6.1.3 SPC/E water

The third tested system comprises 300 molecules of water modelled by the simple SPC/E water model. All atoms carry partial charges and thus interact via cutoff electrostatic forces (MACSIMUS-style cutoff). Lennard-Jones interactions are restricted to oxygen atoms. All



(a) Energy slope (common logarithm) as a function of time step (common logarithm).



(b) Maximum relative error of the length of the rigid bonds as a function of time step (both as a common logarithm).

Figure 4: Nitrogen system integrated by various integrators and 5 different time steps.

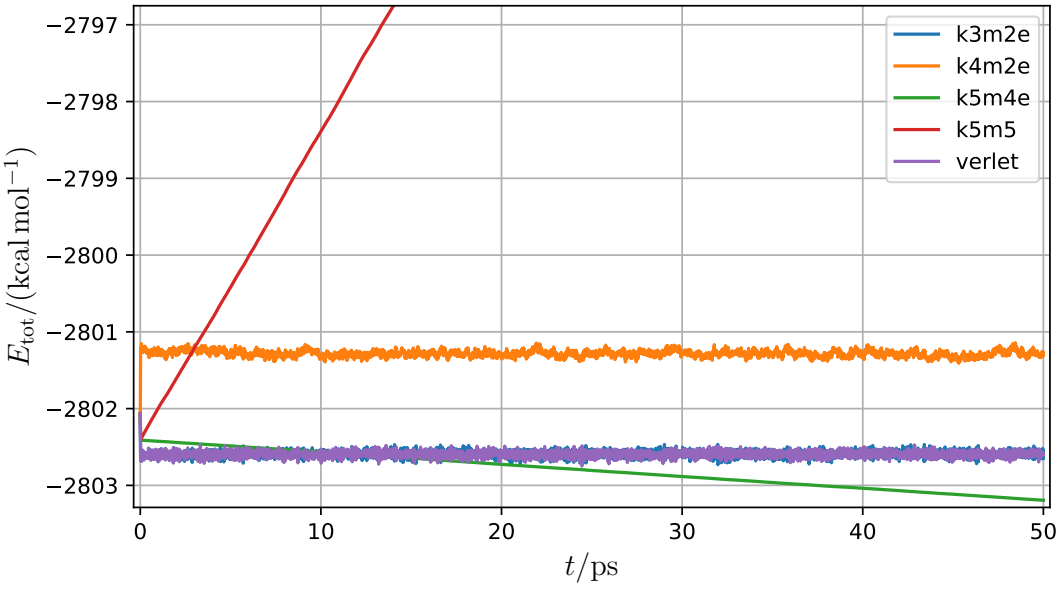


Figure 5: Total energy (in kcal/mol) of the SPC/E water system integrated by different methods with time step $h = 0.001\,\mathrm{ps}$.

bonds are constrained. A series of simulations were performed with two different timesteps. The total energy is plotted in Figures 5 and 6 as a function of time. All verlet-type methods (`verlet`, `k3m2e` and `k4m2e`) provide very good energy conservation. The pronounced rising trend is visible for the original `k5m5`. With the new `k5m4e` method, the trend is less extreme but still evident. However, the orders of energy conservation seem to obey the predicted values (5 for `k5m5` and 7 for `k5m4e`) as can be read from Table 2.

6.2 NVT ensemble – Nosé–Hoover thermostat

Trajectory in the canonical ensemble can be produced by adding one extra degree of freedom to the studied system. This is the key idea of the Nosé–Hoover thermostat. The equations of motion are

$$\begin{aligned} \ddot{\mathbf{r}}_i &= \frac{\mathbf{f}_i}{m_i} - \dot{\mathbf{r}}_i\dot{\xi} \\ \ddot{\xi} &= \frac{1}{\tau_T^2}\left(\frac{T_{\mathrm{kin}}}{T_{\mathrm{ext}}} - 1\right) \end{aligned} \tag{48}$$

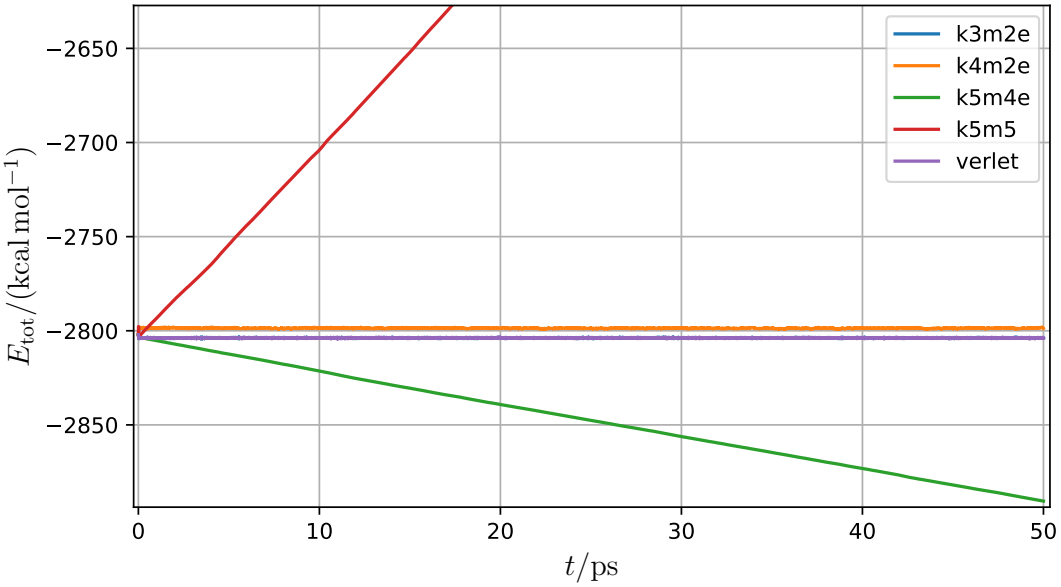


Figure 6: Total energy (in kcal/mol) of the SPC/E water system integrated by different methods with time step $h = 0.002\,\mathrm{ps}$.

Table 2: Energy conservation during NVE simulations of SPC/E water. Non-zero values for Verlet-type integrators are given mainly by the initial step in energy (see Figures 5 and 6). The energy conservation orders were estimated as $\log_2 \left[(\frac{\partial E}{\partial t})_{h=0.002\,\mathrm{ps}} / (\frac{\partial E}{\partial t})_{h=0.001\,\mathrm{ps}} \right]$.

Integrator	Energy slope ($\mathrm{kcal\,mol^{-1}}$)		Estimated order
	$h = 0.001\,\mathrm{ps}$	$h = 0.002\,\mathrm{ps}$	
`k3m2e`	0.000 17(4)	0.000 31(16)	0.886
`k4m2e`	−0.000 40(4)	−0.0019(2)	2.212
`k5m4e`	−0.015 762(6)	−1.7312(4)	6.779
`k5m5`	0.402 94(3)	11.006(6)	4.772
`verlet`	0.000 09(4)	0.000 24(2)	1.401

where T_{ext} is the target thermostat temperature and τ_T is the thermostat correlation time. The important change is the dependence of acceleration on velocity. In section 3.1 TRVP – Time-Reversible Velocity Predictor, we have drafted possible solutions to the problem and chosen the TRVP method as a convenient method which can be formally fitted to the universal scheme which has now the form

1. **PREDICTION.** The prediction of both the physical $(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$ and the extended (ξ) degrees of freedom. This step includes TRVP for all degrees of freedom needed for Verlet equivalent methods. In principle, it is possible to use TRVP also for the higher order integrators.

$$\begin{aligned} \mathbf{R}_i^{\mathrm{P}}(t) &= \mathbf{A}\mathbf{R}_i^{\mathrm{C}}(t-h) \\ \boldsymbol{\xi}^{\mathrm{P}}(t) &= \mathbf{A}\boldsymbol{\xi}^{\mathrm{C}}(t-h) \end{aligned} \tag{49}$$

   where

$$\boldsymbol{\xi}(t) = \begin{bmatrix} \xi(t) \\ h\dot{\xi}(t) \\ \frac{h^2}{2}\ddot{\xi}(t) \\ \vdots \\ \frac{h^{k-1}}{(k-1)!}\xi^{(k-1)}(t) \end{bmatrix} \tag{50}$$

2. **EVALUATION.** The accelerations are computed from the predicted values of velocities.

$$\mathbf{g}_i(t) = \frac{h^2}{2}\left(\frac{\mathbf{f}_i(t)}{m_i} - \dot{\xi}^{\mathrm{P(TRVP)}}(t)\dot{\mathbf{r}}_i^{\mathrm{P(TRVP)}}(t)\right) - \mathbf{R}_i^{\mathrm{P}}(t)^{\mathrm{T}}\delta_2 \tag{51}$$

3. **CORRECTION 1** (if SHAKE needed).

$$\mathbf{R}_i^{\mathrm{I}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) \tag{52}$$

4. **SHAKE** (if needed).

$$\begin{aligned} \lambda &= \frac{\left|\mathbf{r}_{ij}^{\mathrm{PI}}(t+h)\right|^2 - l^2}{2l^2} \\ \mathbf{h}_i(t) &= \frac{1}{\sum_i c_i}\lambda\frac{m_j}{m_i+m_j}\mathbf{r}_{ij}^{\mathrm{P}}(t) \end{aligned} \tag{53}$$

5. **CORRECTION 2.** Final correction of physical degrees of freedom.

$$\mathbf{R}_i^{\mathrm{C}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) + \mathbf{c}\mathbf{h}_i^{\mathrm{T}}(t) \tag{54}$$

6. **EVALUATION & CORRECTION** for $\boldsymbol{\xi}$. Note that $g_\xi(t)$ is here a scalar value.

$$\begin{aligned} g_\xi(t) &= \frac{h^2}{2\tau_T^2}\left(\frac{T_{\text{kin}}}{T_{\text{ext}}} - 1\right) - \boldsymbol{\xi}_i^{\mathrm{P}}(t)^{\mathrm{T}}\delta_2 \\ \boldsymbol{\xi}_i^{\mathrm{C}}(t) &= \boldsymbol{\xi}_i^{\mathrm{P}}(t) + g_\xi(t)\mathbf{c} \end{aligned} \tag{55}$$

7. **MEASUREMENT**.

The introduced algorithm with `k3m2e` integrator is exactly equivalent to the MACSIMUS implementation of Nosé–Hoover thermostat [2]. Nevertheless, it should be compared with the common implementations that use iterations or rather Trotter decomposition of the Liouville operator. One criterion of integration quality is the conserved quantity which is for Nosé–Hoover thermostat

$$E_{\text{conserved}} = E_{\text{pot}} + E_{\text{kin}} + \frac{1}{2}N_{\mathrm{f}}k_{\mathrm{B}}T_{\text{ext}}\tau_T^2\dot{\xi}^2 + N_{\mathrm{f}}k_{\mathrm{B}}T_{\text{ext}}\xi \tag{56}$$

Thus, the conserved quantity includes the kinetic and potential energy of the thermostat (of ξ).

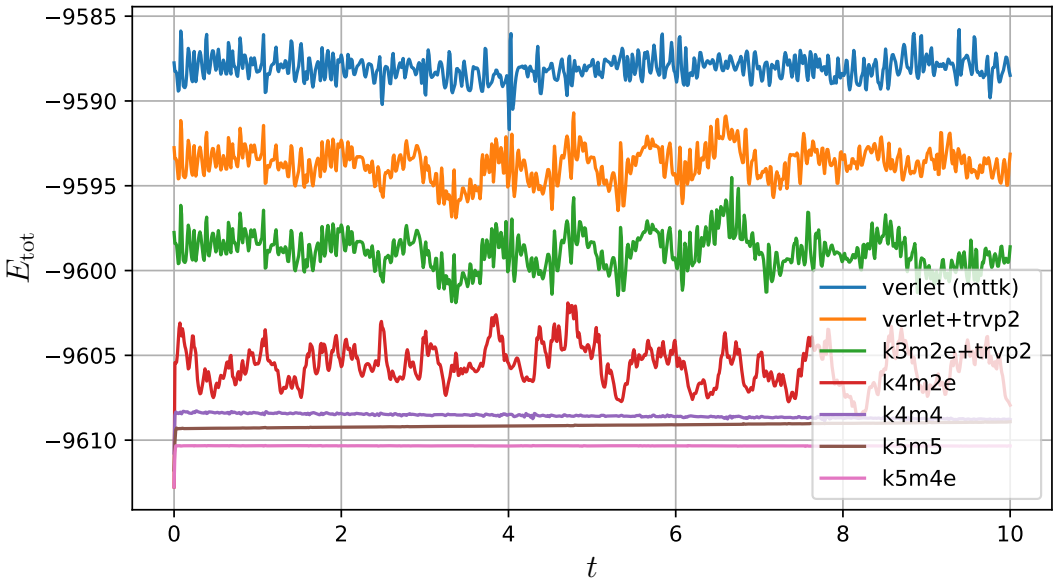


Figure 7: Total energy (in $k_{\mathrm{B}}\,\mathrm{K}$) of the nitrogen system integrated by different methods in NVT Nosé–Hoover ensemble with time step $h = 0.002\,\mathrm{ps}$. Multiples of $k_{\mathrm{B}}\,\mathrm{K}$ are added to enhance visibility.

6.2.1 Nitrogen

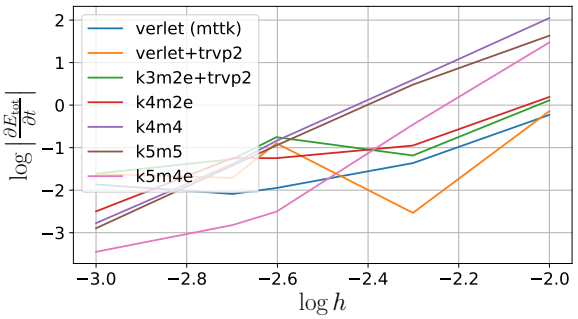
To test the properties of the universal scheme in the NVT Nosé–Hoover ensemble, the same system of rigid nitrogen molecules was used as for the NVE ensemble (see 6.1.2 Nitrogen). Thermostat correlation time $\tau_T = 0.2\,\mathrm{ps}$ was used. Three different implementations were tested. The new method was compared with the traditional implementation using Trotter expansion (MTTK method) [10] and with the MACSIMUS implementation of TRVP method [2]. However, the results obtained by MTTK implementation are not in accordance with results obtained using DL_POLY_4 software, presumably due to an implementation bug.

The conserved quantity time dependence for time step 0.002 ps is shown in Figure 7. The MTTK method gives too large fluctuations. The original TRVP implementation (marked by `verlet+trvp2`) in MACSIMUS leads to similar results as the new implementation (denoted `k3m2e+trvp2`). The other 4 methods shown are the implementations of the new scheme without TRVP. Note the `k4m2e` method, which is a Verlet equivalent method but with a different estimate of velocity. The energy conservation is satisfactory even without the need for TRVP. The higher order methods work as expected, the improved `k5m4e` being the best of them in energy conservation.

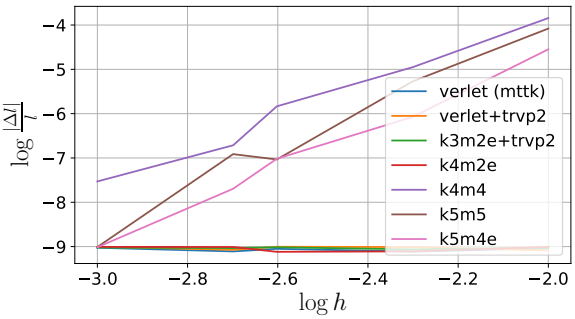
Figure 8 shows the dependence of the conserved energy slope and bond length corruption on the time step. Similar results to the previously discussed are obtained.

6.3 NPT ensemble – Nosé–Hoover (MTK) barostat

The isobaric (NPT) ensemble can be reproduced using equations of motion pioneered by Andersen [24], extended by Nosé [7] and Hoover [21] and eventually formulated by Martyna et al. [9]. Besides the degree of freedom connected with the thermostat (ξ), another degree of freedom representing the size of the simulation box is added (Λ, logarithm of box size L). The



(a) Conserved energy slope (common logarithm) as a function of time step (common logarithm).



(b) Maximum relative error of the length of the rigid bonds as a function of time step (both as a common logarithm).

Figure 8: Nitrogen system integrated by various integrators in NVT Nosé–Hoover ensemble and 5 different time steps.

equations of motion are

$$\begin{aligned}
\Lambda &= \ln L \\
M_T &= N_\mathrm{f} k_\mathrm{B} T_\mathrm{ext} \tau_T^2 \\
M_P &= (N_\mathrm{f}+3) k_\mathrm{B} T_\mathrm{ext} \tau_P^2 \\
\dot{\mathbf{r}}_i &= \mathbf{v}_i + \dot{\Lambda}\mathbf{r}_i \\
\dot{\mathbf{v}}_i &= \frac{\mathbf{f}_i}{m_i} - \left(\dot{\xi} + \frac{3\dot{\Lambda}}{N_\mathrm{f}} + \dot{\Lambda}\right)\mathbf{v}_i \\
\ddot{\xi} &= \frac{1}{M_T}\left(2E_\mathrm{kin} + M_P\dot{\Lambda}^2 - (N_\mathrm{f}+1)k_\mathrm{B}T_\mathrm{ext}\right) \\
\ddot{\Lambda} &= \frac{3}{M_P}\left(V(P_\mathrm{cfg} - P_\mathrm{ext}) + \frac{2E_\mathrm{kin}}{N_\mathrm{f}}\right) - \dot{\xi}\dot{\Lambda}
\end{aligned} \tag{57}$$

where T_ext is the target thermostat temperature and τ_T is the thermostat correlation time, P_ext is the target pressure and τ_P the barostat correlation time. The kinetic energy is computed from velocity $\mathbf{v}_i$, the derivative of position $\dot{\mathbf{r}}_i$ is not used. The full universal integration scheme must be used without skipping any step when constrained bonds are present.

1. **PREDICTION.** The prediction of both the physical and the extended degrees of freedom. This step includes (optional) TRVP for all degrees of freedom.

$$\begin{aligned}
\mathbf{R}_i^\mathrm{P}(t) &= \mathbf{A}\mathbf{R}_i^\mathrm{C}(t-h) \\
\boldsymbol{\xi}^\mathrm{P}(t) &= \mathbf{A}\boldsymbol{\xi}^\mathrm{C}(t-h) \\
\boldsymbol{\Lambda}^\mathrm{P}(t) &= \mathbf{A}\boldsymbol{\Lambda}^\mathrm{C}(t-h)
\end{aligned} \tag{58}$$

   The differential equation for $\dot{\mathbf{r}}_i$ is substituted by $\dot{\mathbf{r}}_i = \mathbf{v}_i$ and the second part ($\dot{\Lambda}\mathbf{r}_i$) is achieved by direct rescaling of the simulation box by

$$\exp\left(\Lambda^\mathrm{P}(t) - \Lambda^\mathrm{C}(t-h)\right) \tag{59}$$

   This scaling is applied to all atomic positions and on the size of the simulated periodic box. The velocities stored in $\mathbf{R}_i$ thus remain intact.

2. **EVALUATION.** The accelerations are computed from predicted positions and velocities (using TRVP values if TRVP is used).

$$\mathbf{g}_i(t) = \frac{h^2}{2}\left(\frac{\mathbf{f}_i(t)}{m_i} - \left(\dot{\xi}^{\mathrm{P(TRVP)}}(t) + \left(\frac{3}{N_\mathrm{f}}+1\right)\dot{\Lambda}^{\mathrm{P(TRVP)}}(t)\right)\dot{\mathbf{r}}_i^{\mathrm{P(TRVP)}}(t)\right) - \mathbf{R}_i^\mathrm{P}(t)^\mathrm{T}\delta_2 \tag{60}$$

3. **CORRECTION 1** (optional, if SHAKE needed).

$$\mathbf{R}_i^{\mathrm{I}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) \tag{61}$$

4. **SHAKE** (optional).

$$\begin{aligned}
\left(\mathbf{R}_i^{\mathrm{PI}}(t+h) &= \mathbf{A}\mathbf{R}_i^{\mathrm{I}}(t)\right) \\
\mathbf{\Lambda}^{\mathrm{PI}}(t+h) &= \mathbf{A}\mathbf{\Lambda}^{\mathrm{P}}(t) \\
\sigma &= \exp(\Lambda^{\mathrm{PI}}(t+h) - \Lambda^{\mathrm{P}}(t)) \\
\lambda &= \frac{\left|\sigma\mathbf{r}_{ij}^{\mathrm{PI}}(t+h)\right|^2 - l^2}{2l^2} \\
\mathbf{h}_i(t) &= \frac{1}{\sum_i c_i}\lambda\frac{m_j}{m_i+m_j}\mathbf{r}_{ij}^{\mathrm{P}}(t)
\end{aligned} \tag{62}$$

5. **CORRECTION 2.**

$$\mathbf{R}_i^{\mathrm{C}}(t) = \mathbf{R}_i^{\mathrm{P}}(t) + \mathbf{c}\mathbf{g}_i^{\mathrm{T}}(t) + \mathbf{c}\mathbf{h}_i^{\mathrm{T}}(t) \tag{63}$$

6. **EVALUATION & CORRECTION** for the **EXTENDED DOFs**. Here, $g_\xi(t)$ and $g_\Lambda(t)$ are scalars.

$$\begin{aligned}
g_\xi(t) &= \frac{h^2}{2M_T}\left(2E_{\mathrm{kin}} + M_P\dot{\Lambda}^2 - (N_{\mathrm{f}}+1)k_{\mathrm{B}}T_{\mathrm{ext}}\right) - \boldsymbol{\xi}_i^{\mathrm{P}}(t)^{\mathrm{T}}\delta_2 \\
\boldsymbol{\xi}_i^{\mathrm{C}}(t) &= \boldsymbol{\xi}_i^{\mathrm{P}}(t) + g_\xi(t)\mathbf{c} \\
g_\Lambda(t) &= \frac{h^2}{2}\left(\frac{1}{M_P}\left(V(t)^{\mathrm{P}}(P_{\mathrm{cfg}}(t) - P_{\mathrm{ext}}) + \frac{2E_{\mathrm{kin}}(t)}{N_{\mathrm{f}}}\right) - \dot{\xi}^{\mathrm{C}}(t)\dot{\Lambda}^{\mathrm{P(TRVP)}}(t)\right) - \mathbf{\Lambda}_i^{\mathrm{P}}(t)^{\mathrm{T}}\delta_2 \\
\mathbf{\Lambda}_i^{\mathrm{C}}(t) &= \mathbf{\Lambda}_i^{\mathrm{P}}(t) + g_\Lambda(t)\mathbf{c}
\end{aligned} \tag{64}$$

Finally, the simulation box and atomic positions should be rescaled by

$$\exp\left(\Lambda^{\mathrm{C}}(t) - \Lambda^{\mathrm{P}}(t)\right) \tag{65}$$

7. **MEASUREMENT**.

Similarly to the NVT Nosé–Hoover ensemble, there is a conserved quantity that can be used to monitor the quality of the integration. It is the sum of the total energy of the system and kinetic and potential energies of the extended degrees of freedom.

$$E_{\mathrm{conserved}} = E_{\mathrm{kin}} + E_{\mathrm{pot}} + \frac{1}{2}M_T\dot{\xi}^2 + (N_{\mathrm{f}}+1)k_{\mathrm{B}}T_{\mathrm{ext}}\xi + \frac{1}{2}M_P\dot{\Lambda}^2 + P_{\mathrm{ext}}V \tag{66}$$

The new method is not exactly equivalent to the MACSIMUS NPT implementation despite the idea being similar. Therefore, it should be carefully tested.

6.3.1 Nitrogen

Using barostat, the system volume fluctuates and a bigger system is needed so that the cutoff distance can be sufficiently large and still safely shorter than a half of the current box size. The system of 512 nitrogen molecules was therefore used for tests. Force field parameters are the same as above (see section 6.1.2). Target temperature was 300 K, target pressure 101 325 Pa, correlation times $\tau_T = 0.2$ ps and $\tau_P = 2$ ps.

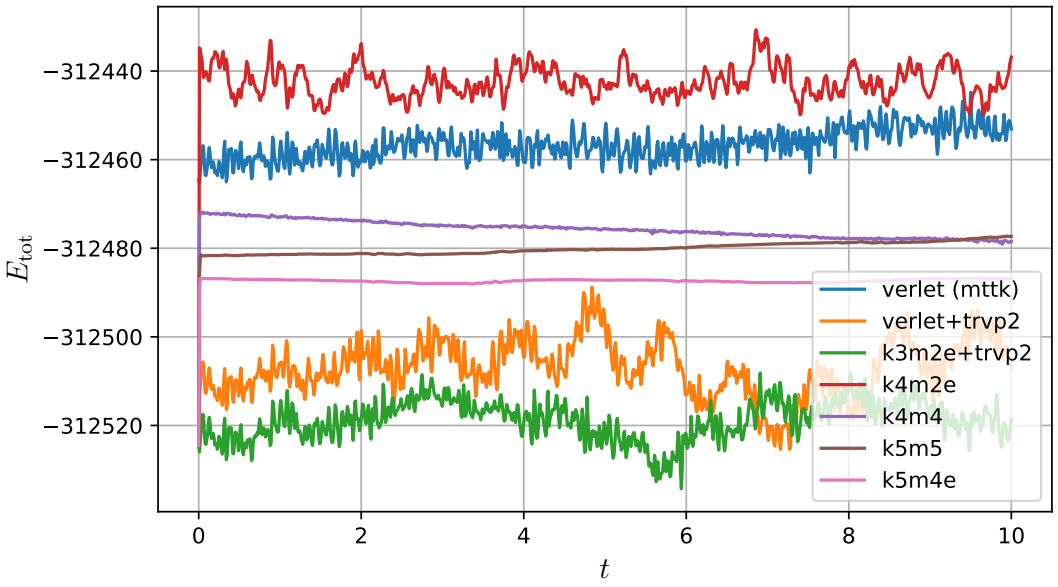
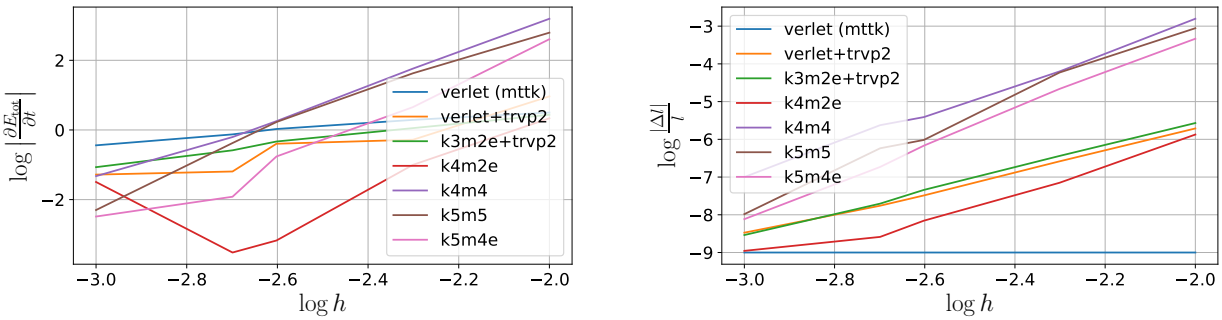


Figure 9: Total energy (in $k_{\mathrm{B}}\,\mathrm{K}$) of the nitrogen system integrated by different methods in NPT Nosé–Hoover ensemble with time step $h = 0.002\,\mathrm{ps}$. Multiples of $5\,k_{\mathrm{B}}\,\mathrm{K}$ are added to enhance visibility.



(a) Conserved energy slope (common logarithm) as a function of time step (common logarithm).

(b) Maximum relative error of the length of the rigid bonds as a function of time step (both as a common logarithm).

Figure 10: Nitrogen system integrated by various integrators in NPT Nosé–Hoover ensemble and 5 different time steps.

The new method was again compared to the MACSIMUS style implementation (denoted by `verlet+trvp2`) and DL_POLY_4 style implementation using Trotter decomposition (`verlet`). Unfortunately, total energy conservation is reproduced neither by DL_POLY_4 software (linearly decreasing total energy). Nevertheless, our implementation seems to work well and was thus used for comparison (`verlet (mttk)`). The obtained results are shown in Figure 9 and Figure 10. The overall behaviour of new methods confirms our expectations. The new method performs reasonably well, the `k5m4e` method being the best.

The constrained bonds error is higher than in the case of the NVT ensemble. One source of inaccuracies – correction (constraints are exactly fulfilled for the predicted values) – is accompanied by the imprecise guess of the future box size. Nonetheless, the error is still smaller than 0.1 % for all tested time steps.

7 Conclusion

In this work, we have outlined the derivation of the universal integration scheme for molecular dynamics starting from the traditional Verlet + SHAKE algorithm. To solve equations with

velocity-dependent right-hand sides, the TRVP algorithm was adopted. To maintain the rigid bonds, a new method based on SHAKE was developed. Not to restrict ourselves to using only the Verlet algorithm, the Gear-type predictor-corrector integrators are incorporated. Altogether, the proposed scheme offers a universal recipe, how the equations of motion can be solved.

Despite being theoretically equivalent (or almost equivalent in the case of NPT) to the methods currently used, the new method should be extensively tested. A few preliminary tests have already shown that results similar to those obtainable by classical methods can be produced. However, a much deeper study is needed. This is particularly true for the simulations in the NPT ensemble, as the proposed method is not even theoretically exactly equivalent to the methods currently used.

References

[1] Verlet, L. Computer 'Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, *159*, 98–103.

[2] Kolafa, J. *MACSIMUS manual*; http://old.vscht.cz/fch/software/macsimus/macsimus.pdf (accessed May 4, 2019).

[3] Beeman, D. Some Multistep Methods for Use in Molecular Dynamics Calculations. *J. Comput. Phys* **1976**, *20*, 130–139.

[4] Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **1982**, *76*, 637–649.

[5] Yoshida, H. Construction of higher order symplectic integrators. *Phys. Lett. A* **1990**, *150*, 262–268.

[6] Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: New York, NY, 1989.

[7] Nosé, S. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.* **1984**, *52*, 255–268.

[8] Hoover, W. G. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A* **1985**, *31*, 1695–1697.

[9] Martyna, G. J.; Tobias, D. J.; Klein, M. L. Constant pressure molecular dynamics algorithms. *J. Chem. Phys.* **1994**, *101*, 4177–4189.

[10] Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. Explicit reversible integrators for extended systems dynamics. *Mol. Phys.* **1996**, *87*, 1117–1157.

[11] Janek, J.; Kolafa, J. Novel Gear-like predictor–corrector integration methods for molecular dynamics. *Mol. Phys.* **2020**, *118*, e1674937.

[12] Janek, J. Improving time-reversibility of predictor-corrector integrators. Master thesis, University of Chemistry and Technology, Prague, 2019.

[13] Gear, C. W. Maintaining Solution Invariants in the Numerical Solution of ODEs. *SIAM Journal on Scientific and Statistical Computing* **1986**, *7*, 734–743.

[14] Gear, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*; Prentice-Hall: Engelwood Cliffs, NJ, 1971.

[15] Kolafa, J.; Lísal, M. Time-Reversible Velocity Predictors for Verlet Integration with Velocity-Dependent Right-Hand Side. *J. Chem. Theory Comput.* **2011**, *7*, 3596–3607.

[16] Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. 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.

[17] Andersen, H. C. Rattle: A "velocity" version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.* **1983**, *52*, 24–34.

[18] Kolafa, J.; Nezbeda, I.; Kotrla, M. *Úvod do počítačových simulací. Metody Monte Carlo a molekulární dynamiky*; Karolinum: Praha, 2003.

[19] de Leeuw, S. W.; Perram, J. W.; Petersen, H. G. Hamilton's equations for constrained dynamical systems. *J. Stat. Phys.* **1990**, *61*, 1203–1222.

[20] Frenkel, D.; Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed.; Academic Press: San Diego, CA, 2002.

[21] Hoover, W. G. Constant-pressure equations of motion. *Phys. Rev. A* **1986**, *34*, 2499–2500.

[22] Klíma, M.; Kolafa, J. Direct Molecular Dynamics Simulation of Nucleation during Supersonic Expansion of Gas to a Vacuum. *J. Chem. Theory Comput.* **2018**, *14*, 2332–2340.

[23] Chen, R. Y.; Yuen, W. Y. D. Short-Time Oxidation Behavior of Low-Carbon, Low-Silicon Steel in Air at 850–1180 °C: II. Linear to Parabolic Transition Determined Using Existing Gas-Phase Transport and Solid-Phase Diffusion Theories. *Oxid. Met.* **2009**, *73*, 353–373.

[24] Andersen, H. C. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* **1980**, *72*, 2384–2393.