TBD

Varadarajan Rengaraj,* Michael Lass,† and Christian Plessl‡
Department of Computer science, Paderborn University, Warburger Str. 100, D-33098 Paderborn, Germany

Thomas D. Kühne§
Department of Chemistry, Paderborn University, Warburger Str. 100, D-33098 Paderborn, Germany
(Dated: August 30, 2018)

PACS numbers: 31.15.-p, 31.15.Ew, 71.15.-m, 71.15.Pd

I. INTRODUCTION

Molecular dynamics is a standard technique to study the movement of atoms in a substance over time. It involves computing the forces on all atoms for every time step as a product of the bonded and non- bonded interactions. This is done by numerically solving the Newton's law of motions and update the parameters such as velocity and position of each atom. Computing the forces from non-bonded interactions is computationally expensive and our conventional multicore processors falls behind on the computational requirements. There has been numerous efforts going in this area to accelerate the MD simulations especially the ones based on GPU and FPGA. Microchips sizes of FPGA and GPU, are on a constant decline to accommodate more transistors but it also makes the transistors susceptible to both temporary and permanent failures. These hardware faults occasionally propagate to the software and considering this aspect, there is a renewed interest in approximate computing that can be applied in the software to give us the outputs that does not diverge too much from the ideal outputs. Approximate computing also ensures that the portion of investment needed in detecting the hardware faults, avoidance and recovery is avoided. The research goal of approximate computing is to explore techniques to gain more efficiency by relaxing the exactness of calculated outputs compared to the ideal outputs. In this paper, we describe one such technique that relaxes the exactness of the output and we explore to what extent it diverges from the ideal output.

II. METHODOLOGY

To demonstrate approximate computing, we introduce a computational error, a statistical noise to the forces computed on the atom when running the MD simulation. In this section, we describe in detail on how we introduce the error, a process we mimic in our standard MD system instead of running the MD simulation in the actual FPGA or GPU hardware. We classify the computational errors into two types 1. Fixed point error 2. Floating point error. Fixed point error is described by the following equation.

$$\begin{pmatrix} \mathbf{F}_I^{L(x)} \\ \mathbf{F}_I^{L(y)} \\ \mathbf{F}_I^{L(z)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_I^{x} \\ \mathbf{F}_I^{y} \\ \mathbf{F}_I^{z} \end{pmatrix} + \begin{pmatrix} c_1.10^{-\beta} \\ c_2.10^{-\beta} \\ c_3.10^{-\beta} \end{pmatrix} \tag{1}$$

where c1, c2 and c3 is a random value in the range [-0.5, 0.5]. The values of β that were used in our test simulation runs are discussed in more detail in the computational details section. Floating point error is described by the following equations.

$$\begin{pmatrix} \mathbf{F}_I^{L(x)} \\ \mathbf{F}_I^{L(y)} \\ \mathbf{F}_I^{L(z)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_I^{x}.10^{-\alpha 1} \\ \mathbf{F}_I^{y}.10^{-\alpha 2} \\ \mathbf{F}_I^{z}.10^{-\alpha 3} \end{pmatrix} + \begin{pmatrix} c_1.10^{-(\alpha 1+\beta)} \\ c_2.10^{-(\alpha 2+\beta)} \\ c_3.10^{-(\alpha 3+\beta)} \end{pmatrix} \tag{2}$$

where c1, c2 and c3 is a random value in the range [-0.5, 0.5]. The values of β that are used in our test simulation runs are discussed in more detail in the computational details section.

We use Langevin dynamics, a popular molecular dynamics technique for the purpose to demonstrate that the computational errors introduced by the methods described above can be effectively compensated by an existing framework. For the langevin dynamics molecular simulation run on a FPGA or GPU based accelerators, we assume at this point a computational error Ξ_I^N is added to the force computed and the force we get at the output is not the exact force $\mathbf{F}_I$ but an approximation

$$\mathbf{F}_I^{FPGA} = \mathbf{F}_I + \Xi_I^N \tag{3}$$

and therefore there is no guarantee that correct Boltzmann averages are obtained from the solutions of the Eq. (1). In principle, Ξ_I^N is also a white noise obeying

$$\left\langle \Xi_I^N(0)\, \Xi_I^N(t) \right\rangle \cong 6k_B T M \gamma_I^N \delta(t) \tag{4}$$

$$\left\langle \mathbf{F}_I(0)\, \Xi_I^N(t) \right\rangle \cong 0 \tag{5}$$

* rengaraj@campus.uni-paderborn.de
† michael.lass@uni-paderborn.de
‡ christian.plessl@uni-paderborn.de
§ tdkuehne@mail.upb.de

Fortunately in our case, it is still possible to accurately obtain Boltzmann sampling by means of a modified Langevin equation,

$$M_I\ddot{\mathbf{R}}_I = \mathbf{F}_I + \Xi_I^N - \gamma_N M_I\dot{\mathbf{R}}_I \quad (6)$$

which in our case is,

$$M_I\ddot{\mathbf{R}}_I = \mathbf{F}_I^{FPGA} - \gamma_N M_I\dot{\mathbf{R}}_I \quad (7)$$

We further present in this paper the method to effectively compensate for the noise introduced on the force by the computational errors using existing langevin parameters.

III. COMPUTATIONAL DETAILS

To demonstrate our method as described above, we have implemented it in the Frontiers in Simulation Technology(force field implementation) code which is part of the publicly available suite of programs CP2k[**?**]. We configured to run the molecular dynamics(MD) code for the Silicon(Si) atom in Langevin dynamics mode with a time step 1 at a temperature of 3000 K. The Empirical Interatomic Potential (EIP) calculation model for our MD simulations is Bazant potentials. The total number of Si atoms used for the MD simulation is thousand with a cubic cell length of 27.155 Angstrom.

Molecular dynamics simulation configured for the above settings is run in the Langevin mode with langevin section parameter γ set to 1/1000. This simulation run is considered as the reference with which the computational error introduced MD simulations were compared. As described in the Methodology section, computational errors were classified as fixed point and floating point. With necessary changes incorporated to the cp2k software, two different builds for two different errors were made and MD simulations were performed with those builds. Three different variants of fixed point errors were tested For the equation representing fixed point error, the corresponding β values used were 1, 2 and 3. For the equation representing floating point error, the corresponding β values used were 0, 1, 2.

For the procedure described above if we take a γ value of 1/1000, we find that we need to add a correction in order to get the average kinetic energy to a fixed value. The correction is through the parameter shadow γ and this parameter is available through LANGEVIN section of the cp2k software.

The following tables lists down shadow γ values used for fixed point and floating point errors. Units for γ and shadow γ is fs^{-1}.

Table1. Shown are the shadow γ values for fixed point errors.

β	*Shadow* γ
1	0.0004
2	0.000009
3	0.0000009

Table2. Shown are the shadow γ values for floating point errors.

β	*Shadow* γ
0	
1	0.000005
2	0.000005

IV. RESULTS AND DISCUSSION

In Fig 1 and 2 we compare the pair correlation function g(r) calculated with forces to which computational errors are added to the pair correlation function g(r) calculated with forces evaluated with the regular approach. In Fig 1 we show the pair correlation functions g(r) calculated with forces to which fixed point computational errors are added. In Fig 2 we show the pair correlation functions g(r) calculated with forces to which floating point computational errors are added. We see from both the Figures 1 and 2, that the use of forces with computational errors added does not degrade the quality of the simulation.

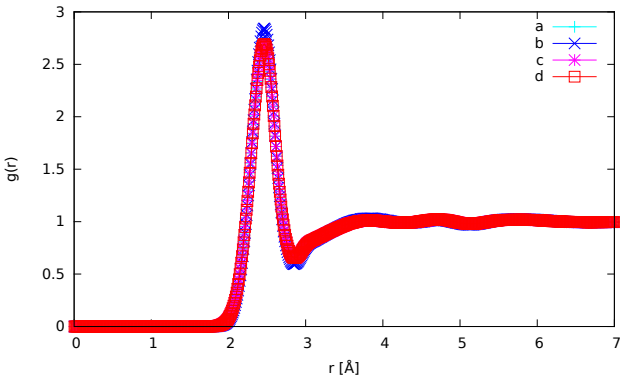

Figure 1. Pair correlation function for (a) liquid silicon (3000 K) and (b) liquid silicon with noisy forces introduced by different range of fixed point errors.

In Fig 3, the ionic kinetic energy distribution calculated with forces to which a computational error is added is compared with the exact Maxwell distribution line and it is seen that not only the average energy is correct but also its fluctuations follow the Maxwell distribution.

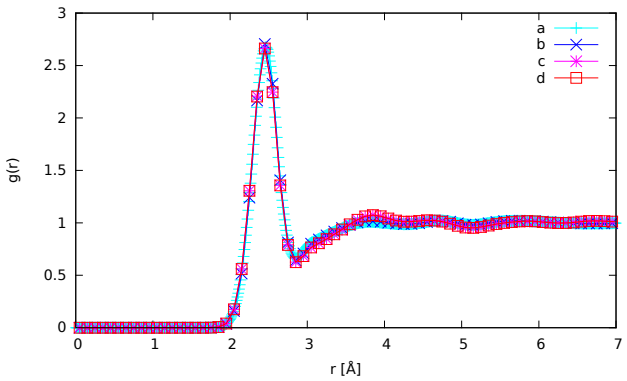

Figure 2. Pair correlation function for (a) liquid silicon (3000 K) and (b) liquid silicon with noisy forces introduced by different range of floating point errors.

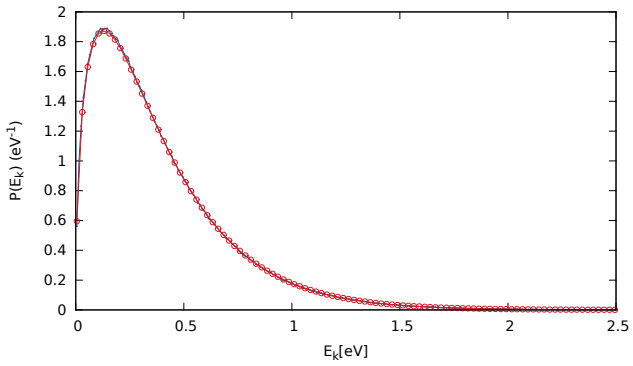

Figure 3. Statistical properties in a 1000 atoms liquid Si simulation at 3000 K. (a) The ionic kinetic energy distributions (line) is compared with the exact Maxwell distribution (circles)

V. SUMMARY

ACKNOWLEDGMENTS

The authors would like to thank the Gauss Center for Supercomputing (GCS) for providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS share of the supercomputer JUQUEEN at the Jülich Supercomputing Centre (JSC). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 716142).

[1] CP2K Open Source Molecular Dynamics, `http://cp2k.berlios.de`
[2] EIP Bazant potential, *tofill*
[3] Stephen C. Harvey, Robert K.-Z. TAN, Thomas Cheatham. *The Flying Ice Cube: Velocity Rescaling in Molecular Dynamics Leads to Violation of Energy Equipartition*
[4] Gerald Knizia, Wenbin Li, Sven Simon, Hans-Joachim Werner. *Determining the Numerical Stability of Quantum Chemistry Algorithms*