To appear in *Molecular Simulation*
Vol. 00, No. 00, Month 20XX, 1–13

Efficient recursive Adams-Bashforth methods in Molecular Dynamics simulations of N-body systems interacting through pairwise potentials

Jordi Martí[a] and Bernat Diaz[b]

[a]*Department of Physics, Technical University of Catalonia-Barcelona Tech, B5-209 Northern Campus UPC. Jordi Girona, 1-3. 08034 Barcelona. Catalonia. Spain;* [b]*Faculty of Computer Science, Technical University of Catalonia-Barcelona Tech. Jordi Girona, 1-3. 08034 Barcelona. Catalonia. Spain*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

A recursive multistep Adams-Bashforth method applied to the Molecular Dynamics simulations of N-body systems interacting through pairwise force fields is introduced and analysed. Equations of motion are obtained using a set of Cartesian coordinates solved by means of an Adams-Bashforth numerical integration scheme of order s, which requires the iterative computation of function time derivatives. The proposed algorithm has been implemented using a programming approach that makes it possible to re-use a source code resulting in small codes, easy to maintain. Practical examples and benchmarks that illustrate the performance of these implementations are included. The study of its performance gives clues to evaluate its efficiency and precision. Numerical tests for a N-particle system are made on the equilibrium configuration of liquid argon near its triple point at 86.5 K and 0.021 $\AA^{-3}$. In most cases, the algorithms here presented outperform those implemented traditionally as the Gear corrector-predictor or the Verlet family, leading to important savings in terms of total computation times and significantly increasing the numerical precision obtained with standard algorithms.

Keywords: RECURSIVE FORMULATION; ADAMS-BASHFORTH METHODS; INTEGRATION ALGORITHMS; MOLECULAR DYNAMICS

1. Introduction

Time-dependent simulation of N-body systems of any size is a challenging problem mainly because the force field governing the underlying dynamics is related to interactions between all particles. It is well known that unlike two-body problems, no general closed-form solution exists for the three or higher N cases, as the resulting dynamical system is chaotic for most initial conditions, so that numerical methods are required. For the microscopic case, the forces between N particles are usually unknown and, therefore, choosing a reliable model for the interactions is crucial for the success and efficiency of the numerical calculations. Further, the use of efficient integration algorithms is relevant for the precision of the dynamical simulations in real time and also for their performance in terms of computational costs.

The search for efficient integration algorithms applied to Molecular Dynamics (MD) simulations has already a long history. After the seminal contributions by Verlet[1, 2], another classical pioneering work was due to Beeman[3], who considered an Adams-Bashforth-Moulton predictor-corrector procedure and compared it to an algorithm using the prescription of Rahman[4], but only up to third order, because problems of storage prevented him to consider higher orders. In the framework of the solution of ordinary differential equations, Gear[5] provided a comprehensive treatise on a wide variety of methods and their applications to all sort of mathematical and physical

Email: jordi.marti@upc.edu

problems, with particular interest due to his own proposal of a specific predictor-corrector routine for simulations. Later on, Howe[6] focussed his attention on the comparison of a new two-pass predictor-corrector integration algorithm and the Adams-Moulton family of algorithms, finding ways to apply these algorithms to real-time inputs, such as in computer simulations.

Regarding the Adams-Bashforth and leapfrog-like algorithms, there is a contribution by Durran[7] in a purely mathematical context, where the Asselin-filtered leapfrog scheme is compared to the third-order Adams–Bashforth method in the framework of ordinary differential equations. A more recent contribution by Rodriguez et al.[8] reported a semi-recursive procedure based on velocity transformation between Cartesian and relative velocities in the context of macroscopic multi-body systems.

The approach reported here has not been proposed before in the same terms. Ours is a novel contribution based on the consideration of a recursive Adams-Bashforth predictor routine of variable arbitrary order, explored here between orders 2 and 6, and the comparison of its performance to the likes of the standard leapfrog Verlet algorithm (LFV) and the Gear predictor-corrector (GPC). In order to establish the reliability of our proposal, we have considered the three algorithms and applied them to MD simulations of liquid argon (Ar) at its triple point. Also, mechanical textbook problems such as the motion of an object of mass m in a non-relativistic gravitational field and the motion of the same mass attached to a totally flexible spring i.e. the simple harmonic oscillator are used as benchmarks for the determination of the precision of the integration algorithms considered in the present work.

2. Computational Methods

The natural, primary ensemble where MD simulations construct trajectories is the microcanonical ensemble, that is, with the system kept at constant energy, volume and number of particles, following Newtonian dynamics. Newton's equations of motion describe with all precision the motion of massive classical particles in gravity fields, such as the one at the surface of Earth. So we can use it as a textbook benchmark to refine our integration algorithms. In Newtonian dynamics, the equations of motion of a system of N particles are given by:

$$\ddot{\mathbf{r}}_i = -\frac{1}{m_i}\vec{\nabla} V_i(\mathbf{r}) \tag{1}$$

where m_i and $\mathbf{r}_i$ are the mass and the vector coordinates of the i-th particle and $V_i(\mathbf{r})$ is the interparticle potential. This system can be transformed into a system of two first-order ordinary equations through the relationships

$$\begin{aligned} \dot{\mathbf{v}}_i &= -\frac{1}{m_i}\vec{\nabla} V_i(\mathbf{r}), \\ \dot{\mathbf{r}}_i &= \mathbf{v}_i. \end{aligned} \tag{2}$$

Numerical methods for the solution of ordinary differential equations given by Eq.2 are generally solved by algorithms falling within the multistep family with either a Runge-Kutta or an Adams-Bashforth-Moulton formulation (see for instance Refs.[5, 9]). In the general case of multistep methods (which include Adams-Bashforth, Adams-Moulton and backward differentiation formula, as the main families) the solution of the equations of type $y'(t) = f(t, y)$ consists of assuming that initial values of the function $y(t)$ and its first derivative $y'(t)$ are known and that the following values can be computed from linear combinations of y and y':

$$\begin{aligned} y_{n+s} + a_{s-1} \cdot y_{n+s-1} + a_{s-2} \cdot y_{n+s-2} + \cdots + a_0 \cdot y_n &= h \left[b_s \cdot f(t_{n+s}, y_{n+s}) + \right. \\ &+ b_{s-1} \cdot f(t_{n+s-1}, y_{n+s-1}) + \cdots + \\ &+ \left. b_0 \cdot f(t_n, y_n) \right], \end{aligned} \tag{3}$$

or, equivalently,

$$\sum_{j=0}^{s} a_j \cdot y_{n+j} = h \sum_{j=0}^{s} b_j \cdot f(t_{n+j}, y_{n+j}), \tag{4}$$

where $h \equiv t_1 - t_0$, i.e. the (arbitrary) size of the grid in the abscissae axis. In the case of $b_s = 0$ (explicit methods) the solution is directly computable, such as in the Adams-Bashforth family, where $a_{s-1} = -1$ and $a_{s-2} = ... = a_0 = 0$. The b_j for $j = 0, ...s-1$ can be obtained by polynomial interpolation in order to find a polynomial acting as the function $f(t, y)$ in a finite series of time instants: $p(t_{n+i}) \equiv f(t_{n+i}, y_{n+i})$, for $i = 0, ..., s-1$. Using Lagrange formula, we get:

$$p(t) = \sum_{j=0}^{s-1} \frac{(-1)^{s-j-1} f(t_{n+j}, y_{n+j})}{j! \cdot (s-j-1)! h^{s-1}} \prod_{i=0, i \neq j}^{s-1} (t - t_{n+i}). \tag{5}$$

Since $p(t)$ is a good local approximation of the function $f(t, y)$, we can use the relationship:

$$y_{n+s} = y_{n+s-1} + \int_{t_{n+s-1}}^{t_{n+s}} \mathrm{d}t \, p(t). \tag{6}$$

Replacing $f(t, y)$ by $p(t)$, we get the coefficients b_j as:

$$b_{s-j-1} = \frac{(-1)^j}{j! \cdot (s-j-1)!} \int_0^1 \mathrm{d}u \prod_{i=0, i \neq j}^{s-1} (u + i), \qquad j = 0, ..., s-1. \tag{7}$$

Here we are assuming an error of the order h^s. The particular recipes used in the present work concern the recursive forms for positions and velocities. For a time instant t_i, we provide the following operative formulas for the recursive Adams-Bashforth (RAB) algorithm of order s, with $j < s$:

$$\begin{aligned} v(t_i+dt) &= v(t_i)+dt\cdot\sum_{j=0}^{s-1} b_{s-j-1}\cdot a(t_i-j\cdot dt) \\ &= v(t_i)+dt\cdot[b_{s-1}\cdot a(t_i)+b_{s-2}\cdot a(t_i-dt)+\cdots \\ &\quad +b_1\cdot a(t_i-(s-2)dt)+b_0\cdot a(t_i-(s-1)dt)] \\ r(t_i+dt) &= r(t_i)+dt\cdot\sum_{j=0}^{s-1} b_{s-j-1}\cdot v(t_i-j\cdot dt) \\ &= r(t_i)+dt\cdot[b_{s-1}\cdot v(t_i)+b_{s-2}\cdot v(t_i-dt)+\cdots \\ &\quad +b_1\cdot v(t_i-(s-2)dt)+b_0\cdot v(t_i-(s-1)dt)]\,. \end{aligned} \tag{8}$$

Note that here $h \equiv dt$ i.e. the time step of the simulations and it is constant. For any integration algorithm there are relevant questions to address such as its time-reversibility or its symplecticity, as it is described in details in the conference book by Marsden et al.[10], aspects which, at the level of detail reported in the present paper, can be only discussed in a general way. On the one hand and on invariance under time-reversibility, we must note that RAB is initially designed to move the system forward in time, since it is based in getting positions and velocities (see eq.8) from previous instants, such as most of standard procedures. In this work we did not test time-reversibility of the RAB algorithm (for instance by changing $dt \to -dt$, what could become a topic for a future study). On the other hand, RAB can be considered symplectic in a general sense, since it serves as a numerical integration scheme for a Hamiltonian system. Nevertheless, concerning conservation properties we must state that we have only checked energy conservation and observed (see results from Table 4) a good stability at all orders ($s = 2, ..., 6$). Despite this fact, we believe we cannot ensure symplecticity in a more strict way without a deeper analysis, which is out of the scope of this work. As benchmarks for this method, we have considered the Gear predictor-corrector algorithm[5] for some properties and also the well known and widely used LFV algorithm in all cases, applied to a personal MD code and also used within a simulation package (NAMD). For LFV latter the equations are:

$$\begin{aligned} v(t_n+\frac{dt}{2}) &= v(t_n-\frac{dt}{2})+dt\cdot a(t_n) \\ r(t_n+dt) &= r(t_n)+dt\cdot v(t_n+\frac{dt}{2}), \end{aligned} \tag{9}$$

where the velocities at time $t_n - \frac{dt}{2}$ are computed from

$$v(t_n-\frac{dt}{2}) = \frac{r(t_n)-r(t_n-dt)}{dt}. \tag{10}$$

About the characteristics of the simulations, we considered MD trajectories of 100 ps for liquid Ar samples formed by 500 atoms at the triple point (86.5 K, particle density of 0.021 $\AA^{-3}$) and 1 atm of pressure and of 25 s for the classical mechanical systems (gravitational field and harmonic oscillator). In the former case, the time step δt was taken of 2 fs and in the latter it was in the range 10^{-5} to 1 s. For the molecular dynamics simulations, periodic boundary conditions were taken in all three directions of space. The force field for Ar was the standard Lennard-Jones 6-12 [11] with parameters $\sigma = 3.405$ $\AA$ and $\varepsilon = 119.8$ K, as it was taken in the first numerical simulations of liquid argon[1, 2, 12].

3. Performance of leapfrog Verlet, Gear predictor-corrector and recursive Adams-Bashforth algorithms for benchmark systems

The first step in the study of the performance of the different integration algorithms is the establishment of benchmarks able to validate their precision and efficiency. We have chosen two textbook simple models such that we can compare our results to "exact" ones, i.e. those derived from the algebraic solutions of the differential equations. These are: (1) the problem of a given mass m moving inside the (non-relativistic) gravitational field close to Earth's surface and (2) the simple, non-interacting classical harmonic oscillator. In the first case, equations 1 or 2 with $\ddot{\mathbf{r}}(t) = (0, -g)$ (for $g = 9.8$ m/s^2 for instance) should be solved and in the second case, the traditional Newton equation for the Hooke's law $m{\cdot}\ddot{r}_y(t) = -k{\cdot}y(t)$ for a mass m attached to a spring of elastic constant k along the Y-axis gives us the solution for the position and velocity of the harmonic oscillator in terms of pure harmonic terms, namely $y(t) = y_0 \sin(\omega t + \phi_0)$ and $v_y(t) = -y_0\omega \cdot \cos(\omega t + \phi_0)$, where $\omega = \sqrt{(k/m)}$. Of course the solution of the first case is given by the typical parabolic trajectory of coordinates: $(x_0 + v_{0x} \cdot t, y_0 + v_{0y} \cdot t - \frac{1}{2}gt^2)$ and velocity $(v_{0x}, v_{0y} - gt)$.

Once remembered this, we can perform several comparisons on the precision and efficiency of each algorithm concerning the size of time step (and correspondingly the conservation of such precision during "long" trajectories) as well as on the order taken for the Adams-Bashforth recursion. First, let us consider the two algorithms and their accuracy as a function of the time step. We have reported in Table 1 the results concerning the estimated errors found in the two benchmark systems described above.

We considered the difference between the computed and algebraic values, as sketched above, as the "error" $\epsilon \equiv y_{theoretical} - y_{calculated}$. Although the harmonic oscillator is more sensitive then the gravitational field to these particular precision issues, in both cases the errors are large for long time steps ($\delta t = 1, 0.1$ s), as expected. Errors are significantly smaller as time step size decreases, so that for $\delta t = 0.001$ highest errors in the harmonic oscillator case are of the order of centimetres (see Table 1). For the minimum time step considered ($\delta t = 10^{-5}$ s), the magnitude of errors are very similar in the case of the gravitational field. In the case of the harmonic oscillator, we found an error for the leapfrog Verlet algorithm remarkably larger than for the recursive Adams-Bashforth, which is in agreement with the fact that Verlet family of algorithms (due to the second order truncation of Taylor series) is not well suited to the case of the harmonic oscillator. It has been shown that due to the truncation, velocities are systematically underestimated whereas the turning points of the period are overestimated (see more details in Refs.[13–15]). Our results show such drawbacks in a plain and direct way, but more importantly, indicate that third order RAB algorithm gives better stability and precision than LFV for the harmonic oscillator. As we will show below, taking higher orders in the RAB algorithm do not improve significantly its accuracy. For the sake of comparison another widely used algorithm, GPC, was also considered. We can observe that it shows the largest errors, about twice those of the LFV.

As an alternative view, cumulative errors along long trajectories show a similar tendency, as it is indicated by data reported in Table 2. As expected, the errors grow as the simulation time progresses, in a very regular way in both cases, although we can observe that the RAB method with $s = 3$ is able to maintain the error nearly constant, whereas LFV and GPC have a poorer behavior and a tendency to larger errors, a consequence of the truncation indicated above.

As a final use of the benchmark mechanical models, we have compared the accuracy of the RAB algorithm as a function of s through results reported in Table 3. We can observe that in both cases order $s = 2$ gives worse performance than $s = 3$ but that including higher orders ($s > 3$) does not produce a better agreement between computed and exact results. For this reason we decided to use $s = 3$ in the previous testing of time step values (Tables 1 and 2).

4. Performance of leapfrog Verlet, Gear predictor-corrector and recursive Adams-Bashforth algorithms for liquid argon

After characterisation of the RAB algorithm using the two benchmark cases, we have applied the acquired knowledge to a N-body interacting system well known in the literature, namely the case of liquid argon, that has been studied for decades and that will work as a third testing case. The most relevant difference with the trivial benchmarks employed above is that the N-body system does not have an exact solution. Instead, we have considered experimental information on total (internal) energy of argon[16, 17], on its microscopic structure[18] and on its diffusion coefficient[19]. The main parts of the code we have used are reported in Figure 1.

The first comparison we can do is on the values for the total energy of the system. The numbers are reported in Table 4. As expected, the force field employed in the present work gives values very close for all the algorithms studied, but showing some differences with experimental data, between 0.5 and 3.8 kJ/mol. This is an indication of the reliability of the model concerning thermodynamics of the system. Better results were obtained by Barker et al.[20] using three force fields different to the one considered here, including quantum corrections. However, the reliability of potential models has already been established and it is not the goal of the present work. Furthermore, we should point our that the thermodynamic conditions are not the same for the simulation and the experiments and that, as shown by Gosman et al.[17], slight changes in the temperature or pressure of the samples can lead to sensible changes of the total energy of the system. Besides, no significant energy drift has been observed in any simulation with the RAB integrator, with averaged values and standard deviations of the order of less than 1% in each case for trajectories of the order of 100 ps, as reported in Table 4.

Concerning the microscopic structure of liquid argon, we have computed argon-argon radial distribution functions (rdf) $g(r)$, shown in Figure 2. They are an excellent match of the experimental rdf at 85 K, reported in Ref.[18], what forced us to focus on their first and second maxima and also the first minimum of the rdf in order to find out the tiny differences between the benchmark LFV algorithm and the family of RAB with variable order s. We also included the value reported by one of most employed simulation packages, namely the NAMD code[21] for the sake of comparison. It is well know that NAMD employs a standard LFV algorithm. In the version we employed here, the force field included in the package was CHARMM36[22]. As expected, results of LFV and the NAMD package are almost an exact match. The results of Figure 2 indicate that:

(1) In all cases the height of the selected maxima and minimum shows a deviation between 0.8 and 3.5 %;
(2) The position r of the first maximum (labelled 'A' in Fig. 2) is fairly well reproduced (perfect match around 3.7 Å), but the agreement becomes worse at larger r, for points 'B' and 'C' of Fig. 2, with deviations of about 2% in both cases. This might be a drawback of the potential model but also it can be related with the more difficult determination of the experimental rdf at large distances, which is correlated with the regions of low momenta in the primary structure factors determined experimentally, prior to be Fourier-transformed to obtain the rdf;
(3) The computed rdf closest to the experiment is the one obtained with RAB with $s = 3$. Furthermore, the increase of order in the RAB algorithm does not produce enhancement of the agreement with experimental position of the highlighted features, being the worse overall agreement for the case $s = 2$.

We have also checked the performance of the two classes of algorithms (LFV, RAB family) on dynamical properties of liquid argon, including the results produced by a standard MD package such as NAMD. Again the results of LFV and the NAMD package are extremely close. In particular, we explored argon self-diffusion coefficients, obtained from the well-known long time slopes of the mean square displacements of argon atoms (Einstein's formula), argon velocity autocorrelation functions $C(t) =< \vec{v}(0) \cdot \vec{v}(t) >$ and argon spectral densities $S(\omega)$, computed as the Fourier transforms of $C(t)$[23]:

$$S_i(\omega) = \int_0^\infty \mathrm{d}t \, C(t) \, \cos(\omega t). \tag{11}$$

The reported diffusion coefficients D_{Ar} are reported in Table 5. There we can observe how, for this particular property, LFV gives a better agreement with experimental data than RAB, although again the order of the algorithm able to produce (slightly) better results is $s = 3$. The performance of the algorithms considered in this work is compared again for the argon velocity autocorrelation functions and the corresponding spectral densities $S(\omega)$ of argon, computed as the Fourier transform for the former. The results are reported in Figure 3. There we can observe remarkable agreement between LFV and RAB. In this particular property we highlighted the first minimum (point 'A' of Figure 3, related to backscattering) and the region labelled 'B', where $C(t)$ shows a change of behavior including local maximum and minimum features. Assuming LFV as the benchmark (since it is the standard algorithm most employed in literature, including its use in popular MD simulation packages such as GROMACS[24], LAMMPS[25] or NAMD[21]), again the closest RAB order is $s = 3$ (point 'A') or, in this particular case, $s = 6$ for both points 'A' and 'B'. In the case of $S(\omega)$, only the maximum is closely explored (see inset at right side of Figure 3), giving a very close agreement of all cases with the results obtained with the LFV algorithm, but perhaps with the best agreement obtained for $s = 2$. Here we should point out that the single maximum appearing in the spectral densities is due to restricted translations of argon atoms inside the cage formed by their nearest neighbors, as it has been shown before after some debate, for the case of liquid water[26–28].

Our final comparison concerns the CPU time per time step of the simulations. The results are reported in Table 6. We performed the velocity checks in a single-node CPU in all cases. The results indicate a performance of LFV slightly better than RAB, as expected due to the larger number of calculations required in the latter case. However, the difference is not large and it tends to increase in a quasi-linear fashion with the order s of the RAB algorithm. GPC is the slowest algorithm at all, with CPU times about 52 % longer than the former. The comparison to the CPU time obtained with a parallel calculation using the NAMD package with 8 nodes, indicates that the time is largely reduced in a factor 45 when compared to the benchmark case of the RAB of order $s = 3$, what cannot be charged on the performance of the algorithm but to the superior computational architecture.

5. Conclusions

We have introduced a new prescription in the so-called Adams-Bashforth predictor algorithm originally introduced as a solver of ordinary differential equations in order to be employed in standard MD simulations of N-body systems. This method should be adapted in most simulations using pairwise potentials, but we should keep in mind that other multidimensional methods able to explore rare events in classical dynamics such as transition path sampling[29] can use a wide variety of algorithms to generate new configurations around stable states in the phase space whose dynamics may be also based on recursive or semi-recursive classical algorithms such as those presented here.

We have considered a simple formulation very easy to implement in all sort of MD codes, described by Eq. 8, with order s between 2 and 6. After its comparison with two standard algorithms widely used in MD such as leapfrog Verlet and Gear predictor-corrector, we have first established that RAB is more exact when applied to the macroscopical problems of the single mass point inside a non-relativistic gravitational field and of the simple harmonic oscillator. In order to evaluate its performance for a N-body microscopic problem typical of MD simulations, we analysed the case of liquid argon at its triple point. Our results indicate that the optimal order of calculation is $s = 3$ and that in such case the radial distribution function of argon matches very well the experimental values, whereas diffusion coefficient and the spectral density of argon are in excellent agreement with results from leapfrog Verlet. No significant energy drift has been observed in any simulation with the RAB integrator. In summary, we have found that the performance of RAB in terms of CPU time matches that of leapfrog Verlet, being both algorithms more than 50% faster than Gear predictor-corrector.

Due to the simplicity of the proposed algorithm, it would be straightforward for the teams responsible of the maintenance of large simulation packages to replace the actual standard leapfrog algorithms included in most academic simulation free access packages (LAMMPS, NAMD, GROMACS) by the RAB integrator of order $s = 3$ in order to gain further precision without significant loss of performance. For this purpose, we have provided the core parts of the code employed in the present work in Figure 1. A further relevant test for the algorithm RAB reported in the present work would be its use combined with an *Embedded Atom Model* potential as those proposed by Mishin and coworkers to model metal-metal interactions such as in copper[30], nickel-aluminium[31] or cobalt[32] to cite a few, which can easily run in a LAMMPS environment.

Acknowledgement

JM gratefully acknowledges financial support from Spanish Ministry of Science, Innovation and Universities (project number PGC2018-099277-B-C21, funds MCIU/AEI/FEDER, UE).

Tables

Table 1. Comparison of the efficiency of simulation time steps (δt) for benchmark models (point mass in a gravitational field and simple harmonic oscillator) for the tested algorithms (leapfrog Verlet, Gear predictor-corrector and recursive Adams-Bashforth with $s = 3$). Errors (ϵ) are defined as the difference between exact and computed positions (see Section 4).

Model	δt (s)	ϵ_{LFV} (m)	ϵ_{GPC} (m)	ϵ_{RAB} (m)
Gravitational field	1	5.4×10^{-4}	–	0.392
	0.1	4.7×10^{-6}	–	0.048
	0.01	7.4×10^{-7}	–	0.00049
	0.001	7.2×10^{-8}	–	4.9×10^{-6}
	0.0001	6.9×10^{-9}	–	4.8×10^{-8}
	0.00001	5.3×10^{-9}	–	5.7×10^{-9}
Harmonic oscillator	1	> 10	> 10	> 10
	0.1	> 10	> 10	> 10
	0.01	1.1	2.1	0.6
	0.001	0.01	0.02	0.006
	0.0001	1.1×10^{-4}	2.1×10^{-4}	6×10^{-5}
	0.00001	1.1×10^{-6}	2.1×10^{-6}	6×10^{-7}

Table 2. Precision of the algorithms (LFV, GPC, RAB) as a function of the time length of a trajectory. Errors (ϵ) are defined as in Table 1. Time step of 10^{-5} s.

Model	Time length (s)	ϵ_{LFV} (m)	ϵ_{GPC} (m)	ϵ_{RAB} (m)
Gravitational field	5	6.3×10^{-9}	–	6.6×10^{-9}
	10	2.5×10^{-8}	–	2.5×10^{-8}
	15	5.5×10^{-8}	–	5.5×10^{-8}
	20	9.6×10^{-8}	–	9.7×10^{-8}
	25	1.5×10^{-7}	–	1.5×10^{-7}
Harmonic oscillator	5	8.5×10^{-7}	1.7×10^{-6}	4.8×10^{-7}
	10	1.7×10^{-6}	1.7×10^{-6}	3.4×10^{-7}
	15	2.5×10^{-6}	5.1×10^{-6}	4.9×10^{-7}
	20	3.4×10^{-6}	$6.8\ \times10^{-6}$	4.9×10^{-7}
	25	4.2×10^{-6}	8.5×10^{-6}	4.9×10^{-7}

Table 3. Precision of the RAB algorithm as a function of the order s. Errors (ϵ) are defined as in Tables 1 and 2. Time step of 10^{-5} s.

Model	s	ϵ (m)
Gravitational field	2	7.45336×10^{-9}
	3	7.45327×10^{-9}
	4	7.45353×10^{-9}
	5	7.45337×10^{-9}
	6	7.45339×10^{-9}
Harmonic oscillator	2	1.37×10^{-5}
	3	7.97×10^{-7}
	4	7.98×10^{-7}
	5	7.98×10^{-7}
	6	7.98×10^{-7}

Table 4. Total energy (U) of liquid argon, in kJ/mol with statistical errors in parenthesis. Temperature (T, in Kelvin) and pressure (P, in atm) of each system are also reported. For the sake of comparison, the result of the CHARMM36 force field obtained by means of simulations made with the NAMD package has been also included.

System	T	P	U
Experiment[16]	100	3.2	5.46
Experiment[17]	91	1.0	8.79
LFV	86.5	1.0	4.97 (0.05)
GPC	86.5	1.0	4.97 (0.05)
NAMD	86.5	1.0	4.91 (0.03)
RAB,$s = 2$	86.5	1.0	4.95 (0.04)
RAB,$s = 3$	86.5	1.0	4.95 (0.05)
RAB,$s = 4$	86.5	1.0	4.95 (0.05)
RAB,$s = 5$	86.5	1.0	4.95 (0.06)
RAB,$s = 6$	86.5	1.0	4.95 (0.06)

Table 5. Liquid argon self-diffusion coefficients (D, all values are expressed in units of 10^{-5} cm^2/s). Temperature (T, in Kelvin) and pressure (P, in atm) of each system are also reported. The result of the CHARMM36 force field obtained by means of simulations made with the NAMD package has been also included.

System	T	P	D
Experiment[19]	90	1.32	2.43
LFV	86.5	1.0	3.50
GPC	86.5	1.0	4.04
NAMD	86.5	1.0	3.90
RAB,$s = 2$	86.5	1.0	4.08
RAB,$s = 3$	86.5	1.0	3.98
RAB,$s = 4$	86.5	1.0	4.04
RAB,$s = 5$	86.5	1.0	4.05
RAB,$s = 6$	86.5	1.0	4.03

Table 6. CPU times (t$_{CPU}$, in ms) per time step of each simulation. For the sake of comparison, the result of the CHARMM36 force field obtained by means of the package NAMD has been also included. A parallel mode calculation with 8 cores has been used, whereas the remaining calculations were made with sequential single-processor runs.

System	t$_{CPU}$
LFV	28.79
GPC	44.28
NAMD	0.657
RAB,$s = 2$	29.60
RAB,$s = 3$	30.22
RAB,$s = 4$	33.03
RAB,$s = 5$	35.05
RAB,$s = 6$	36.10

Figures

```
C
C     Coming from main loop...
C
C     s1 order of the recursive Adams-Bashforth integrator
C     with s1=3, optimal value for Lennard-Jones systems
C
      a_prev(:,:,1) = accel
      v_prev(:,:,1) = vinf
      do si = 1,s1 - 1
         do j = 0,si - 1
            call b_coefficient(si, j, bi)
            b(j + 1) = bi
         end do
         call rab(b,si,natoms,r,vinf,accel,v_prev,a_prev,deltat,nf)
         a_prev(:,:,si + 1) = accel
         v_prev(:,:,si + 1) = vinf
      end do
C
C     end of main
C
      subroutine rab(b,s,natoms,r,v,accel,v_prev,a_prev,deltat,nf)
C
      do i = 1,natoms
         do j = 1,3
            sa(j,i) = 0
            sv(j,i) = 0
         end do
      end do
      do i = 1,s
         do j = 1,natoms
            do k = 1,3
               sa(k,j) = sa(k,j) + b(i) * a_prev(k,j,s - i + 1)
               sv(k,j) = sv(k,j) + b(i) * v_prev(k,j,s - i + 1)
            end do
         end do
      end do
      ecin = 0.d0
      do i = 1,natoms
         v2 = 0.d0
         do j = 1,3
            v(j,i) = v(j,i) + deltat * sa(j,i)
            r(j,i) = r(j,i) + deltat * sv(j,i)
            v2 = v2 + v(j,i) * v(j,i)
         end do
      end do
      return
```

```
      subroutine b_coefficient(s, j, b)
      p(1) = 1
      n = 1
      ni = 2
      do i = 0,s - 1
         if (i /= j) then
            pi(1) = i
            pi(2) = 1
            call multi_poly(p, pi, pr, n, ni, nr)
            n = nr
            p = pr
         endif
      end do
      call integrate_poly(p, n, integral)
      call factorial(j, factorialj)
      call factorial(s-j-1,factorialsj1)
      b = ((-1) ** j) / (factorialj*factorialsj1) * integral
      return
C
      subroutine multi_poly(p1, p2, pr, n1, n2, nr)
      nr = 0
      do i = 1,n1
         do j = 1,i-1
            p(j) = 0
         end do
         do j = 1,n2
            p(i + j - 1) = p1(i) * p2(j)
         end do
         call sum_poly(pr, p, pr, nr, i + n2 - 1,nr)
      end do
      return
C
      subroutine sum_poly(p1, p2, pr, n1, n2, nr)
      m = min(n1, n2)
      do i = 1,m
        pr(i) = p1(i) + p2(i)
      end do
      if (n1 > n2) then
        do i = m + 1, n1
           pr(i) = p1(i)
        end do
      else if (n1 < n2) then
        do i = m + 1, n2
           pr(i) = p2(i)
        end do
      endif
      nr = max(n1, n2)
      return
C
      subroutine integrate_poly(p, n, r)
      do i = 1,n
         r = r + p(i) / i
      end do
      return
```

Figure 1. Core of the RAB integrator reported in the present work, written as a pseudo-code (left). Auxiliary (pseudo-)codes used in the calculations with the RAB integrator (right). The routine to obtain the factorial of a number has been omitted due to its simplicity.

References

[1] Verlet L. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Physical review. 1967;159:98.

[2] Verlet L. Computer" experiments" on classical fluids. ii. equilibrium correlation functions. Physical Review. 1968;165:201.

[3] Beeman D. Some multistep methods for use in molecular dynamics calculations. Journal of computational physics. 1976;20:130–139.

[4] Rahman A. Correlations in the motion of atoms in liquid argon. Physical review. 1964;136:A405.

[5] Gear CW. Numerical initial value problems in ordinary differential equations. Prentice Hall PTR; 1971.

[6] Howe RM. A new family of real-time redictor-corrector integration algorithms. Simulation. 1991;57:177–186.

[7] Durran DR. The third-order adams-bashforth method: An attractive alternative to leapfrog time differencing. Monthly weather review. 1991;119:702–720.

[8] Rodriguez JI, Jiménez JM, Funes FJ, et al. Recursive and residual algorithms for the efficient numerical integration of multi-body systems. Multibody System Dynamics. 2004;11:295–320.

[9] Press WH, Teukolsky SA, Flannery BP, et al. Numerical recipes in fortran 77: volume 1, volume 1 of fortran numerical recipes: the art of scientific computing. Cambridge university press; 1992.

[10] Marsden JE, Shadwick WF, Patrick GW, et al. Integration algorithms and classical mechanics. 10; American Mathematical Soc.; 1996.

[11] Lennard-Jones JE. Cohesion. Proceedings of the Physical Society. 1931;43:461.

[12] McDonald I, Singer K. Examination of the adequacy of the 12–6 potential for liquid argon by means of monte carlo calculations. The Journal of Chemical Physics. 1969;50:2308–2315.

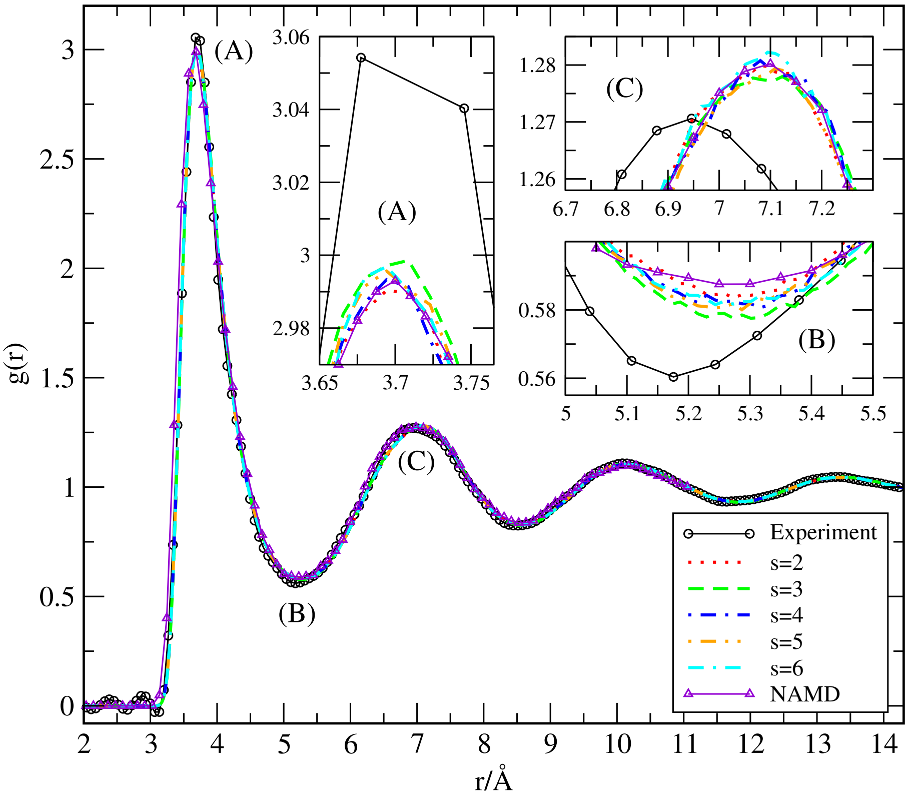

Figure 2. Radial distribution function $g(r)$ of liquid argon at 86.5 K. Comparison of experiment at 85 K[18] and recursive Adams-Bashforth with variable order ($s = 2, ..., 6$). Points 'A', 'B' and 'C' shown with more details in the corresponding insets.

[13] Pastor RW, Brooks BR, Szabo A. An analysis of the accuracy of langevin and molecular dynamics algorithms. Molecular Physics. 1988;65:1409–1419.

[14] Kast SM, Brickmann J. Constant temperature molecular dynamics simulations by means of a stochastic collision model. ii. the harmonic oscillator. The Journal of chemical physics. 1996;104:3732–3741.

[15] Grønbech-Jensen N, Farago O. A simple and effective verlet-type algorithm for simulating langevin dynamics. Molecular Physics. 2013;111:983–991.

[16] Streett W, Staveley L. Experimental study of the equation of state of liquid argon. The Journal of Chemical Physics. 1969;50:2302–2307.

[17] Gosman A, Hust JG, Mc Carty RD. Thermodynamic properties of argon from the triple point to 300 k at pressures to 1000 atmospheres. 1969;.

[18] Yarnell J, Katz M, Wenzel RG, et al. Structure factor and radial distribution function for liquid argon at 85 k. Physical Review A. 1973;7:2130.

[19] Naghizadeh J, Rice SA. Kinetic theory of dense fluids. x. measurement and interpretation of self-diffusion in liquid ar, kr, xe, and ch4. The Journal of Chemical Physics. 1962;36:2710–2720.

[20] Barker J, Fisher R, Watts R. Liquid argon: Monte carlo and molecular dynamics calculations. Molecular Physics. 1971;21:657–673.

[21] Phillips JC, Braun R, Wang W, et al. Scalable molecular dynamics with namd. Journal of computational chemistry. 2005;26:1781–1802.

[22] Huang J, MacKerell Jr AD. Charmm36 all-atom additive protein force field: Validation based on comparison to nmr data. Journal of computational chemistry. 2013;34:2135–2145.

[23] Martí J, Padró J, Guàrdia E. Computer simulation of molecular motions in liquids: Infrared spectra of water and heavy water. Molecular Simulation. 1993;11:321–336.

[24] Van Der Spoel D, Lindahl E, Hess B, et al. Gromacs: fast, flexible, and free. Journal of computational chemistry. 2005;26:1701–1718.

[25] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. Journal of computational physics. 1995;117:1–19.

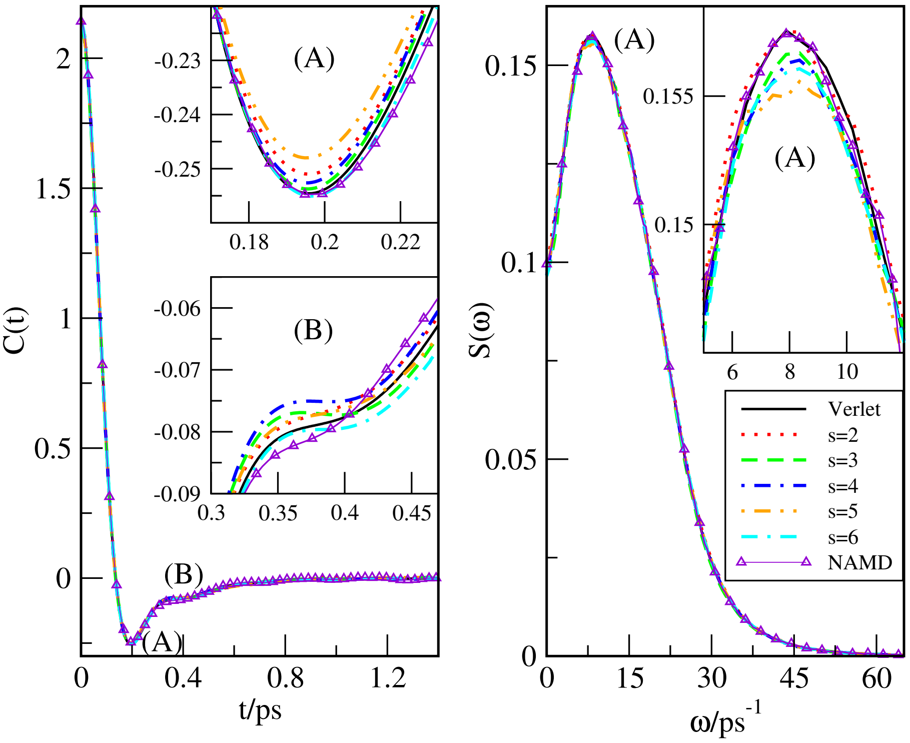

Figure 3. Velocity self-correlation functions and corresponding spectral densities of liquid argon at 86.5 K. Comparison of leapfrog Verlet and recursive Adams-Bashforth of order 2 to 6. Points 'A' and/or 'B' shown with more details in the corresponding insets.

[26] Padró J, Martí J. An interpretation of the low-frequency spectrum of liquid water. The Journal of chemical physics. 2003;118:452–453.

[27] De Santis A, Ercoli A, Rocca D. Comment on "an interpretation of the low-frequency spectrum of liquid water"[j. chem. phys. 118, 452 (2003)]. The Journal of chemical physics. 2004;120:1657–1658.

[28] Padró J, Martí J. Response to "comment on 'an interpretation of the low-frequency spectrum of liquid water'"[j. chem. phys. 118, 452 (2003)]. The Journal of Chemical Physics. 2004;120:1659–1660.

[29] Martí J, Csajka FS. Transition path sampling study of flip-flop transitions in model lipid bilayer membranes. Physical Review E. 2004;69:061918.

[30] Mishin Y, Mehl M, Papaconstantopoulos D, et al. Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations. Physical Review B. 2001;63:224106.

[31] Purja Pun G, Mishin Y. Development of an interatomic potential for the ni-al system. Philosophical Magazine. 2009;89:3245–3267.

[32] Pun GP, Mishin Y. Embedded-atom potential for hcp and fcc cobalt. Physical Review B. 2012; 86:134116.