XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

ON THE DEVELOPMENT OF DOMAIN PARTITIONING TECHNIQUES APPLIED TO THE MATERIAL POINT METHOD

João Gabriel da Costa de Souza Duarte
joao.duarte@ctec.ufal.br
Undergraduate Student, Center of Technology, Federal University of Alagoas
Av. Lourival Melo Mota, S/N, 57072-900, Maceió/Alagoas, Brazil
Adeildo Soares Ramos Júnior
adramos@lccv.ufal.br
Associate Professor, Center of Technology, Federal University of Alagoas
Av. Lourival Melo Mota, S/N, 57072-900, Maceió/Alagoas, Brazil
Diogo Tenório Cintra
diogotc@lccv.ufal.br
Researcher, Laboratory of Scientific Computing and Visualization
Av. Lourival Melo Mota, S/N, 57072-900, Maceió/Alagoas, Brazil
Ricardo Garske Borges
garske@petrobras.com.br
Project manager, CENPES, Petrobrás
Av. Horácio Macedo, 950, 21941-915, Rio de Janeiro/Rio de Janeiro, Brazil

Abstract. This paper aims the development of domain partitioning techniques applied to numerical simulations using the Material Point Method (MPM). The MPM can be used to simulate various engineering problems, including those involving submarine landslides, installation of torpedo anchors and dynamical analysis of structures. The parallel solution of MPM involves the use of parallel computing libraries and domain partitioning techniques. This procedure potentially increases the processing speed of the problem and, consequently, reduces computational time. In this work, two geometric domain partitioning techniques were adopted: 1) division by horizontal or vertical bands; 2) the Recursive Coordinate Bisection (RCB) method. Both partitioning algorithms were developed using the C/C++ language and the Message Passing Interface (MPI) library. The MPI library allows the exchange of data among the several processors used in the simulations considering the use of distributed memory systems. In order to measure the partitioning quality for the obtained results in several simulations, the load balance parameter was adopted. The obtained values were compared to the optimal value calculated analytically. The results show that the RCB implementation drastically reduces the communication regions between the processors, which represents a computational gain in simulation time. Therefore, the RCB partitioning technique has a better performance, compared to the technique of division by bands, since the method optimizes the partitioning, due to its recursive and iterative characteristics.

Keywords: Parallel Computing, Domain Partitioning Techniques, Material Point Method, Message Passage Interface, Load Balance

1 Introduction

Following the advances of computing power and data processing, large-scale problem solving, from a wide range of engineering fields, has become possible and recurrent. For the solution of this problems, a good computational performance is necessary, considering an efficient numerical method the that reproduce suitable results to the problematic in question. In this sense, in order to perform the simulation of dynamics of many particles, or material points, the Material Point Method (MPM) can be used. MPM allows the discretization of a continuous environment in a finite number of material points, as for example in the dynamical analysis of structures, simulation of submarine landslides and installation of torpedo anchors. Its formulation is based on solutions obtained through meshes and sets of discrete points [1]. However, aiming to deal with the large number of particles present in simulations, the MPM requires domain partitioning techniques to improve the computational performance of the method and to ensure load balancing between sub-domains created.

Cintra *et. al* [2] state that the domain partitioning techniques provide an approach to the parallel simulation of particles. Typically, these techniques operate by dividing the set of particles that represent the original domain. Each sub-domain (or rank) generated is assigned to a processor which is responsible for the related calculations. The used method brings together the nearby points, distributing them approximately equally to the sub-domains created, since distribution errors of points will always exist, however small. But, these errors tend to increase when the domain is dynamic, thus damaging the load balance between sub-domains. According to Eibl and Rüde [3], this leads to load imbalances, that can slow down the whole simulation. To overcome this problem, the domain partitioning must be adapted dynamically during the simulation and / or the sub-domains must be reassigned to different processes.

This paper is structured as follows. In section 2, we describe the work approach. Section 3 describes the techniques applied, implementation procedures for each technique and the tools used. The results obtained from the study are shown in the same section. The conclusions are explained in section 4.

2 Our approach

In a general context, MPM simulations are costly due to the large number of particles that the method uses. So they require a high computational performance, especially when working with a dynamic particle domain. Choosing the proper partitioning technique is important in order to obtain appropriate results against the adopted metric. Our approach is based on two types of geometric domain partitioning techniques to generate the eight ranks used in all simulations: 1) division by horizontal or vertical bands and 2) the Recursive Coordinate Bisection (RCB) method. In addition, we use the load balance as a comparative parameter between the processes to investigate which partitioning technique produces the best results considering a static domain with thousands of particles. To analyze the dynamic domain, we use the RCB method.

3 Geometrical partitioning techniques and implementation procedure

Considering the large number of calculations to solve the problem and a large number of elements moving in the mesh of the MPM, dividing the domain of material points becomes essential, in order to increase the speed of problem solving. The study was started by a static domain with 1,068,285 points, where no partitioning technique (geometric or topological) was used. Table 1 shows the huge difference for the number of points per sub-domain created. Percent errors obtained, in absolute value, from an optimal number of points per rank are also shown in the same table, which in this case is 133,910 points. The optimal number was obtained by dividing the total number of domain points by the number of processes used in the application.

Table 1. Percent unbalance per sub-domain

Rank	Number of points obtained			Relative Error (%)		
	Without any method	DbB	RCB	Without any method	DbB	RCB
0	196435	134327	133671	46.69	0.311	0.178
1	131307	133974	133896	0.073	0.048	0.010
2	156204	134760	133863	16.65	0.634	0.035
3	143512	134589	134051	7.170	0.507	0.105
4	196099	133419	133764	46.44	0.367	0.109
5	35338	134672	133907	73.61	0.569	0.002
6	158909	134843	133813	18.67	0.697	0.072
7	53481	129703	134322	60.06	3.140	0.307

3.1 The Division by Bands method in a static domain

The geometric technique of Division by Bands (DbB) consists of two basic models: vertical or horizontal bands. In this technique of partitioning, considered of low complexity of computational algorithm implementation, the domain is sectioned in the direction orthogonal to bands to the horizontal or vertical axis.

The computational algorithm implementation procedure made use of the C/C++ languages, aided by the Message Passing Interface (MPI) library, which aimed to establish the connection between the sub-domains created to make it possible to transfer data between them through communicative functions [4].

In addition, the Cartesian coordinates of each point were used to group the points and to create balanced sub-domains and cutting coordinates. The algorithm was developed in a generic way, but was executed to operate in eight different ranks. In this sense, process 0, or root process, is responsible for defining the position of each section of the domain, taking into account the ideal number of points in each sub-domain, dividing the original domain in a way that all sub-regions created have approximately the same amount of material points. So the total load is balanced among the processes and none of them is overloaded. After the section coordinates are found by the root process, it sends such information to the other processes, through the communication functions of the MPI library. After that, the procedure of loading of points in the RAM memory is started and performed individually and simultaneously by the all ranks, according to the previously defined cutting coordinates.

In this sense, from the input file containing the points coordinates, the domain was sectioned according to the decomposition algorithm and output files were generated, one for each process. Table 1 shows the errors obtained by the partitioning algorithms used in this work.

3.2 The Recursive Coordinate Bisection method in a static domain

The technique of partitioning through the Recursive Coordinate Bisection (RCB) method consists of analyzing the amplitudes of the Cartesian axes and performing a section perpendicular to the axis of greater amplitude, creating two sub-domains with each cut. This procedure happens in a recursive way,

and it is stopped when there are no more processes available to be used by the partitioning algorithm. It is worth mentioning that the implementation of the RCB method is more complex and higher computational cost, compared to the Division by Bands method, and there may be several ways of accomplishing it. However, the main characteristic of the RCB method is to decrease the areas of communication between the processes, which represents a computational gain with respect to the processing of data that travels between ranks.

As in the Division by Bands method, the implementation procedure of the RCB computational algorithm used C/C++ languages, as well as the parallel computing MPI library. The difference between the methods is the way the RCB works. At each iteration, process 0 performs a check to find out if ranks are available. If so, partitioning continues until all sub-domains are defined and the load balance is approximately equal to all, dynamically storing the cut coordinates obtained in that iteration; otherwise, partitioning is complete.

After the partitioning step is completed, the root process sends the Coordinate Vectors to the other ranks of the simulation through the communication network created by the MPI library and the algorithm advances to the step of loading points in the RAM memory. For each region created from of the original domain partitioning, where each point will be allocated in its respective sub-domain. The loading process takes place simultaneously among all ranks.

The Table 1 shows more details about the error caused by the partitioning algorithm using the load balance in each rank on metric. The improvement in load balancing is remarkable when compared to Division by Bands method. This is due to the fact that the technique starts from a statistical principle of median to find the cut coordinate, which explains the improvement in the load balance of each sub-domain, and its recursion to perform the partitioning.

3.3 The Recursive Coordinate Bisection method in a dynamic domain

When we verify that the RCB method was the one that presented the best results in the load balance question, we started to execute the partitioning algorithm in simulations with dynamic domains. In these domains, the material points move after each time step. Since we already have studied the load balancing capability of each partitioning technique, we are now interested in verifying the integrity of partitioning and the need to partition the domain since it is dynamic. For that, a domain with 1,500 points and 200 steps of time was used to complete the computational simulation of two colliding spheres, as Landau and Lifshitz approached [5].

Initially, the partitioning algorithm performed its function and did not re-partition the domain, regardless of the error values. This is shown in Fig. 1, where the largest relative error among all ranks is shown in the course of the simulation time steps. Then, we adopted a tolerance of 0.1 for the relative maximum load balance error. Therefore, when some rank presented an error greater than 0.1, it would verify this condition and automatically execute a new re-partition. In this sense, the time error graph was elaborated, as shown in Fig. 1.

Therefore, when comparing the two histories below, we can note the difference of errors that exist between the two types of partitioning using the RCB method. In addition, the simulation time is impaired because there is no effective load balance between the ranks. In the same Fig. 1, it may be noted that the error present in the ranks is ”controlled”, since there is an admittance limit. Thus, when some rank has an error greater than or equal to 0.1, the partitioning algorithm is executed again and redistribution of points happens in the next step of the simulation, which is indicated by the red dot.

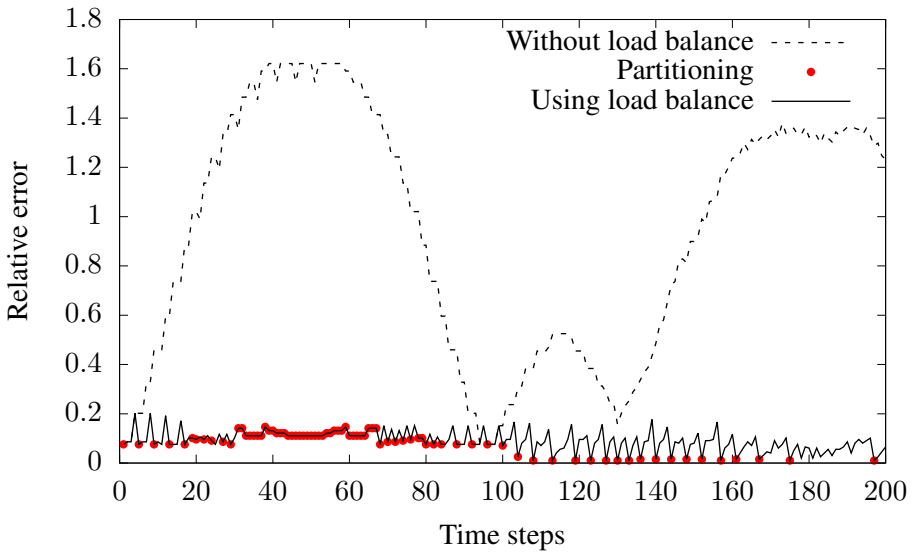

Figure 1. Relative error history

4 Conclusions

This paper showed how geometric domain partitioning techniques behave when implemented together with MPM, as well as producing errors that can not be avoided, but minimized. Considering the above, we can conclude that the RCB method presented better load balance results among the ranks used in the execution of the domain partitioning algorithm. In addition, the RCB method was tested in a dynamic domain and also demonstrated good results for dynamic load balancing.

Thus, the RCB is efficient in this adopted parameter of tolerance previously established, and is still able to reduce the interaction regions between the sub-domains created in the original domain partitioning, which represents a significant performance gain, since the areas of communication between the processes were reduced, allowing an optimization in the passage of information.

References

[1] K. J. al Kafaji, I., 2013. *Formulation of a Dynamic Material Point Method (MPM) for Geomechanical Problems*. PhD thesis.

[2] Cintra, D. T., Willmersdorf, R. B., Lyra, P. R. M., & Lira, W. W. M., 2016. A hybrid parallel DEM approach with workload balancing based on HSFC. *International Journal for Computer-Aided Engineering and Software*, vol. 33, pp. 2264–2287.

[3] Eibl, S. & Rüde, U., 2018. A systematic comparison of dynamic load balancing algorithms for massively parallel rigid particle dynamics. *CoRR*, vol. abs/1808.00829.

[4] Pacheco, P., 1997. *Parallel Programming with MPI*. Morgan Kaufmann.

[5] Landau, L., Lifshitz, E., Sykes, J., Reid, W., & Dill, E. H., 1960. Theory of elasticity: Vol. 7 of course of theoretical physics. *Phys. Today*, vol. 13, pp. 44.