Computer Physics Communications 185 (2014) 836–840

Contents lists available at ScienceDirect

Computer Physics Communications

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

Data structure for on-lattice cluster–cluster aggregation model performance optimization

Hailing Xiong [a,b,*], Hang Li [c], Weiping Chen [d], Laosheng Wu [e]

[a] College of Computer and Information Science, Southwest University, Chongqing 400715, China
[b] Key Laboratory of Eco-environments in Three Gorges Reservoir Region (Ministry of Education), Southwest University, Chongqing 400715, China
[c] College of Resources and Environment, Southwest University, Chongqing 400715, China
[d] State Key Lab of Urban and Regional Ecology, Chinese Academy of Sciences, Beijing 100085, China
[e] Department of Environmental Sciences, University of California, Riverside, CA 92521, USA

ARTICLE INFO

Article history:
Received 4 August 2013
Received in revised form
10 November 2013
Accepted 19 November 2013
Available online 25 November 2013

Keywords:
Data structure
On-lattice simulation
Metropolis sampling
Cluster–cluster aggregation
Computational complexity

ABSTRACT

A compounded data structure is developed to optimize the simulation of colloidal aggregation using the on-lattice Cluster–Cluster Aggregation (CCA) model. Brownian motion, collision detection and aggregation as the basic operations in the CCA simulation are illustrated and evaluated based on the compounded data structure, respectively. The critical improvement of our algorithm is in distinguishing any selected clusters consisting particles and ascertaining their neighboring positions efficiently in simulation, which was traditionally performed by the exhaustive search in the whole system. Analytical results show that the new algorithm achieves linear computational complexity in each of the main operations, which is very appealing in performance optimization in using on-lattice CCA simulations.

1. Introduction

Colloidal aggregation is an important environmental phenomenon as pollutants often are present as colloids in water and aerosols in the atmosphere [1–3]. The process starts with a suspension of finely divided particulates at a low initial volume fraction in which stable colloids must overcome the repulsive energy barrier between two particles to coalesce and grow in size [3]. The Cluster–Cluster Aggregation (CCA) model assumes that initially a collection of equal sized individual spherical particles are randomly dispersed in a box at low concentrations and then these particles are allowed to self-diffuse in Brownian motion by random walk, collide, and form clusters [4,5]. The CCA model is developed to simulate the particle growing mechanism [6–9]. It is widely adopted to describe the aggregation of colloids and aerosols with good agreements with experimental results [10–14]. The simulated outcomes provide insights in understanding the aggregation process and determine important features such as fractal dimension and structure factor of clusters, shapes of cluster mass distribution, and aggregation kinetics [12].

The CCA models perform well in small-scale simply simulations. The computation processes however challenge the capabilities of current day digital computers requiring large memories and high speed processors. The computational complexity is a significant handicap to simulate colloidal systems comparable with real-world experimental data [15,16]. Hawick, etc. had publicized a serious of technical reports about the simulation method, one technical report similar to this research discussed space partitioning methods for the CCA simulation and demonstrate complexity improvements by taking advantage of information about locations and interaction distances of the microscopic model components [16]. Kusaka, etc. studied CCA simulation in a concentrated suspension, reported on the characteristics of the collision radius of fractal aggregates, and discussed aggregation kinetics based on the value of the estimated collision radius [13]. The CCA model is authentically a recognized and widely used ingenious model to mimic fractal aggregation. However, emphasis has long been focused more on the structure of the aggregates and the kinetics of aggregation processes than on the algorithm efficiency.

The conventional algorithm of a CCA model is depicted in Fig. 1 [17] in which the Brownian movement, collision detection and aggregation are the main steps. The simulation starts with N non-overlapping identical particles distributed randomly in a cubic

* Corresponding author at: College of Computer and Information Science, Southwest University, Chongqing 400715, China. Tel.: +86 23 68367731; fax: +86 23 68252051.
E-mail address: xionghl@swu.edu.cn (H. Xiong).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.11.015

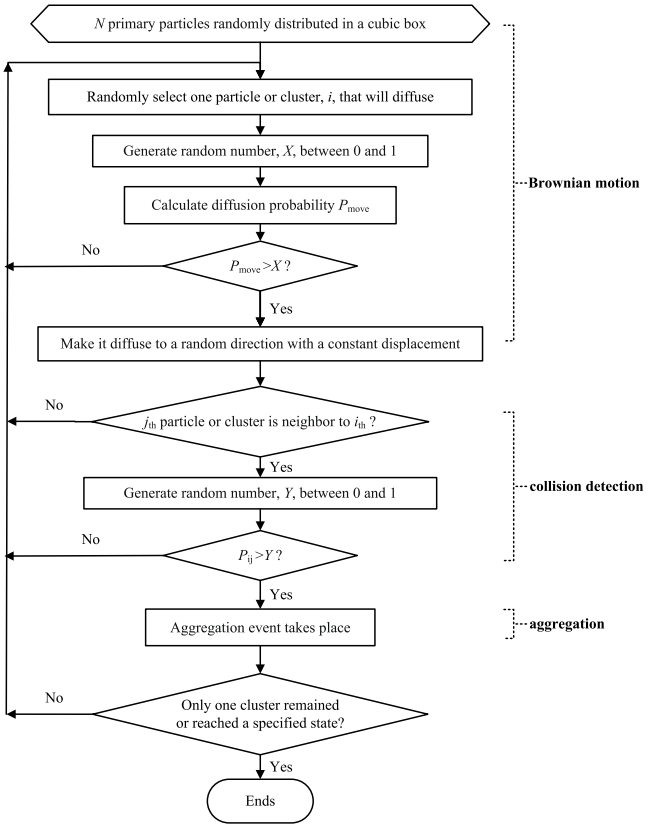

Fig. 1. Algorithmic flow chart of on-lattice CCA simulations.

box with side-lengths of L. In Brownian movement, all particles are in motion and all particles have chance to collide in realistic situations. For the CCA models, Meakin separates and serializes the Brownian movements by adopting the Metropolis sampling method [4]. It lies on the basic premise that in simulations only one randomly selected particle (or cluster) moves at a time and with the distance identical to diameter of the primary particle [6].

To realize the CCA simulation, a three dimensional array, $CUBE[L][L][L]$, was used to represent the cubic box. Each particle in the cubic box occupies an element of the three dimensional array and are labeled with a different integer (>0), therefore on-lattice. When particles and/or clusters collide and aggregate, all particles in the resulting cluster have the same label Fig. 2. The progression of Brownian movement and aggregation are realized by updating the labels of the corresponding array elements.

A critical issue of the simulation is how to efficiently distinguish all of the particles in any selected cluster based on the three dimensional array $CUBE[L][L][L]$ when the cluster is to be moved. Similarly, there are difficulties in the process of collision detection to locate all neighboring positions of the cluster. For example, if a cluster labeled with i was selected to undergo Brownian movement i.e. self-diffuse, all elements that are identified by the label i in the three dimensional array $CUBE[L][L][L]$ must be tested. It is a time-consuming process of $O(L^3)$ computational complexity in three dimensional simulations because an exhaustive search of the entire system must be performed upon each simulated Brownian motion. According to the flow chart (Fig. 1) only the selected cluster need to be checked in the collision detection step. It nevertheless remains to be a time-consuming process because the clusters are quite anomalous and the labels of particles in the selected cluster have to be checked for collision detection one by one with those in all other clusters. A three dimensional array $CUBE[L][L][L]$ is not efficient for the on-lattice CCA simulations.

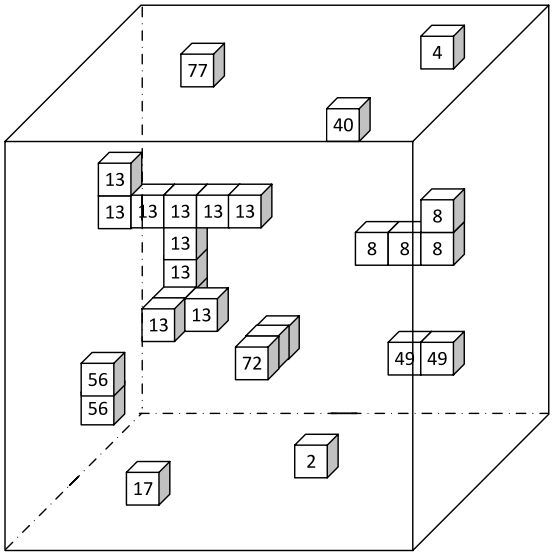

Fig. 2. Single particles and clusters in the cube (in part) are distinguished by the labels marked on the little boxes, which are stored in the corresponding array elements.

We have developed a compounded data structure for the on-lattice CCA simulation that improves the performance efficiency of the simulation algorithm. In the following sections we delineate the simulation model, new data structure, algorithm realization, and efficiency evaluation.

2. Simulation model

The details of the CCA model are given as follows [6]: we take the unit $a = 1$ for a lattice constant. In the model, N particles are randomly distributed in a cubic box with side-length of L. Thus, the particle concentration c becomes:

$$c = N/L^3. \tag{1}$$

Assuming the shape effects can be ignored, the diffusion coefficient D_i of a cluster (consists of i particles) is proportional to the inverse of its gyration radius R_{gi} [18]:

$$D_i = D_1 \times R_1/R_{gi} \tag{2}$$

where D_1 is the diffusion coefficient of the single particle with the gyration radius of R_1.

According to the Metropolis sampling method, one cluster will be selected in each step for a potential movement. The selection is accepted or rejected based on the probability P_{move} [4]:

$$P_{move} = D_i/D_{\max} \tag{3}$$

where D_i is the diffusion coefficient of the selected cluster, D_{max} is the maximum diffusion coefficient for any cluster in the system. For the potential movement, a random number x uniformly distributed over the range [0, 1] is generated and the cluster is moved only if $x < P_{move}$.

The moving direction is also randomly chosen among six directions. If the cluster does not collide with another one, the displacement is performed and the algorithm goes on by choosing another cluster for the next step. If a collision occurs between two clusters (one consists of i particles, and another consists of j particles) they stick together forming a new larger cluster with the sticking probability P_{ij} [4,12]:

$$P_{ij} = P_1 \times (i \times j)^{\sigma} \tag{4}$$