Computer Physics Communications 184 (2013) 1191–1202

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Parallel Brownian dynamics simulations with the message-passing and PGAS programming models

C. Teijeiro [a,*], G. Sutmann [b,c], G.L. Taboada [a], J. Touriño [a]

[a] Computer Architecture Group, Department of Electronics and Systems, University of A Coruña, 15071 A Coruña, Spain
[b] Jülich Supercomputing Centre, Institute for Advanced Simulation, Forschungszentrum Jülich, D-52425 Jülich, Germany
[c] ICAMS, Ruhr-University Bochum, D-44801 Bochum, Germany

ARTICLE INFO

Article history:
Received 16 October 2012
Received in revised form
11 December 2012
Accepted 14 December 2012
Available online 19 December 2012

Keywords:
Brownian dynamics
Parallel simulation
MPI
PGAS
UPC

ABSTRACT

The simulation of particle dynamics is among the most important mechanisms to study the behavior of molecules in a medium under specific conditions of temperature and density. Several models can be used to compute efficiently the forces that act on each particle, and also the interactions between them. This work presents the design and implementation of a parallel simulation code for the Brownian motion of particles in a fluid. Two different parallelization approaches have been followed: (1) using traditional distributed memory message-passing programming with MPI, and (2) using the Partitioned Global Address Space (PGAS) programming model, oriented towards hybrid shared/distributed memory systems, with the Unified Parallel C (UPC) language. Different techniques for domain decomposition and work distribution are analyzed in terms of efficiency and programmability, in order to select the most suitable strategy. Performance results on a supercomputer using up to 2048 cores are also presented for both MPI and UPC codes.

1. Introduction

Dynamical particle simulations aim at exploring the phase or configuration spaces of the underlying physical system in order to gather statistics for the calculation of expectation values of observables. Depending on the level of resolution, either *ab initio* methods, force field or effective medium descriptions are used to propagate particles according to their equations of motion. Interactions between solvated particles as well as particle–solvent and particle–wall interactions are modeled according to their physical characteristics, their statistical properties and information from experimental data. Brownian dynamics is a class of simulation that takes into account the systematic interactions between particles, as well as the interaction with a surrounding medium described by their statistical and transport properties, which are often represented by diffusion tensors derived from a velocity field description of the solvent.

It is assumed that solvated particles are very large compared to fluid particles and that individual interactions can be reduced to statistical fluctuations, induced by thermal noise. This allows one to consider a time scale separation, i.e., during a time step in which a so-called Brownian or solute particle (e.g., a macromolecule, polymer or colloid) is propagated, individual solvent particles would perform a large number of micro steps, and therefore the fluid–solute interaction is considered as an average action. Thus, although individual interactions between fluid and solute are not considered, the simulation takes into account the collective properties of the fluid molecules using a mobility tensor, which contains information about the velocity field in the system.

Technically, a finite difference scheme is applied to calculate the trajectory for each particle as a succession of short displacements Δt in time. In a system, containing N particles, the trajectory $\{\mathbf{r}_i(t); t \in [0, t_{\max}]\}$ of particle i is calculated as a succession of small and fixed time step increments Δt. The time step is selected to be (1) large enough, i.e., $\Delta t \gg m_i/6\pi\eta a_i$, with η the solvent viscosity, a_i the radius and m_i the mass of the solute particle i, that the interaction between individual fluid particles and the solutes can be considered as averaged and can be coupled to the solutes via the diffusion tensor, and (2) small enough that the forces and gradients of the diffusion tensor can be considered constant within Δt. According to these conditions, the simulation can be performed by calculating the forces that act on every particle in a time step, determining new positions for all particles and continuing this process in the following time step.

Brownian dynamics simulations are nowadays used to perform many studies in different areas of physics and biology [1], and

* Corresponding author.
E-mail addresses: cteijeiro@udc.es (C. Teijeiro), g.sutmann@fz-juelich.de (G. Sutmann), taboada@udc.es (G.L. Taboada), juan@udc.es (J. Touriño).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.12.015

there are several software tools that help implementing these simulations, such as BrownDye [2] and the BROWNFLEX program included in the SIMUFLEX suite [3]. Some relevant work has also been published on parallel implementations of these simulations on GPUs [4], also including a simulation suite called BD_BOX [5]. However, there is still little information on parallelization methods for these simulations, especially about their performance and scalability on high-performance computing (HPC) systems. This work overcomes these limitations by providing an accurate description of the parallelization of a Brownian dynamics simulation for a set of solvated particles in a certain period of time, in order to build a suitable implementation for HPC systems. The parallel algorithm has been developed using MPI and Unified Parallel C (UPC), which illustrate two programming models, a message-passing model and a Partitioned Global Address Space (PGAS) model, respectively, to obtain high efficiency and scalability. The most relevant information about the parallelization of the different parts of the simulation is presented, and its performance is analyzed on a supercomputer in order to explain the behavior of the code for different test cases.

The rest of this work is organized as follows. First, Section 2 presents the formal explanation of the simulation. Section 3 contains a detailed computational description of the simulation code. In Section 4, details about the parallelization strategies are outlined, and Section 5 presents performance results of the parallel codes using different workloads and computational resources. Finally, Section 6 extracts the main conclusions from this work, and the Appendix provides additional theoretical information about the simulation.

2. Theoretical background of Brownian dynamics

The equation of motion, governing Brownian dynamics for solvated molecules in a fluid, has been stated by Ermak and McCammon [6] (based on the Fokker–Planck and Langevin descriptions):

$$\mathbf{r}_i(t+\Delta t) = \mathbf{r}_i(t) + \sum_{j=1}^{N} \frac{\partial \mathbf{D}_{ij}(t)}{\partial \mathbf{r}_j}\Delta t + \sum_{j=1}^{N} \frac{1}{k_B T}\mathbf{D}_{ij}(t)\mathbf{F}_j(t)\Delta t + \mathbf{R}_i(t+\Delta t). \tag{1}$$

This one-step propagation scheme takes into account the coupling of the particles to the flow field via the diffusion tensor $\mathbf{D} \in \mathbb{R}^{3N\times 3N}$ and the systematic forces $\mathbf{F}$, acting onto the particles with the global property $\sum_j \mathbf{F}_j = 0$. The vector $\mathbf{R} \in \mathbb{R}^{3N}$ contains correlated Gaussian random numbers with zero mean, which are constructed according to the fluctuation–dissipation theorem; i.e.,

$$\langle R_{i,\alpha}\rangle = 0, \quad \alpha = x, y, z \tag{2}$$

$$\langle \mathbf{R}_i(t+\Delta t)\mathbf{R}_j^T(t+\Delta t)\rangle = 2\mathbf{D}_{ij}(t)\Delta t, \tag{3}$$

with $\mathbf{R}_i \in \mathbb{R}^3$ and $\mathbf{D}_{ij} \in \mathbb{R}^{3\times 3}$ being subvectors and block matrices corresponding to particle i and particle pairs i, j, respectively. k_BT is the thermal energy of the system, where T is the temperature and k_B the Boltzmann constant. Depending on the approximation for the diffusion tensor, the partial derivative on the right-hand side of Eq. (1) might drop out. This is the case for example of the Oseen tensor and the Rotne–Prager tensor [7,8]. The latter takes into account the finite size of solute particles, and its regularized version is considered in the present work [7], thus fulfilling the requirement of positive definiteness also for inter-particle distances $r_{ij} < 2a$, where $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$. Here, we give the expression for the so-called minimum image convention. The expression for periodic boundary conditions applied in the code is outlined in the Appendix. The minimum image formulation is given by

$$\mathbf{D}_{ii} = \frac{k_B T}{6\pi\eta a}\mathbf{I} \tag{4a}$$

$$\mathbf{D}_{ij} = \begin{cases} \dfrac{k_B T}{8\pi\eta r_{ij}}\left[\left(\mathbf{I} + \hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}^T\right) + \left(\dfrac{2}{3}\dfrac{a^2}{r_{ij}^2}\mathbf{I} - 3\hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}^T\right)\right] & : r_{ij} > 2a \\ \dfrac{k_B T}{6\pi\eta a}\left[\left(1 - \dfrac{9}{32}\dfrac{r_{ij}}{a}\right)\mathbf{I} + \dfrac{3}{32}\dfrac{r_{ij}}{a}\hat{\mathbf{r}}_{ij}\hat{\mathbf{r}}_{ij}^T\right] & : r_{ij} \le 2a, \end{cases} \tag{4b}$$

where $\hat{\mathbf{r}}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/r_{ij}$. Applying this form of the diffusion tensor, the displacement vector of the Brownian particles, $\Delta\mathbf{r} = \mathbf{r}(t+\Delta t) - \mathbf{r}(t)$, can be rewritten in a more simple way:

$$\Delta\mathbf{r} = \frac{1}{kT}\mathbf{D}\mathbf{F}\Delta t + \sqrt{2\Delta t}\,\mathbf{Z}\boldsymbol{\xi}, \tag{5}$$

where $\boldsymbol{\xi}$ is a vector of independent Gaussian random numbers. According to Eq. (3), the relation $\mathbf{R} = \sqrt{2\Delta t}\mathbf{Z}\boldsymbol{\xi}$ holds with $\mathbf{D} = \mathbf{Z}\mathbf{Z}^T$, which relates the stochastic process to the diffusion matrix. Therefore, $\mathbf{Z}$ may be calculated via a Cholesky decomposition or via the square root of $\mathbf{D}$. Both approaches are very CPU-time consuming, with a computational complexity of $\mathcal{O}(N^3)$, and they impose a large computational load. Therefore the development of faster and more efficient and scalable methods with smaller complexity is an important task, in order to overcome the limitations in the system size. Such an approach was introduced by Fixman [9], who applied an expansion of the random displacement vector $\mathbf{R}$ in terms of Chebyshev polynomials, approximating its values without constructing $\mathbf{Z}$ explicitly and reducing the computational complexity to $\mathcal{O}(N^{2.25})$.

Both methods for the construction of correlated random variates, based on the Cholesky decomposition and the Chebyshev approximation, will be considered in the present work.

3. Implementation of the simulation code

The Brownian dynamics simulation has been initially implemented in sequential C code. The system under study consists of a cubic box where periodic boundary conditions are applied, and the propagation of Brownian particles is performed by evaluating Eq. (5). The systematic interactions between particles are modeled by a Lennard-Jones-type potential, from which the forces are obtained via the negative gradient:

$$V(r_{ij}) = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \tag{6a}$$

$$\mathbf{F}_{ij} = -\nabla_{\mathbf{r}_{ij}}V(r_{ij}) = 24\varepsilon\left[2\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right]\frac{\hat{\mathbf{r}}_{ij}}{r_{ij}^2}, \tag{6b}$$

where σ is the diameter of the particles and ϵ is the depth of the potential minimum. This potential has a short-range character and practically interactions between particles are neglected for mutual distances $r_{ij} > R_c$, where R_c is the radius of a so-called cutoff sphere, which is chosen as $R_c = 2.5\,\sigma$. The distance r_{ij} is chosen according to the minimum image convention; i.e., the shortest distance between particle i (located in the central simulation box) and particle j or one of its periodic images is taken into account (see Fig. 1). In the code, the diffusion tensor $\mathbf{D}$ is calculated in periodic images, which implies a summation of particle pair contributions over all periodic images. The expression, which consists of a