VI International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2019
E. Oñate, M. Bischoff, D.R.J. Owen P. Wriggers & T. Zohdi (Eds)

ASSESSMENT OF NEIGHBOR PARTICLES SEARCHING METHODS FOR DISCRETE ELEMENT METHOD (DEM) BASED SIMULATIONS

LUIS ANGELES[1], CESAR CELIS[2]

[1]Department of Engineering, Pontificia Universidad Católica del Perú (PUCP)
Av. Universitaria 1801, San Miguel, Lima 32, Peru
langelesr@pucp.edu.pe

[2]Mechanical Engineering Section, Pontificia Universidad Católica del Perú (PUCP)
Av. Universitaria 1801, San Miguel, Lima 32, Peru
ccelis@pucp.edu.pe

Key words: Discrete element method (DEM), Neighbor particles, Searching methods.

Abstract. The performance of discrete element method (DEM) simulations is highly dependent on the requirements of the associated algorithms in terms of computer memory usage and CPU time. In particular, computer CPU time heavily depends on the identification of neighbor particles and the computation of particle-particle interactions. Over the years several neighbor particles searching methods have been developed. Accordingly, in this work the performance of two of well-known searching methods, linked cell and Verlet tables algorithms, are assessed in the context of the development of a new DEM-based tool. More specifically, the neighbor searching methods performance and related computational costs are parametrically analyzed and an assessment of their suitability for carrying the intended numerical simulations is provided. The referred numerical simulations are performed accounting for a canonical configuration used for the verification of the algorithms included in the new computational tool under development. The referred tool incorporating state of the art physical and numerical models will be used for modelling following a CFD-DEM approach mineral transport and grinding processes present in concentrator plants.

1 INTRODUCTION

Particulate flows (two-phase liquid-solid flows) known as pulp or slurry are common in both industry and nature. The pulp transport and wet milling processes present in the mining industry are difficult to characterize experimentally. Numerical models based on computational fluid dynamics (CFD) for instance allows getting a better understanding of the associated phenomena occurring in such flows. There are two main approaches commonly used for the modeling of particulate flows, Eulerian-Eulerian and Eulerian-Lagrangian. The approaches focused on the modeling of the referred flows can be also divided according to the treatment of the particulate's phase, continuous and discrete [1]. The continuous approach models large amounts of particles as an artificial continuous medium. One disadvantage of this approach is that the local behavior of individual particles is not accounted for. This modeling approach is closely related to the Eulerian-Eulerian one since usually the continuous approach relies on Eulerian treatments. The discrete approach in turn describes the movement and contact of each particle individually. The discrete element method (DEM) is one of the most important methodologies relying on a discrete approach. Compared to the continuous approach, the main disadvantage of the discrete one relates to its relatively high computational cost. The use of Lagrangian based techniques for tracking the transport and contact of the particles present in the flow is the main responsible for such high costs.

In the past pulps have been mainly simulated using Eulerian approaches [2] for both the solid phase and the transporting fluid. Eulerian methods are indeed able to accurately reproduce the particle concentration and velocity profiles, as well as they are capable of handling a large number of particles with a relatively low computational cost [3]. The main disadvantage of such approaches is that the detailed information at the macroscale and mesoscale is compromised by the approximations used. Contrarily, Lagrangian methods [4] are able to provide detailed information on the interactions between particle-particle, particle-fluid and particle-solid, but their computational costs are high. The referred costs are highly dependent on the number of Lagrangian particles being transported. Numerical simulations involving large numbers of particles generally requires the use of high performance computing (HPC).

The discrete element method (DEM) is one of the most important particle-based simulation methods that has been used for applications in several fields including chemical engineering, pharmaceutics, agriculture, energy, mining, environment and geological engineering [1]. The performance of DEM simulations is highly dependent on computer memory usage and CPU time [5]. The computer memory requirements come usually from the memory size used for storing both the mesh and the particle neighbor related information. Computer CPU time heavily depends in turn on the identification of neighbor particles and the computation of particle-particle interactions. Selecting the adequate neighbor particles searching methods and adjusting their associated parameters is thus criterial for the performance of DEM simulations.

There are several neighbor searching methods that have been tried in the past. They include for instance the linked cell [6] and the Verlet tables [7] ones. This work assesses the performance of these two searching methods in the context of the development of a new DEM-based tool. In particular, the neighbor searching methods performance and related computational costs are parametrically analyzed, and an assessment of their suitability for

carrying the intended numerical simulations is provided. The referred numerical simulations are performed accounting for a canonical configuration used for the verification of the algorithms included in the new computational tool under development. Accordingly, Section 2 briefly describes the general context in which this work has been carried out. Some specifics about the two several neighbor searching methods studied here are highlighted in Section 3. Finally, Sections 4 and 5 discuss, respectively, the main results obtained from the parametric assessments carried out and the main conclusions drawn from them. Notice that the DEM-based tool under development incorporating state of the art physical and numerical models will be used for modelling following a CFD-DEM approach mineral transport and grinding processes present in concentrator plants.

2 WORK CONTEXT

The neighbor particles searching methods assessed in this work constitutes one set of algorithms of the several ones composing a new DEM-based tool, the so-called CFLOWSS (Complex FLOWS Solver), currently under development. The development of this tool for the modeling of inert and reactive complex flows is continuous. Its modules, models and numerical algorithms are constantly updated in order to improve the accuracy of the flow modeling processes undertaken with its aid. The referred DEM-based tool (CFLOWSS) incorporating state of the art physical and numerical models will be used for modelling complex flows such as those involving mineral transport and grinding processes present in concentrator plants.

3 NEIGHBOR SEARCHING METHODS

The interaction between pairs of particles has to be evaluated during DEM-based simulations. A general algorithm accounting for all $N(N-1)/2$ interactions between the N particles involved in a given simulation constitutes the most basic particles searching method. In terms of computational effort, this base algorithm has an overall order of O(N2). There are of course other more time-efficient particles searching methods. Two of these methods, the linked cell algorithm (neighborhood tables) [6] and the Verlet tables [7], are studied here.

3.1 Linked cell algorithm

The linked cell algorithm [6], also called the neighborhood tables one, is based on the creation of a "mesh" of cells, where the cell size is a function of the particles sizes. The potential interacting pairs of a particle belonging to a given particular cell are determined from those ones belonging to the surrounding cells. Briefly, following this particles searching method, the algorithm first creates a mesh of a specified cell size and assigns each particle to its corresponding cell. The algorithm searches next the interacting pairs for a particle located at a given cell by analyzing the particles located in the surrounding cells. In order to avoid counting twice a given particle pair, the particle search processes are carried out from left to right and from bottom to top as shown in Figure 1. As highlighted in this figure, more than one line of neighbor cells around a given cell can also be accounted for during the neighbor particles searching processes.

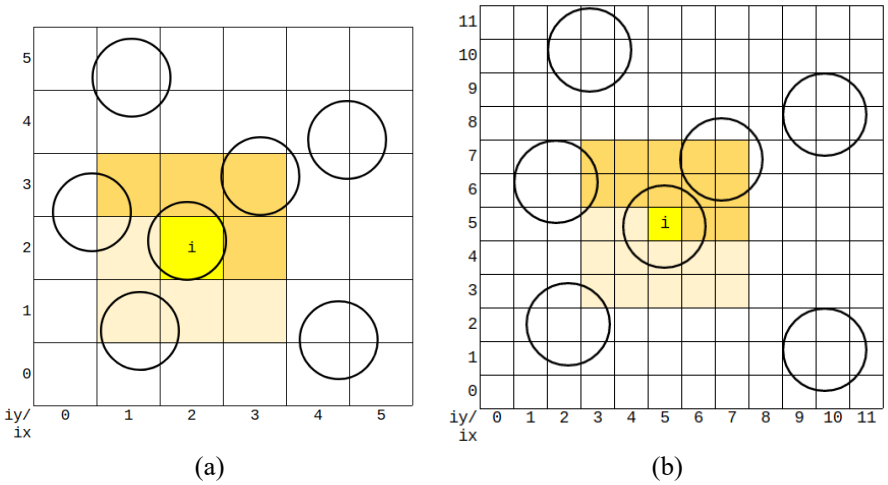

Figure 1: Linked cell algorithm with a uniform grid. The cell under analysis is shown in yellow, the neighbor cells considered for determining the interacting pairs are shown in strong orange, and the cells disregarded during the analyses are shown in light orange. Systems of monodisperse particles accounting for (a) one (1) line and (b) two (2) lines of neighbor cells around cell i.

3.2 Verlet tables algorithm

The Verlet tables algorithm is based on the original Verlet method [7]. This algorithm considers an imaginary sphere of radius R_{nbr} around a particle i, which contains the potential interacting particles. A pair of the round particles i and j is added to the neighborhood list of particle i if [1],

$$\|\bar{X}_i - \bar{X}_j\| < R_{nbr} = s + r_i + r_j \,, \tag{1}$$

where $\bar{X}$ is the particles position vector, s is the Verlet or skin parameter, and r is the particle radius. For a constant time step Δt and a maximum particle velocity V_{max}, the number of time steps in which the Verlet table is valid is computed from,

$$N_{verlet} = \frac{s}{2\,V_{max}}. \tag{2}$$

Following this algorithm thus, a Verlet table is initially created, which is used for N_{verlet} time steps. After this time has elapsed, the Verlet table is updated and a new N_{verlet} is computed. This process is repeated until reaching the end time.

4 RESULTS AND DISCUSSIONS

The numerical modeling of multiple packs of particles is useful for studying some of the physical phenomena characterizing mineral grinding processes and transport of dense slurries. In this work thus, numerical simulations of a dam break-like configuration are carried out in order to compare the performance of two different neighbor particles searching methods used in DEM-based approaches.

4.1 General description

As highlighted in Section 3, in addition to the general one, two neighbor particles searching methods, the linked cell algorithm (neighborhood tables) [6] and the Verlet tables [7], have been analyzed here. These algorithms have been implemented in the DEM-based tool under development (CFLOWSS) using C++ as the main programing language. In order to reduce the associated computational cost and to carry out several parametric studies, only two-dimensional simulations of a dam break-like configuration have been performed. For particle-particle and particle-wall interactions, normal elastic forces without damping (no tangential forces) were accounted for only. Three particle arrangements were tested varying the number of particles along the horizontal and vertical directions, 10 x 10, 16 x 16 and 22 x 22. The simulations were run on a single computer core of an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz processor.

Five dam-break related cases have been simulated considering a time step equal to 1E-6 s, from a physical time between 0 and 1 s. The particle governing equations are integrated using SUNDIALS/CVODE and the Adams scheme [8]. Two lines around a cell have been considered for the linked cell method (Figure 1b), and a skin factor equal to four (4) times the radius for the Verlet tables one. Elastic constants equal to 4000 and 40000 have been considered for the particle-particle and particle-wall contacts, respectively. The radius of the particles is equal to 0.5. For particular cases assessed here, some of the parameters highlighted before have been modified. The first case was simulated using the parameter values as just described. The CVODE solver based on Backward Differentiation Formulas (BDF) was used in the second case. Time steps equal to 2E-6 and 4E-6 are analyzed in the third and fourth cases, respectively. Finally, the skin factor used in the Verlet tables method has been varied in the fifth case. The skin factor variations ranged from 1 to 64 times the particles radius. The main results obtained from the simulations of the five cases highlighted above are discussed in the following section.

4.2 Results

Figure 2 shows as a function of the number of particles the CPU time associated with the general search algorithm and the two studied neighbor searching methods. As noticed from this figure, the CPU time is significantly reduced when neighbor searching algorithms are utilized. In addition, when compared each other the linked cell and the Verlet tables algorithms, the former is slightly less time consuming. This particular outcome has been observed in all results obtained in this work.

The results associated with the use of two different numerical schemes present in the CVODE solver utilized here are highlighted in Figure 3. It is noticed in particular from these results that for relative small time steps the Adams scheme is slightly more time efficient than the BDF one. The referred differences in CPU time increase with the increase in the number of particles accounted for in the numerical simulations.

In Figure 4, the CPU times as a function of the number of particles characterizing the first, third and fourth cases are shown. These results indicate that the CPU time is proportionally reduced when the time step is doubled. It is worth noticing that, as shown in Figure 7, the results do not differ when increasing the time step size. This means that the impact of the numerical errors associated with the use of these bigger time steps on the particles transport is negligible.

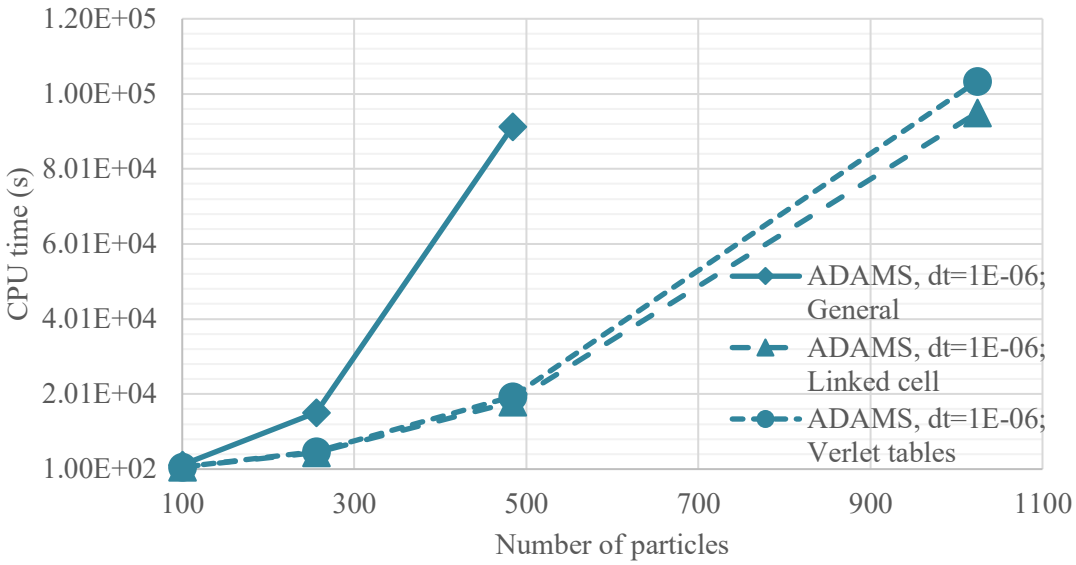

Figure 2: CPU time comparison between neighborhood searching methods for Case 1 (ADAMS, dt=1E-06).

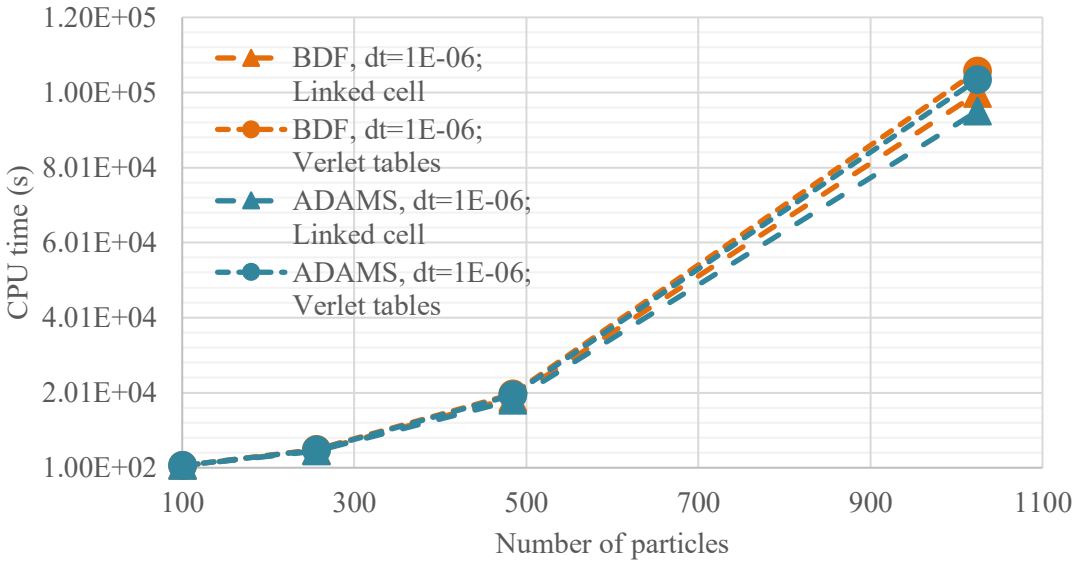

Figure 3: CPU time comparison between Case 1 (ADAMS, dt=1E-06) and Case 2 (BDF, dt=1E-06).

The influence of the skin factor s on the CPU time (Case 5) is highlighted in Figure 5. As noticed on this figure, the CPU time is the lowest when the skin factor takes a value equal to one (1), i.e., equal to the particle radius. In addition, the CPU time increases with the increase in the skin factor values. The highest CPU time that can be obtained by increasing the skin factor is theoretically equal to that characterizing the general algorithm (15114 s). This last

aspect is reflected in the essentially constant CPU time values obtained for skin factors higher than 32 times the particle radius. Finally, Figure 8 compares the results obtained with a $s = 4r$ (Case 1) and $s = r$ (Case 5). The similarity of the results shown in this figure indicates that the implemented algorithms work without any detection problems with the skin factor modifications analyzed here.

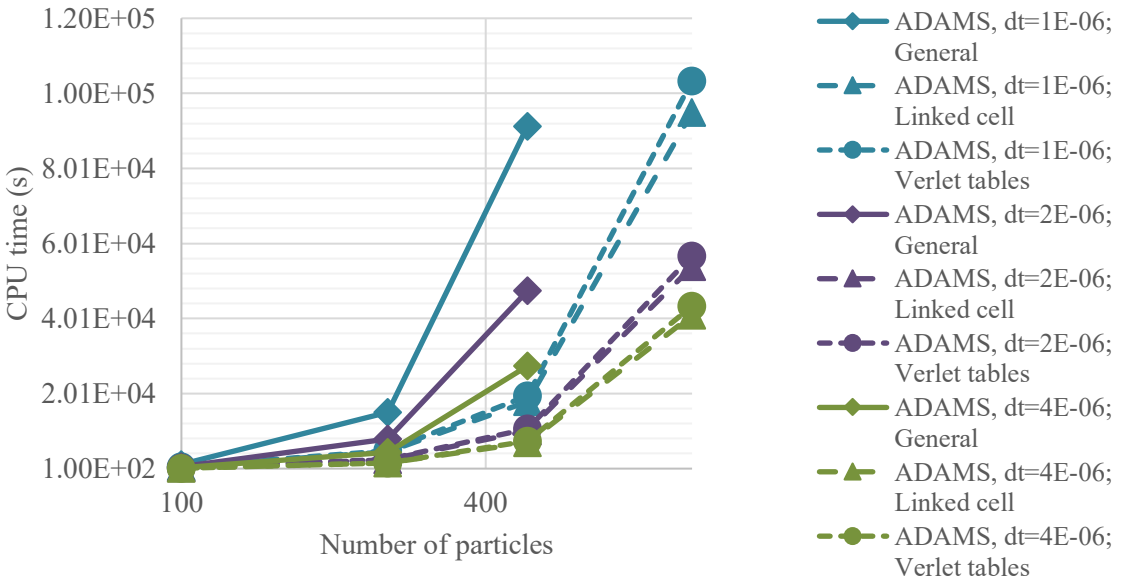

Figure 4: CPU time comparison between Case 1 (ADAMS, dt=1E-06), Case 3 (ADAMS, dt=2E-06) and Case 4 (ADAMS, dt=4E-06).

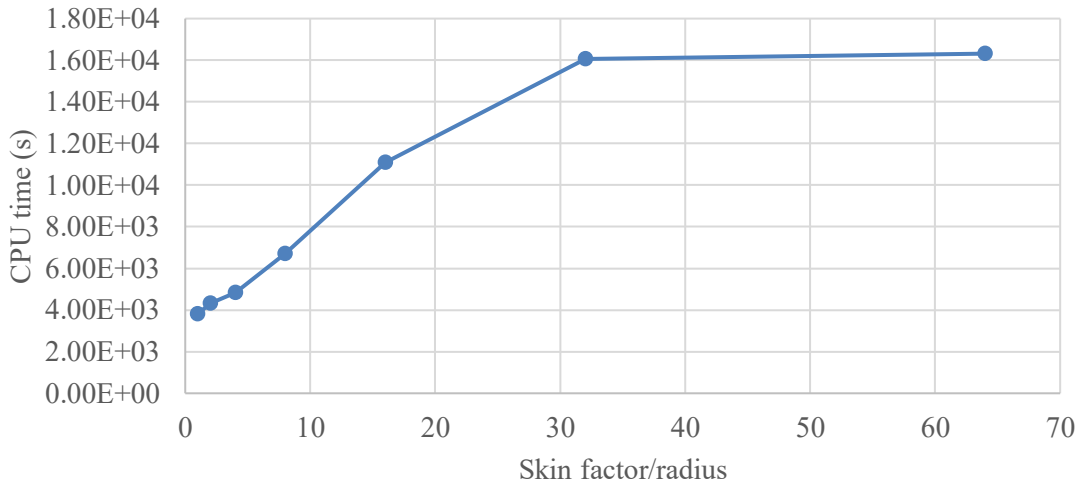

Figure 5: CPU time as a function of the ratio between skin factor and particles radius (Case 5).

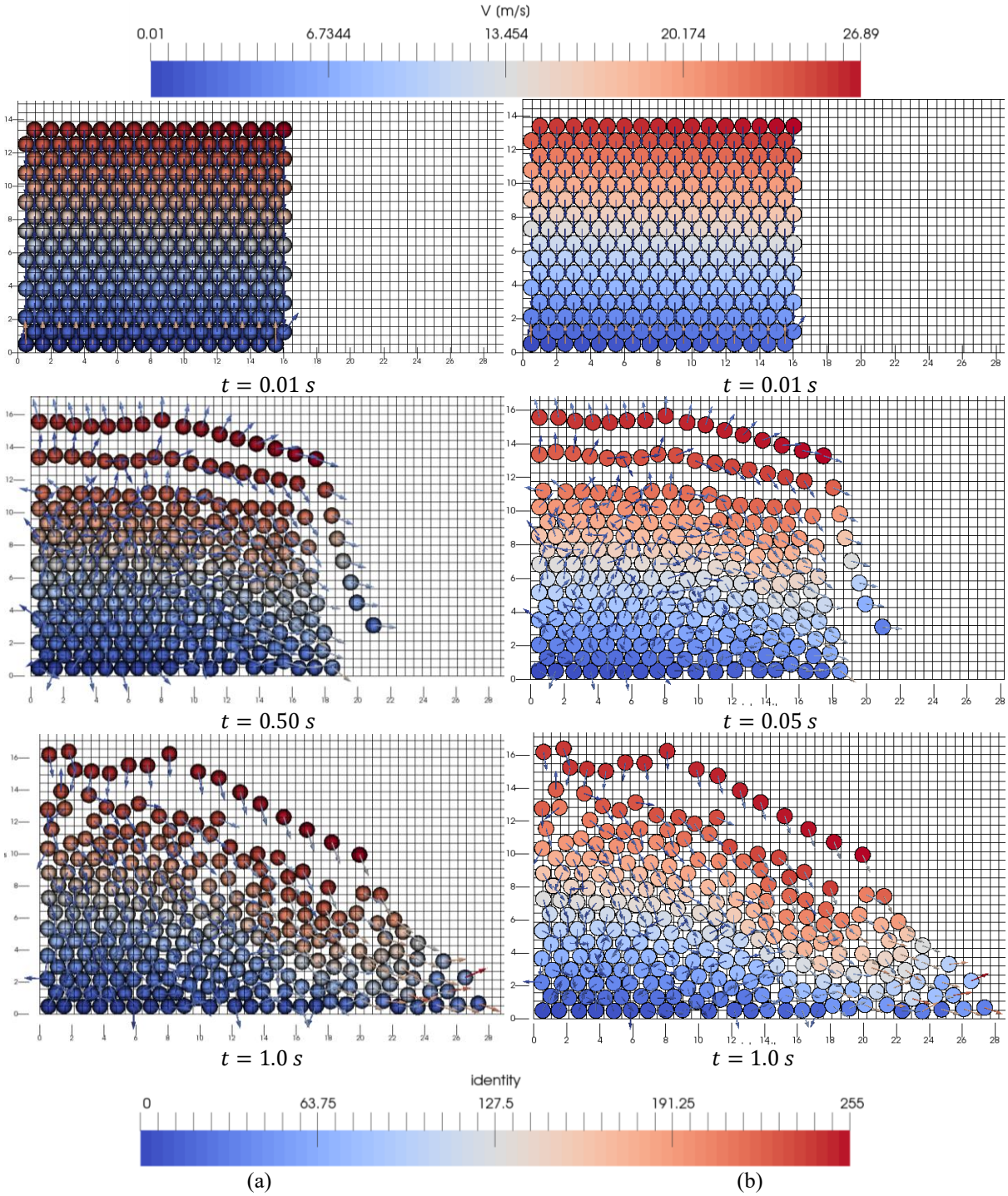

Figure 6: (a) Time evolution comparison between (a) Case 1 (ADAMS, dt=1E-06) and (b) Case 2 (BDF, dt=1E-06) using the Verlet tables algorithm for a dambreak having an initial 16 x 16 (256 particles) arrangement.

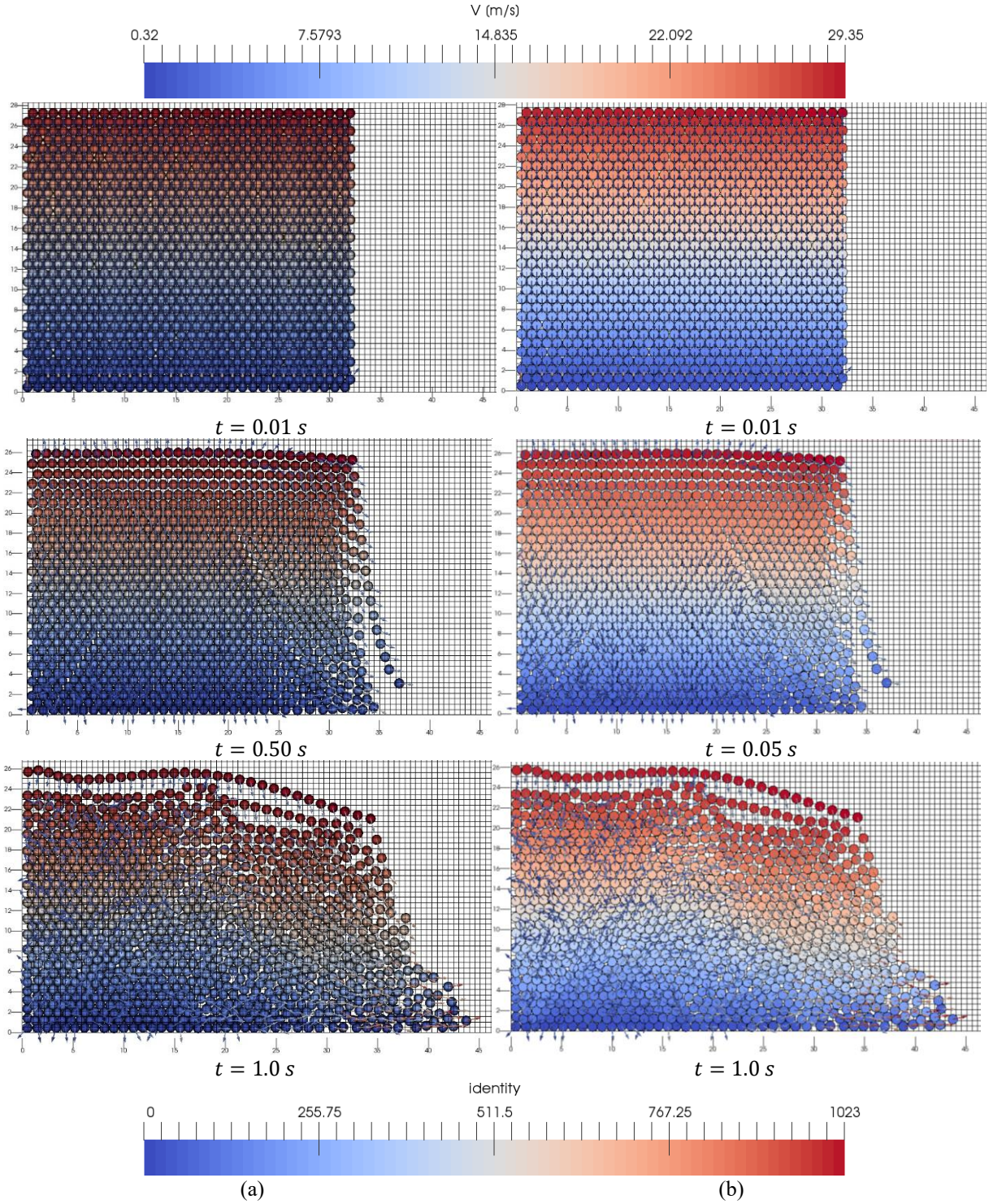

Figure 7: Time evolution comparison between (a) Case 1 (ADAMS, dt=1E-06) and (b) Case 4 (ADAMS, dt=4E-06) using the Linked cell algorithm for a dambreak having an initial 32 x 32 (1024 particles) arrangement.

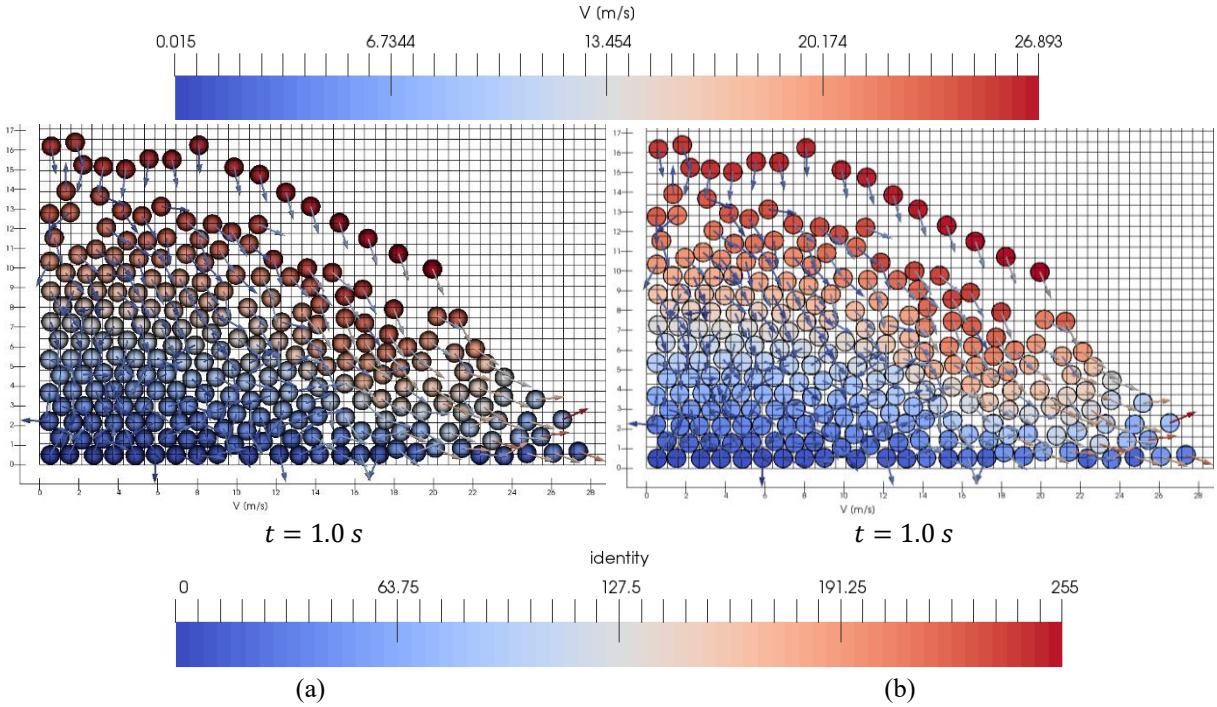

Figure 8: Time evolution comparison between (a) Case 1 (ADAMS, dt=1E-06) with $s = 4r$ and (b) Case 4 (ADAMS, dt=4E-06) with $s = r$ using Verlet tables for a dambreak having an initial 16 x 16 (256 particles) arrangement.

5 CONCLUSIONS

Two classical neighbor particles searching methods, Linked cell algorithms and Verlet tables, have been implemented in a new DEM-based tool, the so-called CFLOWSS. A two-dimensional dam break-like configuration has been numerically simulated using the referred two algorithms and the general searching method. Different number of particles and skin factors for the Verlet tables have been studied. The main results indicate that the Linked cell algorithms and Verlet tables present as expected lower CPU times compared to the general searching method. In addition, there are no significant differences in terms of CPU time when using Adams or BDF solver schemes. Increasing the time step from 1E-06 to 2E-06 and 4E-06 leads to savings in CPU time and there are no noticeable differences in the associated numerical results. Finally, the Verlet tables show great dependence on the skin factor and a value for this parameter equal to the particle radius do not result in particle pair detection problems.

6 ACKNOWLEDGEMENTS

This work has been supported by the Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica (FONDECYT), an initiative from the Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica (CONCYTEC) (Peru) under Contract No. 155-2018-FONDECYT-BM-IADT-AV. This work has been also supported by the Engineering Doctoral program at Pontificia Universidad Católica del Perú (PUCP), partially funded by CONCYTEC and the World Bank.

REFERENCES

[1] C. Kloss, C. Goniva, A. Hager, S. Amberger y S. Pirker, «Models, algorithms and validation for opensource DEM and CFD-DEM,» *Progress in Computational Fluid Dynamics,* vol. 12, nº 2/3, 2012.

[2] L. Chen, Y. Duan, W. Pu y a. C. Zhao, «CFD simulation of coal-water slurry flowing in horizontal pipelines,» *Korean J. Chem. Eng.,* vol. 26, nº 4, pp. 1144-1154, 2009.

[3] J. Capecelatro y O. Desjardins, «Eulerian–Lagrangian modeling of turbulent liquid–solid slurries in horizontal pipes,» *International Journal of Multiphase Flow,* vol. 55, pp. 64-79, 2013.

[4] W. Zhong, X. L. Aibing Yu, Z. Tong y H. Zhang, «DEM/CFD-DEM Modelling of Non-spherical Particulate Systems: Theoretical Developments and Applications,» *Powder Technology,* vol. 302, pp. 108-152, 2016.

[5] R. Cai, L. Xu, J. Zheng y Y. Zhao, «Modified cell-linked list method using dynamic mesh for discrete element method,» *Powder Technology,* vol. 340, pp. 321-330, 2018.

[6] H.-G. Matuttis y J. Chen, Understanding the Discrete Element Method, Singapore: John Wiley & Sons, 2014.

[7] L. Verlet, «Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules,» *Physical Review,* vol. 159, nº 1, pp. 98-103, 1967.

[8] Lawrence Livermore National Laboratory, Center for Applied Scientific Computing (2016), «CVODE v4.1.0».