Parallel-in-time molecular-dynamics simulations

Leonardo Baffico, Stéphane Bernard, Yvon Maday, Gabriel Turinici, Gilles Zérah

Parallel-in-time molecular-dynamics simulations

L. Baffico,[1] S. Bernard,[2] Y. Maday,[1] G. Turinici,[3] and G. Zérah[2,*]
[1]*Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, Boîte Courrier 187, 75252 Paris Cedex 05, France*
[2]*Département de Physique Théorique et Appliquée, Commissariat à l'Energie Atomique, CEA-DAM Ile de France, Boîte Postale 12, 91680 Bruyères le Châtel, France*
[3]*INRIA Rocquencourt, Boîte Postale 105, 78153 Le Chesnay Cedex, France*
(Received 1 February 2002; published 19 November 2002)

While there have been many progress in the field of multiscale simulations in the space domain, in particular, due to efficient parallelization techniques, much less is known in the way to perform similar approaches in the time domain. In this paper we show on two examples that, provided we can describe in a rough but still accurate way the system under consideration, it is indeed possible to parallelize molecular dynamics simulations in time by using the recently introduced *parareal* algorithm. The technique is most useful for *ab initio* simulations.

DOI: 10.1103/PhysRevE.66.057701 PACS number(s): 83.10.Rs, 31.15.Qg

I. MOTIVATION

Whereas microscopic simulations can be performed on larger and larger space scales, in particular, through the use of multiscale techniques and massively parallel computers (for a general survey, see Ref. [1]) very few methods are available to achieve similar results in the time domain. Some schemes have been put forward, for instance following *A.* Voter's ideas, but they are specifically devoted to the sampling of transitions rates assuming that the system obeys rate transition theory. In such situations, attempts to cross the barrier can be run in parallel, and one mainly gather statistics from this many runs. This method is extremely efficient, since it leads to linear scaling with the number of processors, but it addresses quite a specific situation. Since for space, through the use of parallel computers, larger and larger scales have become accessible in microscopic simulations, it could be seducing to transpose the same procedures in the time domain. Naturally, contrary to space, time is sequential and this precludes *a priori* the straightforward implementation of a parallel approach. Here, we try the *parareal* [2] idea which relies on a technique for matching solution segments in parallel. In practice, one uses two time propagators: one approximate, the so-called ''coarse'' propagator and the exact one. As in any parallel implementation of a general problem, there is an unavoidable sequential part which here is the coarse propagation; on the contrary, the (more expensive) exact propagator is run in parallel on portions of the trajectory. More specifically, one starts with a first guess of the trajectory, generated by the coarse propagator for a certain number of time steps. Each point of this first guess trajectory is used as a starting point for the ''fine'' integrator run on one or more time steps. Naturally, these steps can be performed in parallel. Then, one can estimate the error and correct in parallel to generate the next trajectory. From this, one sees that, if the coarse integrator is fast and the convergence toward the exact trajectory is rapid, the method will be efficient. The idea to work on segments of the trajectory, is reminiscent of the one followed in Refs. [3,4], in which they minimize the action, and this variational formulation is the key to their parallel implementation of the ''stochastic difference equation'' approach [5].

II. DESCRIPTION OF THE METHOD

To explain the technique, we first make a sketchy description. Let us denote by $(u_0, \ldots, u_N)$ the successive configurations (position and velocity) of the system we want to describe. If Δt is our time step, and $F_{\Delta t}$ denotes the action of propagating one configuration for one time step, we have naturally for $0 \leqslant n < N$,

$$u_{n+1} = F_{\Delta t}(u_n) \tag{1}$$

with u_0, our initial condition. Propagating for N time steps the initial configuration u_0 is naturally a sequential operation. To introduce parallelism, suppose we are given (the method to obtain these configurations will be described later) a set of initial configurations for each time step denoted by $(u_0^0, \ldots, u_N^0)$, where $u_0^0 = u_0$. We can propagate with $F_{\Delta t}$ for one time step all these configurations in parallel, and generate a new set of configurations, $(\tilde{u}_0^0, \ldots, \tilde{u}_N^0)$. We then have for $0 \leqslant n < N$,

$$\tilde{u}_{n+1}^0 = F_{\Delta t}(u_n^0). \tag{2}$$

If, by accident, we had $u_{n+1}^0 = \tilde{u}_{n+1}^0$ for $0 \leqslant n < N$, our initial configuration would be exactly the trajectory we are looking for, and our problem would be solved. In general this will not be the case and we shall define the error by $\Delta_{n+1}^0 = \tilde{u}_{n+1}^0 - u_{n+1}^0$ and try to generate a new set of configurations (indexed by the superscript 1) $(u_0^1, \ldots, u_N^1)$ such that $\Delta_{n+1}^1 = \tilde{u}_{n+1}^1 - u_{n+1}^1$ is smaller than Δ_{n+1}^0, or at least converge rapidly to zero when one iterates the process (cf. Fig. 1).

To be more specific, we need to define how we construct our first set of configurations, and how we correct it. The most natural way to construct our first set of configurations is to propagate sequentially for N time steps, our configuration

*Corresponding author; email address: gilles.zerah@cea.fr

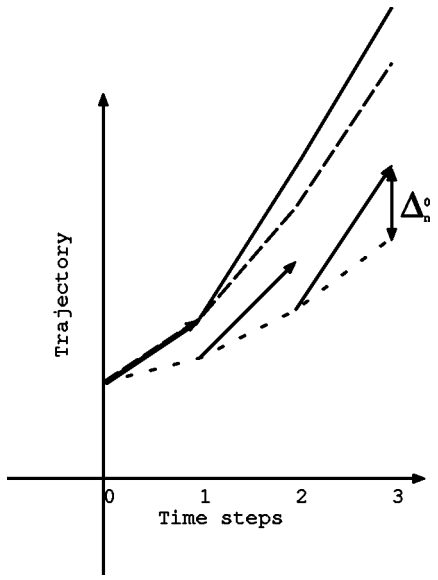

FIG. 1. Schematic presentation of the feedback process. A first trajectory is generated with the coarse force field (dashed line) Then, starting from all points of this trajectory, we advance them for one time step in parallel using the fine force field (solid arrows). The error is measured by Δ_n^0. We then generate the corrected trajectory (long dashed line) using the coarse field now shifted by Δ_n^0. It moves nearer to the exact trajectory (solid line). The first step is exact.

at $t=0$ using another force field, much cheaper to compute than the original one (the coarse force field, as opposed to the fine force field). We will denote the propagator associated with it by G, and note that there exists a considerable freedom of choice for this field, based on physical considerations. Thus our first trial trajectory is defined accordingly,

$$u_{n+1}^0 = G_{\Delta T}(u_n^0),$$

$$u_0^0 = u_0, \tag{3}$$

for $0 \leqslant n \leqslant (N-1)$.

To correct this trajectory, we use a feedback mechanism. If we note that our above definition of Δ_n^0 is equivalent to $\Delta_n^0 = F_{\Delta T}(u_n^0) - G_{\Delta T}(u_n^0)$, our feedback expression is simply

$$u_{n+1}^1 = G_{\Delta T}(u_n^1) + \Delta_{n+1}^0. \tag{4}$$

In other words, we propagate our configurations by the same coarse force field, starting with the same configuration at $t=0$ *corrected* by the error Δ_{n+1}^0. The process can be iterated, and this defines our successive trajectories, denoted by u_n^k where k is the order of iteration, by

$$u_{n+1}^{k+1} = G_{\Delta T}(u_n^{k+1}) + F_{\Delta T}(u_n^k) - G_{\Delta T}(u_n^k),$$

$$u_0^k = u_0. \tag{5}$$

From this formula, one can see how the error in the u_n^k trajectory (the last two terms in the right hand side) is accounted for nonlinearly (in general) from iterations to iterations. Also, this error being dependent only on k, can be computed in parallel, while the first term, dependent on $k+1$, introduce a sequential part in the algorithm. Let us call *gain of the parareal method* the ratio g between the wall clock times, for the computation of the solution by the sequential algorithm, and the parareal method. In practice, there is an optimal value of this gain that depends on the system, its coarse approximation, the number of available processors. Its expression reads

$$g^{-1} = k_{conv}\left(\frac{1}{r} + \frac{1}{N}\right) + \frac{1}{r}, \tag{6}$$

where r stands for the ratio between the computing of the fine and coarse integrators between 0 and ΔT, and k_{conv} being the number of iterations. One can also prove easily by induction that $u_n^n = F_0^n(u_0)$, that is, the algorithm yields exact convergence in at most N iterations ($k_{conv} < N$).

III. A SIMPLE ANALYSIS

Consider for now a simple linear system, in which the propagators are given by multiplication of u by certain operators denoted by F and G. The parareal formula now reads

$$u_{n+1}^{k+1} = G_{\Delta T} u_n^{k+1} + (F_{\Delta T} - G_{\Delta T}) u_n^k.$$

If $F_{\Delta T}$ and $G_{\Delta T}$ commute, the solution is easily found to be

$$u_{n+1}^{k+1} = F_{\Delta T}^{n+1} u_0 - \sum_{p=k+1}^{n+1} \binom{n+1}{p} (F_{\Delta T} - G_{\Delta T})^p G_{\Delta T}^{n+1-p} u_0.$$

Naturally, since $F_{\Delta T}^{n+1} u_0$ is the exact solution at time $(n+1)\Delta T$, and the second term of the right hand side is the measure of the error. Back to the general nonlinear case of Eq. (5), we directly state, without proof, an error estimate. If $\max_{0 \leqslant n \leqslant N} \| F_{\Delta T} - G_{\Delta T} \| \leqslant C \Delta T \varepsilon$, then

$$\max_{0 \leqslant n \leqslant N} \| u_n^k - u(T_n) \| \leqslant C \varepsilon^k e^{\tilde{C} T}. \tag{7}$$

This formula, whose proof is rather technical and will be presented elsewhere, shows the speed of convergence towards the exact solution as a function of the difference of the two propagators.

IV. ONE TOY EXAMPLE OF MOLECULAR DYNAMICS SIMULATIONS

In this section, we consider a simple asymmetric molecule $A-A-B$ composed of three atoms of mass $m_A = 1$ and $m_B = 2$. The bond lengths between atoms are denoted by r_{AA} and r_{AB}, and the angle $A-\hat{A}-B$ of the bonds is denoted as θ. This molecule evolves on the potential surface U given by

$$U(r_{AA}, r_{AB}, \theta) = \mathcal{V}(r_{AA}) + \mathcal{V}(r_{AB}) + f(\theta),$$

where $\mathcal{V}(r) = 4\varepsilon[(\sigma_1/r)^{12} - (\sigma_1/r)^6]$ is the Lennard-Jones potential and $f(\theta) = (\lambda/\sqrt{2\pi}\sigma_2)\exp[-(\theta-\pi)^2/2\sigma_2] + \mu/\sin(\theta/2)$. Here, the coarse and fine propagators differ

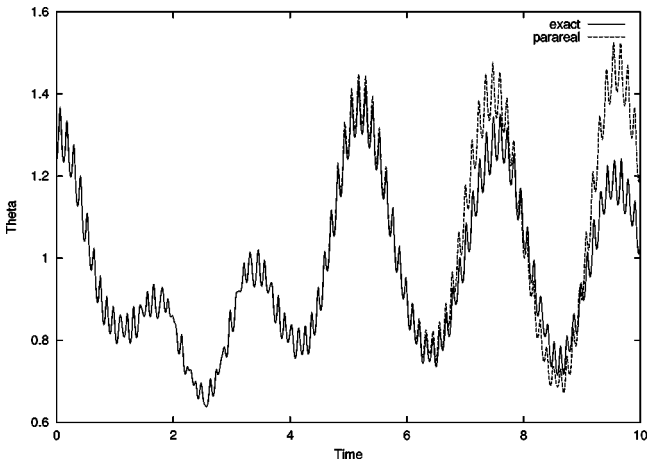

FIG. 2. Behavior of the angle θ for molecule $A-A-B$, solution after two parareal iterations and exact solution. Here $\mu=\sigma_1=\sigma_2=1$, $\varepsilon=\lambda=30$. We have performed 10^4 and 10^6 time steps, for, respectively, the coarse and the fine time steps.

only by the time step used in the discretization of the differential equation. With a coarse propagator corresponding to a time step of $\Delta T=10^{-3}$, we obtain the same accuracy as obtained with a very small time step of order 10^{-6} (required due to the stiffness of this system) after only $k_{conv}=6$ iterations of the parareal algorithm leading to a gain $g\simeq 130$ (see Fig. 2).

Here and in Refs. [2,6], the coarse propagation operator $G_{\Delta T}$ is based on a large time step discretization of Eq. (1), in what follows, we propose a different approach where $G_{\Delta T}$ is rather based on a simpler, physically based, modelization of our system.

V. TWO EXAMPLES OF *AB INITIO* MOLECULAR DYNAMICS SIMULATIONS

Ab initio molecular dynamics simulations are an ideal test bed for such approaches, in the sense that there are many available coarse propagators at our disposal, which are motivated by physical insight. As a first approach, we consider as a coarse propagator a reduced basis set description of our system, and in the plane wave approach used in the *Abinit* package [7], this is tantamount to use a small cutoff.

We consider, in this case, a very small system consisting of four aluminum atoms in the liquid state, enclosed in a cubic box at normal density, and using periodic boundary conditions. We used the simulation program Abinit, and the norm conserving pseudopotential of the Martin Trouilllier type one can find in the site's database, built with the Fritz Haber Institut pseudopotential package. The potential is somewhat hard, and we need a cutoff of 26 Ha to converge forces accurately. For the small cutoff, we took 1 Ha, which yields 16 plane waves, and this is nearly the minimal value we can consider for the 8 bands included in the simulation, since we use a broadening of 4×10^{-2} Ha and one k point. To make the correspondence with the preceding section, u represents collectively the set of all coordinates and velocities of all particles. As integrator, we used the velocity Verlet

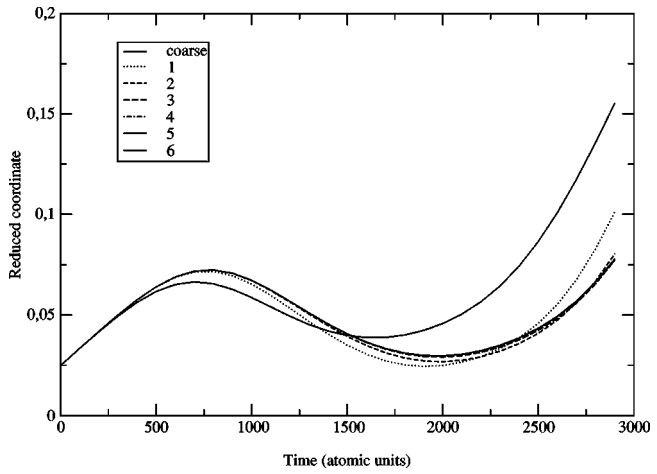

FIG. 3. Convergence of the trajectory of the first coordinate of the first particle. The labels denote the value of the index k. We present here the 30 first time steps.

algorithm, which allows to propagate the positions and velocities according to

$$x_{n+1}=x_n+v_n dt+F(x_n)\frac{dt^2}{2}, \tag{8}$$

$$v_{n+1}=v_n+[F(x_n)+F(x_{n+1})]\frac{dt}{2}, \tag{9}$$

where x and v denotes the set of all coordinates and velocities of all particles respectively.

In Fig. 3, we represent the first coordinate of one of the four particles, for $N=30$ time steps (our time step is $\Delta T=100$ a.u.) for six parareal iteration steps. The first curve, labeled "coarse," is the initial trajectory, and it deviates very rapidly from the correct one, but already the first iterate labeled "1," is very good, while it is impossible to distinguish the trajectories beyond $k=4$, on the scale of the figure. Computations with 16 plane waves take really a negligible

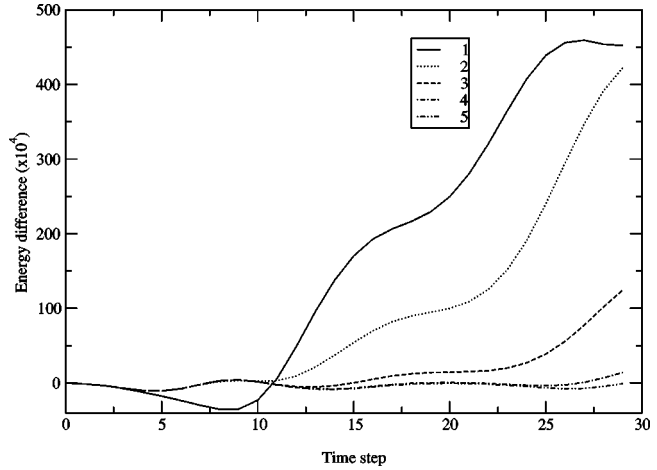

FIG. 4. Variation of the total energy (kinetic+potential) corresponding to u_n^k for the five first values of k. For $k=5$, energy is conserved. We again present here the first 30 time steps.

amount of time, and one could argue that the gain is 7.5 $=30/4$. However, as regarding molecular dynamics, a more pertinent and in our case more stringent criterion is energy conservation. In order to use it as a convergence test, we show, in Fig. 4, the difference between the total energy corresponding to u_n^k and to u_0^0 as a function of time for different parareal iterations. From this figure, it appears that after five parareal steps, the energy fluctuation is down to its residual value due to the finite value of the time step in the Verlet algorithm. Using a small cutoff approximation to the ''exact'' dynamics is in no way the only option we are left with, and for a long time physicists have searched for fast and accurate approximations of *ab initio* force fields. In the case of aluminum, Ercolessi and Adams [8] have published a force field that derives from a potential of the ''glue'' type (very similar to the embedded atom model potentials of Daw and Baskes [9]). The general form of this potential is

$$U=\frac{1}{2}\sum_{i\neq j}\phi(|r_i-r_j|)+\sum_j F(\rho_j),$$

$$\rho_j=\sum_{i\neq j}\rho(|r_i-r_j|), \tag{10}$$

where the sum runs on the atoms coordinates [10]. We obtain, in this case, very similar results: for a 32 particle system, in $N=40$ time steps, the convergence is reached within four iterations. Indeed, one could rightfully argue that we only get what we give, since the Ercolessi-Adams potential is fitted on *ab initio* calculations, but the question of how much we loose by using a glue potential in a simulation has been open for a long time, and the present simulation can also be viewed as a way to take advantage of both approaches. To perform longer simulations, we merely joined trajectories together, and tested the method for longer times. We performed a 1000 time steps simulation this way, and did not observe an energy trend. This is all the more surprising that our algorithm is not time reversible, and this point would require more extensive analysis to be definitevely assesed.

Naturally, very many other combinations of coarse and fine integrators can be considered (tight binding and *ab initio*, simple and complex classical potentials $\cdots$) and we are limited only by our imagination to do so.

VI. CONCLUSION

As a conclusion, we think that the way we introduce the parareal algorithm in materials science simulations can be very useful in the field of multiscale modeling, and open the way to new approaches in the time domain. Nevertheless, some points remain to investigate before having at our disposition a completely efficient method (in particular use of previous time step solution and clarification of symplectic properties). One should also note that situations where we know of one expensive and accurate and one cheap and less accurate description of a physical system are quite common, and extend beyond the limits of material science simulations.

ACKNOWLEDGMENTS

Part of this work was performed during the CEMRACS 2001 meeting held at Luminy, France, and we would like to thank their organizers, particularly F. Nataf, who made this work possible. G.Z. would like to thank F. Jollet, Y. P. Pellegrini, C. Guet, and X. Gonze for useful discussions and suggestions, and M. Mareschal and the participants of the CECAM workshop ''Stress Induced Phase Transformations in Solids'' for the very lively discussions about force field approximations.

[1] J. Broughton, F. Abraham, N. Berstein, and E. Kaxiras, Phys. Rev. B **60**, 2391 (1999).
[2] J. Lions, Y. Maday, and G. Turinici, C. R. Acad. Sci., Ser. I: Math. **232**, 661 (2001).
[3] R. Olender and R. Elber, J. Chem. Phys. **105**, 9299 (1996).
[4] R. Elber, J. Meller, and R. Olender, J. Phys. Chem. B **103**, 899 (1999).
[5] V. Zaloj and R. Elber, Comput. Phys. Commun. **128**, 118 (2000).
[6] G. Bal and Y. Maday, in *Recent Developments in Domain Decomposition Methods*, edited by Luca F. Pavarino and Andrea Toselli, Lecture Notes in Computational Science and Engineering Vol. 23 (Springer-Verlag, Berlin, 2002).
[7] Abinit is a joint project of Université Catholique de Louvain, Corning glass Inc, and other institutions including the Commissariat à l'énergie atomique, http://www.abinit.org
[8] F. Ercolessi and J.B. Adams, Europhys. Lett. **26**, 583 (1994).
[9] M.S. Daw and M.I. Baskes, Phys. Rev. Lett. **50**, 1285 (1983).
[10] F. Ercolessi, http://www.fisica.uniud.it/~ercolessi/potentials/AI/