Accelerated Green's function molecular dynamics

V. R. Coluci

School of Technology, University of Campinas − UNICAMP, Limeira–SP, 13484-332 Brazil

S. O. Dantas

Departamento de Física, ICE, Universidade Federal de Juiz de Fora, 36036-330 Juiz de Fora MG, Brazil

V. K. Tewary

Applied Chemical and Materials Division, National Institute of Standards and Technology, Boulder, Colorado 80305, USA

Abstract

A Green's function formalism has been applied to solve the equations of motion in classical molecular dynamics simulations. This formalism enables larger time scales to be probed for vibration processes in carbon nanomaterials. In Green's function molecular dynamics (GFMD), the total interaction potential is expanded up to the quadratic terms which, which enables an exact solution of the equations of motion to be otained for problems within the harmonic approximation, reasonable energy conservation, and fast temporal convergence. Differently from conventional integration algorithms in molecular dynamics, GFMD performs matrix multiplications and diagonalizations within its main loop, which make its computational cost high and, therefore, has limited its use. In this work, we propose a method to accelerate GFMD simulations by treating the full system of N atoms as a collection of N smaller systems of size n. Diagonalization is performed for smaller $nd \times nd$ dynamical matrices rather than the full $Nd \times Nd$ matrix ($d = 1, 2$, or 3). The eigenvalues and eigenvectors are then used in the GFMD equations to update the atomic positions and velocities. We applied the method for one-dimensional lattices of oscillators and have found that the method rapidly converges to the exact solution as n increases. The computational time of the proposed method scales linearly with N, providing a considerable gain with respect to the $\mathcal{O}(N^3)$ full diagonalization. The method also exhibits better accuracy and energy conservation than the velocity-Verlet algorithm. An OpenMP parallel verison has been implemented and tests indicate a speed up of $14\times$ for $N = 50000$ in affordable computers. Our findings indicate that GFMD can be an alternative, competitive integration technique for molecular dynamics simulations.

Keywords: Molecular dynamics, Green's functions, Parallel processing

1. Introduction

Molecular dynamics (MD) simulations are important tools to study atomic and molecular systems in biology, chemistry, physics, and material science. These simulations rely on the integration of the equations of motion to determine the evolution of atomic positions, velocities, and other quantities. One important component of the integration techniques is the integration time step size. Because atoms in molecules and crystals vibrate very rapidly with periods of the order of 1 fs, the time step size should also be of this same order to properly resolve individual atomic motions. Such small time step sizes limit the time span MD simulations can achieve because of the large number of time steps necessary to reach the time scales of realistic processes. For example, protein folding occurs in the time scale of μs to s, which requires 10^9 to 10^{15} integration steps. On the other hand, error accumulation due to numerical truncation becomes an issue for such long simulations. Therefore, developing techniques that allow the use of larger time step sizes in MD integrators is a route to increase the applicability of MD simulations.

MD occurs through a series of events that change the system from one state to another in the energy landscape. Techniques that simulate such events have been used to accelerate the dynamics [1]. These techniques are ways to provide larger time steps or even time "jumps" to speed up the simulation. Techniques based on the transition state theory have been used to induce changes in the atomic states more rapidly in the energy landscape in order to overcome energy barriers during the simulation [2]. Multiple simulations running in parallel with different atom momenta is the basis of the parallel replica technique [3, 4] where one initial simulation "explores" the system and, if it lasts long enough, other simulations are started to reach another energy minimum states. In order to avoid these states, the hyperdynamics technique modifies the real potential to a biased potential to increase the probability of changing between energy states [5]. Another way to increase these probabilities is by increasing the temperature of the simulated system in temperature-accelerated dynamics [6].

Differently from those techniques that rely on changes in the actual system to accelerate the dynamics, Tewary proposed a different approach based on the Green's function formalism [7]. The Green's functions molecular dynamics (GFMD) integrates the equations of motion with a less severe constraint in the time step size than in conventional MD integrators to obtain the exact solutions of the equations of motion for up quadratic terms in the atomic displacements in the total interatomic potential. Applied to the study of vibrations of one-dimensional lattices of nonlinear oscillators and graphene, GFMD was able to extend the time scale of these simulations by eight orders of magnitude, allowing to study these systems in the μs scale [7]. Initially developed for MD simulations on the micro-canonical ensemble, GFMD has been extended to the canonical ensemble to study systems in contact with thermal baths [8].

The use of GFMD, however, has been partially limited due to its high computational cost when compared to conventional MD integrators. GFMD requires matrix multiplications and diagonalizations of the dynamical matrix during the

integration loop. Tewary proposed some strategies to reduce the GFMD computational cost [7]. For example, for short range interaction potentials, the dynamical matrix is sparse and may be banded. Another suggestion was the single atom iterative calculation where GFMD is applied to a single atom, keeping all the rest fixed during the integration interval t to $t + \Delta t$. In this case, the dynamical matrix is a 3×3 matrix in 3D systems that can be easily diagonalized.

In this work, we propose a method called the n-atom approximation, to accelerate GFMD that generalizes the single atom iterative calculation. In section 2, we present the GFMD technique and its main equations. The details of the n-atom approximation is described in Section 3. A parallel version of the method is described in Section 4 and the analysis of the accuracy and the computational cost of the n-atom approximation is shown in Section 5. Finally, the conclusions are presented in Section 6.

2. Green's Function Molecular Dynamics

We present in this section a detailed derivation of the GFMD main equations [7]. We consider a system composed of N interacting atoms of mass m_i ($i = 1 \ldots N$). The position of the i-atom at the initial time ($t = 0$) is specified by $\mathbf{r}_{0i}$. Expressing $\mathbf{r}_{0i}$ in terms of the atomic displacement $\mathbf{u}_i(t)$ as

$$\mathbf{r}_i(t) = \mathbf{r}_i(0) + \mathbf{u}_i(t), \tag{1}$$

the Newton's equations of motion can be written as

$$m_i \frac{\partial^2 u_{i,\alpha}}{\partial t^2} = -\frac{\partial V}{\partial u_{i,\alpha}}, \tag{2}$$

where V is the total interaction potential, and $u_{i,\alpha}$ is the α-component ($\alpha = x, y, z$) of the atomic displacement.

Expanding V in a Taylor's series in powers of $|\mathbf{u}_i(t)|$, we can rewrite Eq. (2) as

$$m_i \frac{\partial^2 u_{i,\alpha}}{\partial t^2} = \mathcal{F}_{i,\alpha} - \sum_{j=1}^{N} \sum_{\beta=x,y,z} \frac{\partial^2 V(\{|\mathbf{r}_{0i}|\})}{\partial u_{i,\alpha} \partial u_{j,\gamma}} u_{j,\beta}(t) + \Delta\mathcal{F}_{i,\alpha}(t). \tag{3}$$

where $\mathcal{F}_{i,\alpha}$ is the first Taylor's coefficient, and $\Delta\mathcal{F}_{i,\alpha}$ contains all the higher order non-harmonic terms associated with the potential. Defining $\mathcal{U}_{i,\alpha}(t) \equiv \sqrt{m_i}\, u_{i,\alpha}(t)$, Eq. (3) can be rewritten as

$$
\begin{aligned}
\frac{\partial^2 \mathcal{U}_{i,\alpha}}{\partial t^2} &= \frac{\mathcal{F}_{i,\alpha}}{\sqrt{m_i}} - \sum_{j=1}^{N} \sum_{\beta=x,y,z} \frac{1}{\sqrt{m_i}} \frac{\partial^2 V}{\partial \mathcal{U}_{i,\alpha} \partial \mathcal{U}_{j,\beta}}(\{|\mathbf{r}_{0i}|\}) \frac{1}{\sqrt{m_j}}\, \mathcal{U}_{j,\beta}(t) \\
&+ \frac{\Delta\mathcal{F}_{i,\alpha}(t)}{\sqrt{m_i}}.
\end{aligned}
\tag{4}
$$

3

The set of non-homogeneous coupled differential equations represented in Eq. (4) can be written in the following matrix form

$$\left(\mathbb{I}\frac{\partial^2}{\partial t^2} + \mathbb{D}\right)\mathbb{U} = \mathbb{F} + \Delta\mathbb{F}(t) \equiv \mathbb{F}_{\text{eff}}, \tag{5}$$

where $\mathbb{I}$ is the identity matrix (at most size $Nd \times Nd$, with $d =1,2$ or 3) and $\mathbb{U}$ is a column vector (maximum size Nd) containing all the displacements $\mathcal{U}_{i,\alpha}(t)$, $\mathbb{F}$ ($\Delta\mathbb{F}(t)$) is a squared matrix with maximum size $Nd \times Nd$ with elements $\mathcal{F}_{i,\alpha}/\sqrt{m_i}$ ($\Delta\mathcal{F}_{i,\alpha}(t)/\sqrt{m_i}$), and $\mathbb{D}$ is the dynamical matrix with maximum size of $Nd \times Nd$ and elements $\dfrac{1}{\sqrt{m_i}}\dfrac{\partial^2 V(\{|\mathbf{r}_{0i}|\})}{\partial u_{i,\alpha}\partial u_{j,\beta}}\dfrac{1}{\sqrt{m_j}}$.

The formal solution of Eq. (5) can be expressed as

$$\mathbb{U} = \left(\mathbb{I}\frac{\partial^2}{\partial t^2} + \mathbb{D}\right)^{-1}\mathbb{F}_{\text{eff}}. \tag{6}$$

If we consider $\mathbb{G}(t - t')$ as the causal Green's function, which is 0 for $t < 0$, therefore,

$$\left(\mathbb{I}\frac{\partial^2}{\partial t^2} + \mathbb{D}\right)\mathbb{G}(t - t') = \mathbb{I}\delta(t - t'), \tag{7}$$

where $\delta(t - t')$ is the Dirac's delta function.

Applying the Laplace transformation in Eq. (7), we obtain

$$\left[s^2\mathbb{I} + \mathbb{D}\right]\mathcal{L}[\mathbb{G}] - s\mathbb{G}(0) - \mathbb{G}'(0) = \mathbb{I}\mathcal{L}[\delta(t - t')], \tag{8}$$

where s is the Laplace variable conjugated to t. For the boundary conditions $\mathbb{G}(0) = \mathbb{G}'(0) = 0$, Eq. (8) becomes

$$\left[s^2\mathbb{I} + \mathbb{D}\right]\mathcal{L}[\mathbb{G}] = \mathbb{I}, \tag{9}$$

and, therefore,

$$\mathcal{L}[\mathbb{G}] = [s^2\mathbb{I} + \mathbb{D}]^{-1}. \tag{10}$$

The Laplace transform of Eq. (5) is

$$\left[s^2\mathbb{I} + \mathbb{D}\right]\mathcal{L}[\mathbb{U}] - s\mathbb{U}(0) - \mathbb{U}'(0) = \mathbb{I}\mathcal{L}[\mathbb{F}_{\text{eff}}], \tag{11}$$

or

$$\mathcal{L}[\mathbb{U}] = \mathcal{L}[\mathbb{G}]\mathcal{L}[\mathbb{F}_{\text{eff}}] + s\,\mathcal{L}[\mathbb{G}]\,\mathbb{U}(0) + \mathcal{L}[\mathbb{G}]\,\mathbb{U}'(0). \tag{12}$$

Since $\mathbb{D}$ is real and symmetric, we can write $\mathbb{V}^T\mathbb{D}\mathbb{V} = \mathbb{E}^2$, where $\mathbb{V}$ is the matrix containing the eigenvectors of $\mathbb{D}$ and $\mathbb{E}^2$ is a diagonal matrix with the eigenvalues E_i^2 of $\mathbb{D}$.

Multiplying Eq. (12) by $\mathbb{V}^T$ and using $\mathbb{I} = \mathbb{V}\mathbb{V}^T$, it can be written as

$$\mathbb{U}^{L,*} = \mathbb{K}\,\mathbb{F}_{\text{eff}}^{L,*} + s\mathbb{K}\,\mathbb{U}^{L,*}(0) + \mathbb{K}\,\mathbb{U}'^{L,*}(0), \tag{13}$$

4

where $\mathbb{K} = \mathbb{V}^T \mathcal{L}[\mathbb{G}]\mathbb{V}$, $\mathbb{U}^{L,*} = \mathbb{V}^T \mathcal{L}[\mathbb{U}]$, $\mathbb{F}_{\text{eff}}^{L,*} = \mathbb{V}^T \mathcal{L}[\mathbb{F}_{\text{eff}}]$, $\mathbb{U}^{L,*}(0) = \mathbb{V}^T \mathbb{U}(0)$, and $\mathbb{U}'^{L,*}(0) = \mathbb{V}^T \mathbb{U}'(0)$, with L standing for Laplace transform. The superscript * indicates the vectors is "projected" in the space of the eigenvectors of $\mathbb{D}$.

The inverse Laplace transform of Eq. (13) is

$$\mathcal{L}^{-1}\mathbb{U}^{L,*} = \mathcal{L}^{-1}[\mathbb{K} \, \mathbb{F}_{\text{eff}}^{L,*}] + \mathcal{L}^{-1}[s\mathbb{K} \, \mathbb{U}^{L,*}(0)] + \mathcal{L}^{-1}[\mathbb{K} \, \mathbb{U}'^{L,*}(0)], \quad (14)$$

where $\mathbb{K}$ is a diagonal matrix with elements $K_i = (s^2 + E_i^2)^{-1}$.

The term of the left hand side of Eq. (14) is evaluated as $\mathcal{L}^{-1}[W_i^{L,*}] = W_i^*(t)$ whereas the terms of the right hand side are evaluated as follows:

$$\begin{aligned}
\mathcal{L}^{-1}[K_i \mathcal{L}[F_i]] &= \mathcal{L}^{-1}\left[\frac{1}{s^2 + E_i^2} F_i^*(s)\right] \\
&= \mathcal{L}^{-1}\left[\frac{1}{s^2 + E_i^2}\right] * \mathcal{L}^{-1}[F_i^*(s)], \quad (15)
\end{aligned}$$

where $(*)$ here means the convolution of two functions and it can be written as

$$\begin{aligned}
\mathcal{L}^{-1}[K_i \mathcal{L}[F_i]] &= \frac{1}{E_i}\sin(E_i t) * \underbrace{\mathcal{L}^{-1}[F_i^*(s)]}_{F_i^*(t)} \\
&= \frac{1}{E_i}\int_0^t F_i^*(\epsilon)\sin(E_i \epsilon)d\epsilon. \quad (16)
\end{aligned}$$

Here we consider the Laplace transform

$$F_i^*(s) = \int_0^\infty F_i^*(t)e^{-st}dt, \quad (17)$$

for the case where $F_i^*(s) = \dfrac{F_i^*}{s}$ and

$$F_i^*(t) = \begin{cases} F_i^* & \text{if } t \geq 0, \\ 0 & \text{if } t < 0. \end{cases} \quad (18)$$

This case corresponds to the harmonic approximation, which means that the higher order terms in the potential expansion are neglected ($\Delta\mathbb{F}(t) = 0$).

Consequently,

$$\begin{aligned}
\mathcal{L}^{-1}[K_i \mathcal{L}[F_i]] &= \frac{1}{E_i}\int_0^t F_i^* \sin(E_i \epsilon)d\epsilon = \left.\frac{-F_i^*}{E_i^2}\cos(E_i \epsilon)\right]_0^t \\
&= \frac{-F_i^*}{E_i^2}[\cos(E_i t) - H(t)], \quad (19)
\end{aligned}$$

where $H(t)$ is the Heaviside function.

$$\mathcal{L}^{-1}[K_i U_i^{'*}(0)] = \mathcal{L}^{-1}\left[\frac{U_i^{'*}(0)}{s^2 + E_i^2}\right] = \frac{U_i^{'*}(0)}{E_i}\sin(E_i t); \qquad (20)$$

$$\mathcal{L}^{-1}[s\ K_i U_i^*(0)] = \mathcal{L}^{-1}\left[\frac{s}{s^2 + E_i^2}U_i^*(0)\right] = U_i^*(0)\cos(E_i t); \qquad (21)$$

Therefore, the components of $\mathbb{U}^*$ are

$$U_i^*(t) = \frac{-F_i^*}{E_i^2}\left[\cos(E_i t) - H(t)\right] + \frac{U_i^{'*}(0)}{E_i}\sin(E_i t) + U_i^*(0)\cos(E_i t), \qquad (22)$$

where F_i is a constant, and $C_i^*(t) \equiv U_i^{'*}(t) = \dfrac{dU_i^*(t)}{dt}$ are given by

$$C_i^*(t) = \frac{F_i^*}{E_i}\sin(E_i t) + C_i^*(0)\cos(E_i t) - U_i^*(0)E_i\sin(E_i t). \qquad (23)$$

Finally, the atomic displacements and velocities are obtained by $\mathbb{U}(t) = \mathbb{V}\mathbb{U}^*(t)$ and $\mathbb{C}(t) = \mathbb{V}\mathbb{C}^*(t)$.

The quantities $\mathbb{U}(t)$ and $\mathbb{C}(t)$ provide the exact solutions for the atomic displacements and velocities at all times t for the harmonic approximation. For non-harmonic potentials ($\Delta\mathcal{F}_{i,\alpha} \neq 0$), the exact solution is impossible to obtain. However, for these potentials, V can be locally approximated to an harmonic potential at each time step. Within this approximation, Eqs. (22) and (23) can be used to calculate the displacements and velocities for future times. Discretizing time in steps of Δt and knowing the displacement, velocity and first Taylor's coefficient in time t, we keep them fixed during the interval Δt to obtain them at $t = t + \Delta t$. The displacements and velocities at time $t + \Delta t$ are obtained using the values at time t by

$$\begin{aligned} U_i^*(t + \Delta t) &= -\frac{F_i^*(t)}{E_i^2(t)}\left[\cos[E_i(t)\Delta t] - 1\right] + \frac{C_i^*(t)}{E_i(t)}\sin[E_i(t)\Delta t] \\ &\quad + U_i^*(t)\cos[E_i(t)\Delta t], \end{aligned} \qquad (24)$$

and

$$\begin{aligned} C_i^*(t + \Delta t) &= \frac{F_i^*(t)}{E_i(t)}\sin[E_i(t)\Delta t] + C_i^*(t)\cos[E_i(t)\Delta t] \\ &\quad - U_i^*(t)E_i(t)\sin[E_i(t)\Delta t]. \end{aligned} \qquad (25)$$

The value of Δt should be small enough to keep $\Delta\mathbb{F}(t)$ negligible during Δt. In this case, $\mathbb{F}^*$ and $\mathbb{D}$ should be updated at each time step.

6

3. n-atom approximation

GFMD requires the diagonalization of the dynamical matrix $\mathbb{D}$ for the whole system (N atoms), which is a $\mathcal{O}(N^3)$ process. In order to reduce this computational cost, we propose here dividing the whole system into N smaller regions and associate a smaller dynamical matrix $\widetilde{\mathbb{D}}$ to each of them. In general, these regions can have different sizes, comprised of n_i atoms, with a dynamical matrix of size $n_i d \times n_i d$ ($d = 1, 2$, or 3). Additionally, atoms in one region can also belong to other regions. For simplicity, here we take all regions with the same number of atoms n. In this approximation – the so-called n-atom approximation – we apply the GFMD for each region $\mathcal{R}_i$ around atom i, assuming the atoms of the remaining regions to have fixed positions. The n-atom approximation is a generalization of the single atom iteractive calculation proposed earlier by Tewary [7]. The special case of the n-atom approximation when the region $\mathcal{R}$ contains a single atom ($n = 1$) corresponds to the single atom iteractive calculation.

In the n-atom approximation, for each region $\mathcal{R}_i$, the GFMD step involves the calculation of the first Taylor's coefficients and the construction of the dynamical $\widetilde{\mathbb{D}}$ matrix, the calculation of the eigenvalues and eigenvectors of $\widetilde{\mathbb{D}}$, matrix multiplications, and the update of the displacements and velocities using Eqs. (24) and (25). The construction of the dynamical matrix can be time consuming, especially for large three dimensional systems. However, if n is kept relatively small, the computational cost of determining $\widetilde{\mathbb{D}}$ can also be kept small, compared to the one of the diagonalization process. As we will see, the value of n depends on the number of the interacting neighbors for the atoms within the region $\mathcal{R}$.

In order to benchmark the n-atom approximation, we determined the evolution of an one-dimensional atomic chain with fixed boundary conditions considering only interactions between nearest neighbors. The N atoms of the chain have the same mass m ($m = 12$ amu) and are initially separated by 1Å. The central atom is then displaced from equilibrium by a distance d_0 ($d_0 = 0.1$ Å). The interaction between nearest neighboring atoms is described by the harmonic potential with a spring constant μ ($\mu = 55.25$ N/m). For this system, an exact solution is available for the displacements, which provides an appropriate way to measure the accuracy of the proposed method. In particular, the displacement of the central atom in a N-chain (N odd) is given by

$$u_0^{\text{exact}}(t) = \frac{d_0}{2J} \sum_{k_i} \cos[\omega(k_i)t], \qquad (26)$$

where $J = (N-1)/2$, $\omega^2(k) = 2(\mu/m)(1 - \cos k)$, k is the wave vector, and i is an integer varying from $-J$ to $J-1$. The allowed values of k_i are $(2i+1)\pi/2J$.

Algorithm 1 presents the steps to apply the n-atom approximation to a linear chain with N atoms. In the algorithm, $\widetilde{D}$ is a $n \times n$ matrix, calculated for each region $\mathcal{R}_i$. Fig. 1 illustrates the application of the n-atom approximation for a linear chain. Because of the fixed boundary used here and to keep the same size

7

for all the regions, the regions corresponding to the atoms close to the borders and their respective inner atoms (blue rectangle, Fig. 1b) are redefined before applying the GFMD (black rectangle, Fig. 1b). For example, to update the position and velocity of atom 1 (Fig.1 (b)), the region $\mathcal{R}_1$ is considered where the atom 1 will interact with atom 2 and with fixed wall (nearest neighbor interactions). For more general 3D systems, the regions $\mathcal{R}_i$ can be chosen to be spheres based on a cutoff radius dependent on the system. Alternatively, Verlet lists or cell list algorithms, already available in MD codes [9], can be used to provide the list of the atoms for each $\mathcal{R}_i$.
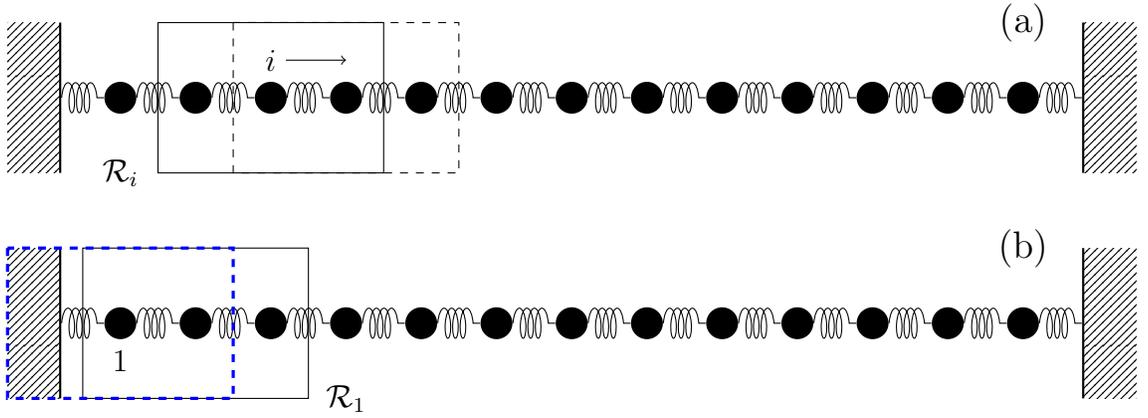
Figure 1: Illustration of the n-atom approximation for a 13-atom chain for $n = 3$. In (a) the rectangle indicates the region $\mathcal{R}_i$ respective to the atom i. For the linear chain, this atom is chosen to be in the middle of $\mathcal{R}_i$. After applying GFMD to $\mathcal{R}_i$, the region is moved to next site and the process is repeated (dashed rectangle). In (b) the blue dashed rectangle indicates the region for atom 1 ($\mathcal{R}_1$). Because the fixed boundary conditions, this region is redefined to the one represented by the black rectangle in order to keep the same size. Analogous procedure is used to the other side of the chain.

Algorithm 1: GFMD pseudo-code for N-atom chain using n-atom approximation.

---

1 Set up simulation parameters (time step size Δt, number of integration steps N_{steps});

2 Initialize $t \leftarrow 0$;

3 Set initial positions $x_i(0)$;

4 Set initial velocities $c_i(0) = 0$;

5 *Integration loop*;

6 **for** $step=1,\ldots,N_{steps}$ **do**

7   *Loop over the N atoms*;

8   **for** $i=-(N+1)/2,\ldots,-1,0,1,\ldots,(N+1)/2$ **do**

9   Build the list of atoms for the region $\mathcal{R}_i$;

10   Calculate the first Taylor's coefficients F_j of $\mathbb{F}_{n\times 1}$ with $x_j(t)$ for $j \in \mathcal{R}_i$ (n atoms) ;

11   Calculate $\widetilde{\mathbb{D}}_{n\times n}$ for the current configuration $\{x_j(t)\}$;

12   Diagonalize $\widetilde{\mathbb{D}}(t)$ to obtain the n eigenvalues $E_{\widetilde{j}}^2$;

13   Set $\mathbb{V}_{n\times n}$ with the normalized eigenvectors of $\widetilde{\mathbb{D}}(t)$;

14   Set $\mathbb{U} \leftarrow [0,\ldots,0]_{n\times 1}$;

15   Set $\mathbb{C} \leftarrow [c_j(t)]_{n\times 1}$;

16   Calculate $\mathbb{U}^*(t) \leftarrow \mathbb{V}^T\mathbb{U}(t)$;

17   Calculate $\mathbb{C}^*(t) \leftarrow \mathbb{V}^T\mathbb{C}(t)$;

18   Calculate $\mathbb{F}^*(t) \leftarrow \mathbb{V}^T\mathbb{F}(t)$;

19   Update $\mathbb{U}^*(t+\Delta t)$ using Eq.(24);

20   Update $\mathbb{C}^*(t+\Delta t)$ using Eq.(25);

21   Calculate $\mathbb{U}(t+\Delta t) \leftarrow \mathbb{V}\mathbb{U}^*(t+\Delta t)$;

22   Calculate $\mathbb{C}(t+\Delta t) \leftarrow \mathbb{V}\mathbb{C}^*(t+\Delta t)$;

23   Update future position of atom i with u_j ($j \in \mathcal{R}_i$, $j=i$): $x_i(t+\Delta t) \leftarrow x_i(t) + u_j(t+\Delta t)$;

24   Update future velocity of atom i with c_j ($j \in \mathcal{R}_i$, $j=i$): $c_i(t+\Delta t) \leftarrow c_j(t+\Delta t)$;

25   **end**

26   Update time $t \leftarrow t + \Delta t$.

27 **end**

The atomic trajectories were obtained by GFMD using the full diagonalization and using the n-atom approximation. We also obtained the trajectories with the commonly used velocity-Verlet method, representing here the conventional MD. In order to determine the accuracy of each of these methods in describing the exact solution, we calculated the cumulative squared difference between the displacement of the central atom, obtained from the testing method and from the exact solution (Eq. 26). The accuracy is then quantified by the

dimensionless parameter ξ given by

$$\xi(t) = \frac{1}{d_0^2} \sum_{p=1}^{N_{steps}} [u_0(p\Delta t) - u_0^{\text{exact}}(p\Delta t)]^2, \tag{27}$$

where N_{steps} is the necessary number of time steps to reach the time t ($t = N_{steps}\Delta t$), u_0^{exact} is the exact displacement (Eq. 26) and u_0 is the displacement of the central atom obtained by the full diagonalization, by the n-atom approximation, or by the velocity-Verlet method. The values of ξ is expected to increase with time due to the accumulation of truncation errors. However, ξ values can be compared among the three methods for a same time t.

In order to measure the energy conservation presented by the methods, we also quantified the cumulative difference in energy given by

$$\xi_E(t) = \frac{1}{E_T^2(0)} \sum_{p=1}^{N_{steps}} [E_T(p\Delta t) - E_T(0)]^2, \tag{28}$$

where $E_T(t)$ is the total energy at time t.

4. Parallel implementation

The computational cost of the n-atom approximation is expected to be of $\mathcal{O}(Nn^3)$ since N diagonalizations of a $n \times n$ matrix are necessary in each time step. However, the steps within the loop over the atoms in Algorithm 1 (line 8 to 25) involve calculations that can be carried out independently at the same time step. Thus, we applied shared memory techniques with OpenMP [10, 11] to parallelize the loop over the atoms. The loop steps were then distributed and executed in parallel by computational threads, which are responsible for executing a group of steps of approximately same size. All the simulations were carried out in a node with 24 Intel Xeon X5650 cores (2.67 GHz, 12 MB Cache) with 50 GB RAM [12]. Thus, the maximum number of threads were 24, the same number of available processing cores. Diagonalizations inside the loop were performed by the LAPACK `dsyev` routine [13]. The computational cost was quantified by the wall time spent by the integration loop (lines 6–27, Algorithm 1).

5. Results and discussions

The size of the $\mathcal{R}$ region is an important parameter for the accuracy of the n-atom approximation. Correct results were only obtained for n values equal to 3 or larger, with $\xi \simeq 10^{-5}$ and $\xi \simeq 10^{-7}$ for $N = 51$ and $N = 501$, respectively (Fig. 2, black data). A region with 3 atoms includes all the interacting neighbors when only the nearest neighbors are considered. It is expected when more atoms interact with the central atom of the $\mathcal{R}$ region, the size of $\mathcal{R}$ should be enlarged enough to cover all the interacting neighbors and, thus, guarantee the accuracy

10

of the n-atom approximation. To test this, we performed some simulations where the interaction of the next-nearest neighbors was included in the force calculations. Indeed, we observed that a larger region (larger n) is necessary to get convergence when more neighbors are included (Fig. 3).

Increasing the size of the region, further than the size of the interacting region, however, provides no additional improvement on the n-atom approximation accuracy. On the other hand, this increases the computational time (Fig. 2, blue data). For a fixed N, we found that the computational time is $\mathcal{O}(n^2)$ for $10 < n < 50$. These sizes are within the n-by-n double precision matrices that can fit in the Xeon X5650's 12 MB cache used here. This leads to a dependence of the computational time on n lower than the expected $\mathcal{O}(n^3)$ for larger matrices (in our case, for n larger than 1224). Thus, if the size of the $\widetilde{\mathbb{D}}$ matrices, which depends on the number of neighbors and on the dimensionality of the system, is such that cache memory can be explored, the computational cost of the n-atom approximation can be reduced to $\mathcal{O}(Nn^2)$.

The dependence of the computational time on the system size N for the GFMD with full diagonalization shows the expected $\mathcal{O}(N^3)$ for $N \gtrsim 300$ (Fig. 4). On the other hand, $\mathcal{O}(N)$ is obtained when the n-atom approximation is used. Considering the cost for the case when the cache memory can be explored, this represents a gain in the computational time of $\sim N^2/n^2$, which can be significant for $N \gg n$. For example, the gain is $\sim 10^6$ for $n \sim 10$ and $N \sim 10^4$.

OpenMP parallelization allowed reducing the computational time of the n-atom approximation even further. Fig. 5 illustrates this by showing the reduction when CPU threads are used for large chain ($N = 50001$). For 24 threads, the computational time for the parallel version is about 7% of the time of the serial version, which corresponds to a $14\times$ speedup. Optimization and fine tunning of the parallelization strategy probably can lead to larger speedups, but this was not explored in this work. For a fixed number of threads, the computational cost of the parallel version of the n-atom approximation exhibits the same linear dependence on the system size as the serial version (Fig. 4).

The parallel version of the n-atom approximation provided considerable reduction on the computational cost with respect to the full diagonalization process used in the original GFMD. In comparison with the velocity-Verlet method, the parallel version has the same $\mathcal{O}(N)$ dependence of the computational time but still presents a larger cost (Fig. 4). However, these methods present different accuracies. Whereas the ξ values obtained from the velocity-Verlet method are of the order of 10^{-6}, the values obtained from the n-atom approximation are of the order of 10^{-11} (for 200 time steps, Fig. 4). The evolution of ξ for a long (2×10^7 steps), 10 ns-simulation is shown in Fig. 6 (top). The difference between the velocity-Verlet method and the n-atom approximation decreases with time but the n-atom approximation still provides a better accuracy over the full time range. Results for the energy conservation (Fig. 6, bottom) also show a better performance of the n-atom approximation, with energy variations about $10\times$ smaller than the ones obtained from the velocity-Verlet method.

Finally, GFMD with the n-atom approximation allows one to use larger

11

time step sizes with better accuracy (smaller ξ values) than the ones necessary in velocity-Verlet method (Fig. 7, black data). Whereas increasing time step size from 0.01 fs up to 4 fs has almost no effect on trajectory accuracy for the n-atom approximation, the accuracy decreases for the velocity-Verlet method. For this entire range, the n-atom approximation exhibits better trajectory accuracy. The quality of the energy conservation starts to worsen after 4 fs for the n-atom approximation and 0.2 fs for the velocity-Verlet method (Fig. 7, blue data). These time step sizes can serve as the upper limit provided by each method for the linear chains studied here in order to guarantee trajectory accuracy and energy conservation. For the oscillation period of the linear chains studied here of about 120 fs, the maximum time step size provided by the n-atom approximation is 20 times larger than the one provided by the velocity-Verlet method.
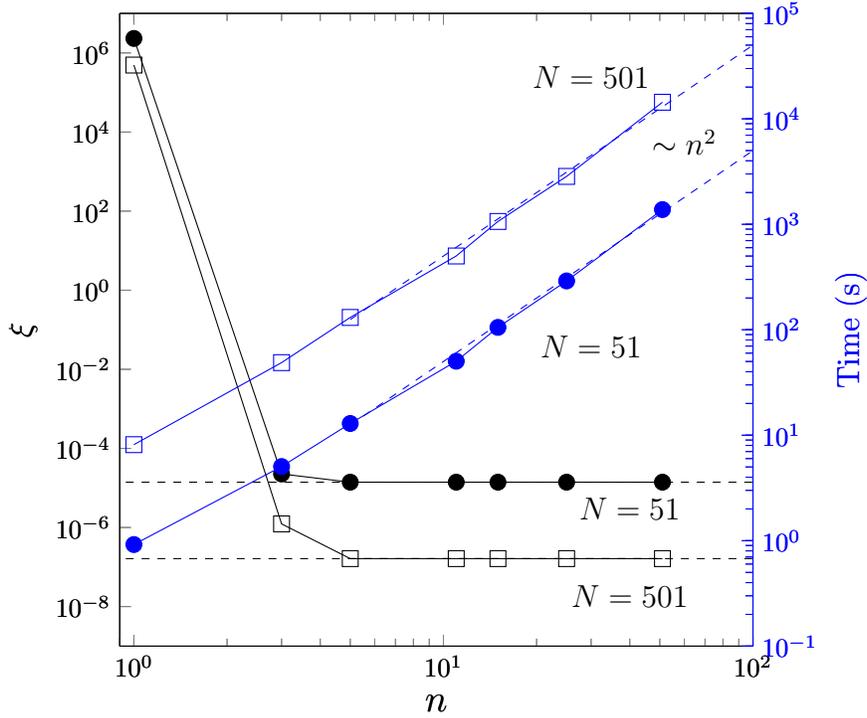
Figure 2: Convergence and computational cost for the n-atom approximation. (Black) Dependence of ξ as a function of the size n of region $\mathcal{R}$ for two linear chains. (Blue) Computational time for 20000 time steps as function of n. The blue dashed lines correspond to the behavior of the computational time $\sim n^2$ and the black dashed lines correspond to the ξ value when the full diagonalization is used.
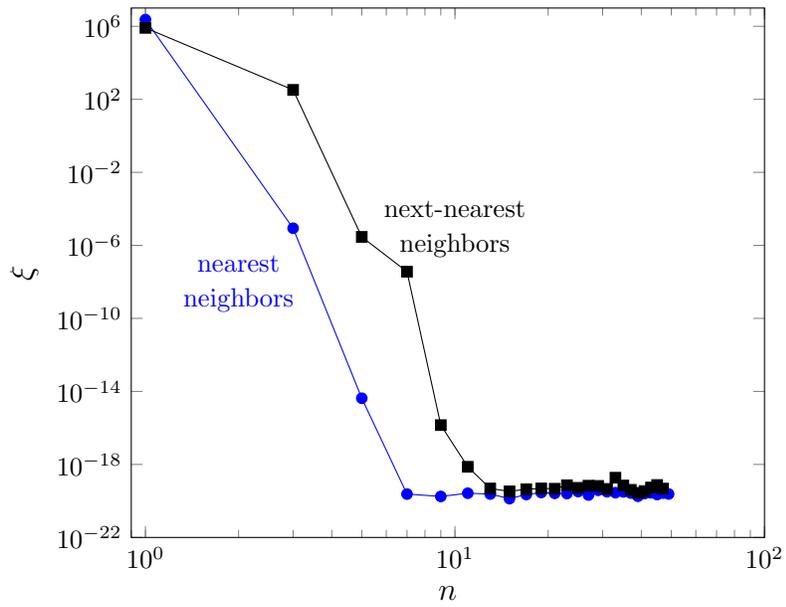
Figure 3: Influence of the interacting neighbors on the convergence of the n-atom approximation. Dependence of ξ on the size of region $\mathcal{R}$ for $N = 51$ taking into account nearest neighbors and next-nearest neighbors interactions. The values of ξ were obtained for 20000 time steps. For this figures, ξ was calculated taking the solution of the full diagonalization instead of the exact solution, which gives smaller values for ξ when compared to the values presented in Fig. 2.

13
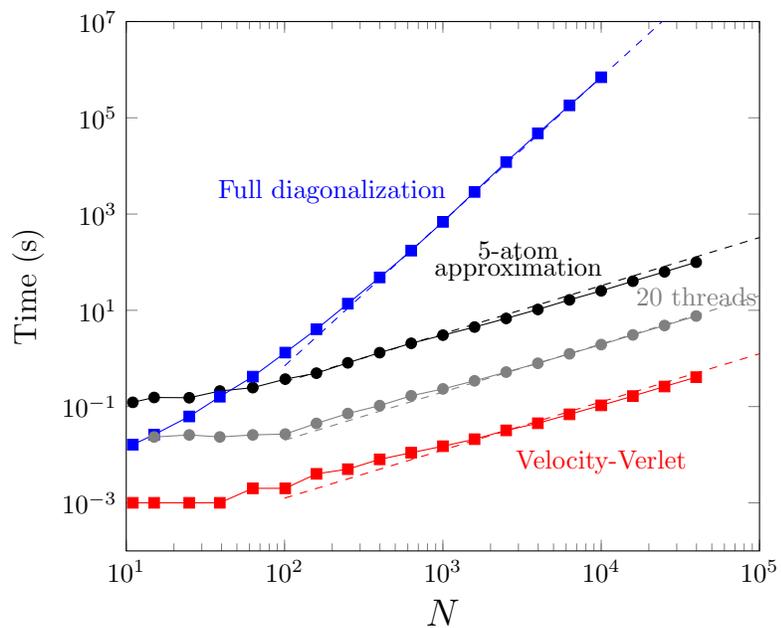
Figure 4: Dependence of the computational time on the system size N for full diagonalization (blue), serial (black) and parallel (gray, with 20 threads) versions of the 5-atom approximation, and velocity-Verlet (red) methods. The dashed lines correspond to times that are proportional to N^3 (blue) and to N (black, gray, and red). The times correspond to a run of 200 time steps using a time step size of 0.5 fs.
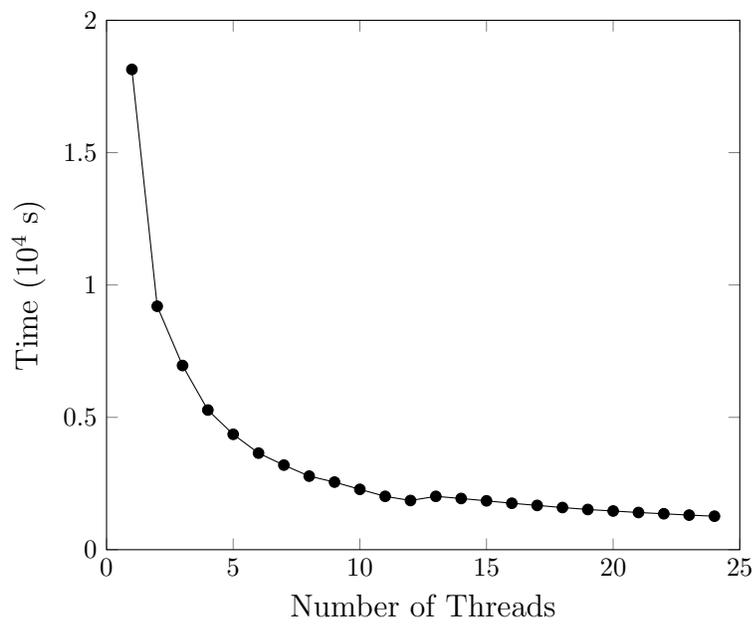
Figure 5: Dependence of the computational time on the number of threads for the 5-atom approximation with N=50001 atoms. The times correspond to a run of 20000 time steps.
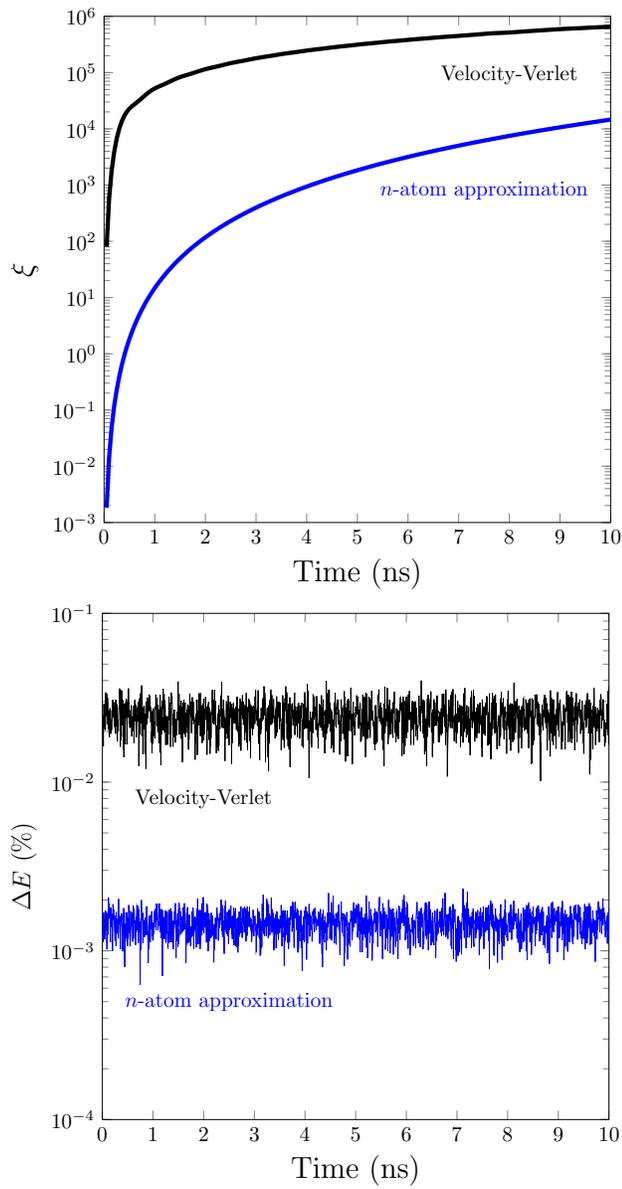
Figure 6: Evolution of ξ and of the difference of total energy with respect to the energy at $t = 0$ for the 5-atom approximation and the velocity-Verlet methods for a chain with 51 atoms (Δt= 0.5 fs).
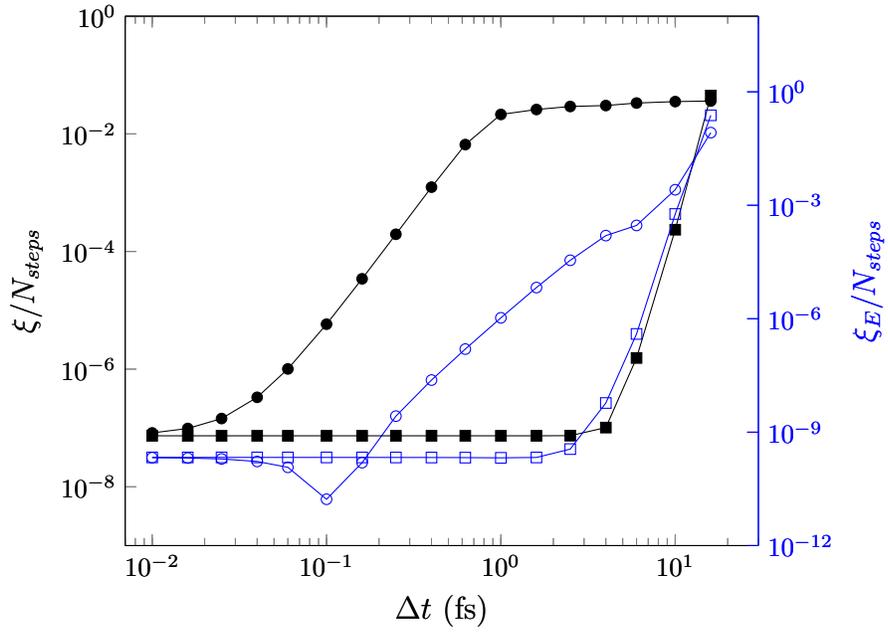
Figure 7: Values of ξ (left) and ξ_E (right) for 100 ps runs using different time step sizes Δt with the 5-atom approximation (filled and empty squares) and velocity-Verlet (filled and empty circles) methods (N=51). In order to compare the curves for the different time step sizes, the ξ and ξ_E values were normalized by the corresponding number of steps (N_{steps}) of each run. This is because different time step sizes require different N_{steps} for a fixed simulation time.

6. Conclusions

We have improved the Green's function molecular dynamics method by treating the full system of N atoms as a collection of N smaller systems of size n. This improved GFMD is then applied to these smaller systems to obtain the updated positions and velocities. This approach – the so-called n-atom approximation – has been parallelized using OpenMP and provides a $\mathcal{O}(N)$ computational time with accuracy and energy conservation better than the ones obtained from the commonly used velocity-Verlet integration algorithm. The proposed approach also allows the use of larger time step sizes and can be used as an alternative integration algorithm for molecular dynamics simulations, especially for studying vibrational phenomena.

7. Acknowledgments

References

[1] B. P. Uberuaga, F. Montalenti, T. C. Germann, A. F. Voter, Accelerated Molecular Dynamics Methods, Springer Netherlands, Dordrecht, 2005, pp. 629–648. doi:10.1007/978-1-4020-3286-8_32.
URL https://doi.org/10.1007/978-1-4020-3286-8_32

[2] J. Durrant, J. A. McCammon, Molecular dynamics simulations and drug discovery, BMC Biology 9 (1) (2011) 71. doi:10.1186/1741-7007-9-71.
URL http://www.biomedcentral.com/1741-7007/9/71

[3] K. L. Joshi, S. Raman, A. C. T. Van Duin, Connectivity-based parallel replica dynamics for chemically reactive systems: From femtoseconds to microseconds, Journal of Physical Chemistry Letters 4 (21) (2013) 3792–3797. doi:10.1021/jz4019223.

[4] A. Binder, T. Lelièvre, G. Simpson, A generalized parallel replica dynamics, Journal of Computational Physics 284 (0) (2015) 595–616. doi:http://dx.doi.org/10.1016/j.jcp.2015.01.002.
URL http://www.sciencedirect.com/science/article/pii/S0021999115000030

[5] A. F. Voter, Hyperdynamics: accelerated molecular dynamics of infrequent events, Phys, Rev. Lett. 78 (20) (1997) 3908–3911. doi:10.1103/PhysRevLett.78.3908.

[6] M. R. Sorensen, A. F. Voter, Temperature-accelerated dynamics for simulation of infrequent events, Journal of Chemical Physics 112 (21) (2000) 9599–9606. doi:10.1063/1.481576.

18

[7] V. K. Tewary, Extending the time scale in molecular dynamics simulations: Propagation of ripples in graphene, Phys. Rev. B 80 (2009) 161409. `doi:10.1103/PhysRevB.80.161409`.
URL `http://link.aps.org/doi/10.1103/PhysRevB.80.161409`

[8] V. R. Coluci, S. O. Dantas, V. K. Tewary, Generalized green's function molecular dynamics for canonical ensemble simulations, Phys. Rev. E 97 (2018) 053310. `doi:10.1103/PhysRevE.97.053310`.
URL `https://link.aps.org/doi/10.1103/PhysRevE.97.053310`

[9] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1) (1995) 1–19. `doi:https://doi.org/10.1006/jcph.1995.1039`.
URL `https://www.sciencedirect.com/science/article/pii/S002199918571039X`

[10] L. Dagum, R. Menon, Openmp: An industry-standard api for shared-memory programming, IEEE Comput. Sci. Eng. 5 (1) (1998) 46–55.

[11] R. Chandra, Parallel programming in OpenMP, High performance computing, Morgan Kaufmann Publishers, 2001.

[12] Certain commercial equipment, software and/or materials are identified in this paper in order to adequately specify the experimental procedure. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment and/or materials used are necessarily the best available for the purpose.

[13] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.