



**HAL**  
open science

## Domain decomposition with discrete element simulations using shared-memory parallel computing for railways applications

Thi Minh Phuong Hoang, Gilles Saussine, David Dureisseix, Pierre Alart

### ► To cite this version:

Thi Minh Phuong Hoang, Gilles Saussine, David Dureisseix, Pierre Alart. Domain decomposition with discrete element simulations using shared-memory parallel computing for railways applications. *Revue Européenne de Mécanique Numérique/European Journal of Computational Mechanics*, 2012, 21 (3-6), pp.242-253. 10.1080/17797179.2012.714723 . hal-00797593

**HAL Id: hal-00797593**

**<https://hal.science/hal-00797593>**

Submitted on 6 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Domain decomposition with discrete element simulations using shared-memory parallel computing for railways applications

**T. M. P. Hoang<sup>\*,\*\*\*</sup> — G. Saussine<sup>\*</sup> — D. Dureisseix<sup>\*\*</sup> — P. Alart<sup>\*\*\*</sup>**

<sup>\*</sup> *Société Nationale des Chemins de Fer (SNCF), Innovation and Research Division, 45 rue de Londres, F-75379 Paris Cedex 08*

*{thi-minh-phuong.hoang, gilles.saussine}@snCF.fr*

<sup>\*\*</sup> *Laboratoire de Mécanique des Contacts et des Structures (LaMCoS), INSA Lyon / CNRS UMR 5259, 18-20 rue des Sciences, F-69621 Villeurbanne Cedex*

*david.dureisseix@insa-lyon.fr*

<sup>\*\*\*</sup> *Laboratoire de Mécanique et Génie Civil (LMGC), Université Montpellier 2 / CNRS UMR 5508, CC048, Place Eugène Bataillon, F-34095 Montpellier Cedex 5*

*pierre.alart@univ-montp2.fr*

---

*ABSTRACT. Numerical simulation with discrete elements leads to several issues for large scale problems and long loading times, as for the granular dynamic simulations of the ballasted railway behavior. To reduce computational costs, we study the use of two strategies: domain decomposition methods and shared-memory parallelization with OpenMP. An example of a maintenance process, the tamping, on a portion of railway track with 7 sleepers, is simulated.*

*RÉSUMÉ. La simulation numérique par éléments discrets présente des difficultés pour l'étude de problèmes de grande taille et en temps de sollicitation long, comme la dynamique des milieux granulaires pour le ballast ferroviaire. Afin de résoudre ce problème à moindre coût, on propose d'allier deux stratégies : la décomposition de domaine (DDM) et le calcul parallèle (en mémoire partagée avec OpenMP). Un exemple traitant d'un procédé de maintenance ferroviaire, le bourrage, sur une portion de voie ballastée de 7 blochets de long est étudié.*

*KEYWORDS: Non Smooth Contact Dynamics, parallelization, LMGC90, ballast, maintenance.*

*MOTS-CLÉS : Non Smooth Contact Dynamics, parallélisation, LMGC90, ballast, maintenance.*

## 1. Motivations: tamping process optimization

The degradation of ballasted railway under commercial exploitation requires frequent and costly maintenance operations. One of these operations is the tamping process (Paderno, 2010; Azéma, 2007; Azéma *et al.*, 2009; Solomon, 2001) which is intended to correct the railway geometry that have been modified by an inhomogeneous vertical settlement. It is performed by a ballast tamper or a tamping machine which works by vibrating the ballast and pushing it under the sleepers. The study of the mechanical behavior of the ballast during tamping is necessary to propose optimizations of this maintenance process (Perales *et al.*, 2009).

The numerical simulation of a portion of a railway submitted to the tamping process is useful to master the physical phenomenons involved in the ballast (Saussine, 2004). However, the numerical complexity of a simulation at the grain scale requires special treatments to get an affordable simulation.

Within this article, the *Non Smooth Contact Dynamics* (NSCD) model is used for the simulation of the ballasted railway, with a discrete element model (DEM). A domain decomposition method (DDM) coupled with a shared-memory parallelization technique (using OpenMP) are used to improve the computational efficiency and to reduce the cost and the run time of the simulation. The domain is geometrically split into several subdomains, allowing their simultaneous treatment on several processors or computer cores.

The main results are related to a representative simulation of a portion of a railway track with 7 sleepers, submitted to successive tamping processes. The structure is composed of up to 90 000 polyhedral grains, and at each time step, about 300 000 frictional contacts are involved. The post-treatment concerns the influence of the process under each sleeper, in terms of compactness, and several indicators are used to assess the quality of the numerical simulation, and of the physical quality of the obtained railway.

## 2. Numerical method

### 2.1. Granular dynamics

The *Non Smooth Contact Dynamics* (NSCD) approach developed by Moreau and Jean (Moreau, 1999; Jean, 1999; Cambou *et al.*, 2001) is a discrete element method that may be applied to a large range of discrete systems involving various interactions between elements and various regimes (quasi-static, slow and fast dynamics, dense and diluted granulates, rigid and deformable bodies). In the present framework the NSCD is applied to a collection of rigid polyhedral grains with frictional contact interactions. The main features of the approach are the reduced dynamics and the nonsmooth formulation of contact laws.

### 2.1.1. Dynamic equation

For each grain, according to the NSCD approach, the discrete dynamic equation over a time step  $[t_i, t_{i+1}]$  may be written,

$$M(V - V^i) = R^d + R \quad [1]$$

where  $M$  denotes the inertia matrix,  $V^i$  and  $V$  are the generalized velocities of the grains at the instants  $t_i$  and  $t_{i+1}$  (the subscript  $i + 1$  is omitted for the following).  $R^d$  represents the external impulses and  $R$  are the impulses due to all contacts on the grain over the time step. All the equations for all the grains may be concatenated formally leading to the same expression.  $V$  and  $R$  have to be determined.

The previous equation [1] may be reduced to the local variables of the contacts, the relative velocity  $v$  and the contact impulse  $r$ . The local variables are defined according to a local-to-global (or contact-to-grain) mapping  $H$  and its transpose  $H^T$ :  $v = H^T V$  and  $R = Hr$ . The dynamics is then given by  $v = v^d + Wr$ , where  $v^d = H^T(M^{-1}R^d + V^i)$ , and  $W = H^T M^{-1} H$  is the so-called Delassus operator.

### 2.1.2. Contact law

According to the viability lemma (Moreau, 1999), for each contact, the unilateral contact may be formulated with a complementarity condition between the normal relative velocity  $v_n^\alpha$  and the normal contact impulse  $r_n^\alpha$ , once the penetration tested with a predicted gap  $g^\alpha$ , for the contact  $\alpha$ . The frictional contact law is then,

$$\left\{ \begin{array}{l} \text{if } g^\alpha > 0, r^\alpha = 0 \\ \text{if } g^\alpha = 0, 0 \leq v_n^\alpha \perp r_n^\alpha \geq 0 \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{l} \text{if } \|v_t^\alpha\| = 0, \|r_t^\alpha\| \leq \mu r_n^\alpha \\ \text{if } \|v_t^\alpha\| \neq 0, r_t^\alpha = -\mu r_n^\alpha v_t^\alpha / \|v_t^\alpha\| \end{array} \right.$$

This law may be written formally :  $\mathcal{R}(r, v) = 0$ .

The reference problem consists then in determining the contact impulse and the relative velocity of each contact satisfying the following system,

$$\left\{ \begin{array}{l} v = v^d + Wr \\ \mathcal{R}(r, v) = 0 \end{array} \right.$$

### 2.1.3. Non Linear Gauss Seidel solver (NLGS)

A Non Linear Gauss-Seidel type algorithm based on a classical block partitioning of the  $W$  matrix amounts to find for each contact  $\alpha$  and for each iteration  $k$  the pair  $(r_\alpha^{k+1}, v_\alpha^{k+1})$  verifying the local nonsmooth system,

$$\left\{ \begin{array}{l} W_{\alpha\alpha} r_\alpha^{k+1} - v_\alpha^{k+1} = -v_\alpha^d - \sum_{\beta < \alpha} W_{\alpha\beta} r_\beta^{k+1} - \sum_{\beta > \alpha} W_{\alpha\beta} r_\beta^k \\ \mathcal{R}(r_\alpha^{k+1}, v_\alpha^{k+1}) = 0 \end{array} \right.$$

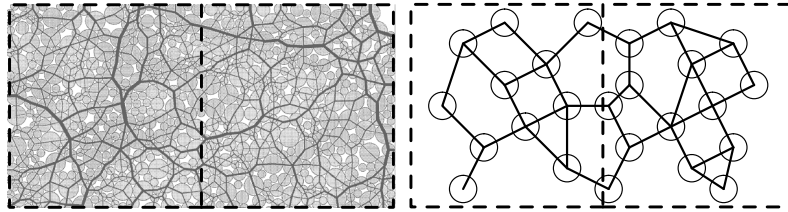
## 2.2. Domain decomposition and parallel technique

### 2.2.1. Geometrical partitioning

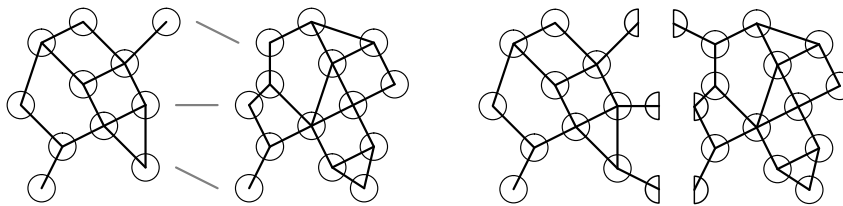
A dense collection of rigid bodies may be viewed as a set of nodes (grains) and a set of links (contacts) shown on Figure 1. Then two partitioning techniques may be performed (Champaney *et al.*, 2007).

– *The primal approach* distributes the grains into the subdomains (Figure 2 left). A grain belongs to a subdomain if the coordinates of its mass center is inside a geometrical box previously defined by a regular grid. The interface consists then of *links* between grains of two neighboring subdomains (Hoang *et al.*, 2011b; Hoang *et al.*, 2011a).

– *The dual approach* distributes the links into the subdomains (Figure 2 right). A link belongs to a subdomain if the barycenter of the two contacting grains is inside the box. The interface consists then of grains with contacts belonging at least to two subdomains (Iceta, 2010; Iceta *et al.*, 2009).



**Figure 1.** Substructuring with a box-type method (a granular example and the schematic principle).



**Figure 2.** Primal vs dual approaches.

In this study the *primal approach* is chosen because it is less intrusive in an industrial software than the dual one. The interface is constituted with contacts only, while the subdomains contain grains and contacts. Therefore the interface is numerically dealt as one subdomain with a nonsmooth solver. This appears clearly when one examines the algebraic partitioning deriving from the geometric substructuring.

### 2.2.2. Algebraic partitioning

According to the formulation developed in (Champaney *et al.*, 2007; Nineb *et al.*, 2007) the subscripts  $E$ ,  $E'$ ,  $\Gamma$  denote in the following the variables of the subdomain  $E$ , respectively  $E'$  and of the interface  $\Gamma$ . For sake of simplicity two subdomains are only considered. The capital letters refer to variables associated with grains whereas the lowercase letters refer to variables of the contacts. So the velocities of the grains and the impulsions applied to the grains are defined for all the subdomains as follows,

$$V = \begin{bmatrix} V_E \\ V_{E'} \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} R_E \\ R_{E'} \end{bmatrix}$$

On the contrary the relative velocities at contacts and the contact impulsions are written for all the subdomains and for the interface,

$$v = \begin{bmatrix} v_E \\ v_{E'} \\ v_\Gamma \end{bmatrix} \quad \text{and} \quad r = \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix}$$

The dynamic equation of the grains belonging to the subdomain  $E$  is given as,

$$M_E V_E = M_E V_E^i + R_E^d + R_E + R_{E\Gamma}$$

where  $R_{E\Gamma}$  are the impulsions of the interface  $\Gamma$  applied to the subdomain  $E$ ,

$$R_E = H_E r_E \quad \text{and} \quad R_{E\Gamma} = H_{E\Gamma} r_\Gamma$$

The dual relations provide the relative velocities starting from the generalized velocities of the grains,

$$v_E = H_E^T V_E \quad \text{and} \quad v_\Gamma = \sum_E H_{E\Gamma}^T V_E$$

The reduction of the dynamics to the contacts leads to a partitionned system,

$$\begin{bmatrix} v_E \\ v_{E'} \\ v_\Gamma \end{bmatrix} = \begin{bmatrix} v_E^d \\ v_{E'}^d \\ v_\Gamma^d \end{bmatrix} + \begin{bmatrix} W_E & 0 & W_{E\Gamma} \\ 0 & W_{E'} & W_{E'\Gamma} \\ W_{\Gamma E} & W_{\Gamma E'} & W_\Gamma \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix} \quad [2]$$

with  $v_E^d = H_E^T (V_E^i + M_E^{-1} R_E^d)$ ,  $v_\Gamma^d = \sum_E H_{E\Gamma}^T (V_E^i + M_E^{-1} R_E^d)$ ,  $W_E = H_E^T M_E^{-1} H_E$ ,  $W_{E\Gamma} = W_{\Gamma E}^T = H_E^T M_E^{-1} H_{E\Gamma}$ , and  $W_\Gamma = \sum_E H_{E\Gamma}^T M_E^{-1} H_{E\Gamma}$ .

The parallel process is performed with a shared memory architecture and multi-threading OpenMP directives inserted in the program. Such a strategy is the simplest one to implement in a pre-existing industrial software. The generic solver for the Non-Smooth Contact Dynamics is a NonLinear Gauss Seidel algorithm. In a first version it is applied simultaneously to all the  $n_{SD}$  subdomains before solving the global interface. This specific treatment of the coupling interface provides then a synchronous algorithm.

The final algorithm described in Algorithm 1 associates a NonLinear Gauss Seidel method, a Domain Decomposition method and Open-MP directives (DDM–OpenMP–NLGS). The algorithmic parameters are: the number of subdomains  $n_{SD}$ , the number of DDM iterations  $n_{DDM}$ , the number of NLGS iterations in the subdomains  $n$ , the number of NLGS iterations in the interface  $m$ . The emphasized parts are additional tasks to allow the DDM and parallel treatment.

---

**Algorithm 1** DDM–OpenMP–NLGS
 

---

```

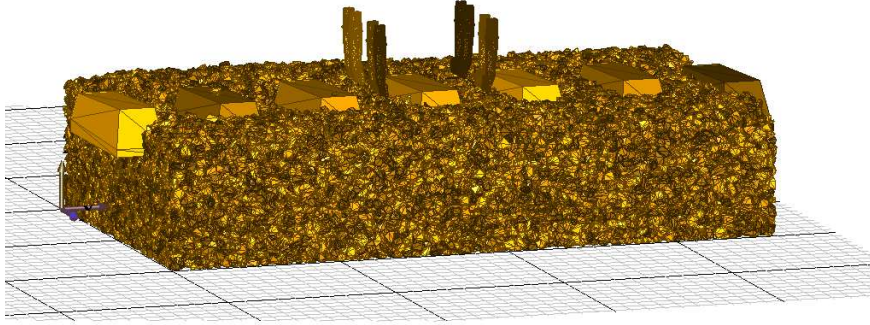
Loop on time steps
for  $i = 1, 2 \dots$  do
  Gap prediction and contact detection
  Domain partitioning
  Parallel loop on subdomains
  for  $E = 1, 2 \dots n_{SD}$  do
    Compute ‘free’ velocity at each grain of the subdomain E
  end for
  Loop on DDM iterations
  for  $j = 1, 2 \dots n_{DDM}$  do
    Parallel loop on subdomains
    for  $E = 1, 2 \dots n_{SD}$  do
      NLGS solve for each subdomain E, with  $n$  iterations
      Compute numerical residual for each subdomain E if required
    end for
    NLGS solve for ‘subdomain’  $\Gamma$ , with  $m$  iterations
    Compute numerical residual for ‘subdomain’  $\Gamma$  if required
    Convergence test
  end for
  Parallel loop on subdomains
  for  $E = 1, 2 \dots n_{SD}$  do
    Compute nodal quantities (update velocities, positions)
  end for
end for

```

---

### 3. Numerical results

The targeted application concerns the behavior of a large slice of the ballasted railway, submitted to the tamping process. The numerical simulation of such an problem requires an unaffordable computational cost. The proposed test in this section relies on the previous solving strategy to reduce the simulation cost, to be able to access extensive parametrical studies of the tamping process. A 3D specimen of size  $3.6\text{ m} \times 2\text{ m} \times 0.56\text{ m}$  is modeled as a slice of a track with 7 sleepers. It is submitted to a tamping cycle on sleeper # 4. Figure 3 depicts the geometry of this specimen.



**Figure 3.** Railway slice with 7 sleepers. The tamping cycle is operated on the sleeper # 4 (numbered from left to right)

The specimen is made of 88 100 polyhedral grains, and approximately 310 000 frictional contacts. Friction coefficient is  $\mu = 1$  between grains, between grains and sleepers, and between grains and tamping tools; it is selected to  $\mu = 0.8$  between grains and planes that define boundary conditions. The numerical parameters are selected as follows:

- The time interval  $[0, T]$  with  $T = 1.764$  s, is discretized with 8 820 time steps,
- $n_{SD} = 7$  subdomains correspond to the areas under each sleeper,
- $n_{DDM} = 740$ ,
- $n = m = 1$ .

The implementation is performed within the LMGC90 platform (Dubois *et al.*, 2007), and a 8 Gb RAM 2 Dual-Core PC is used (2 processors with 2 cores each). The detailed physical response of the problem is studied with several global quantities such as the compactness in (Hoang *et al.*, 2011b). We are more concerned herein with the numerical indicators: computing time, speedup and efficiency, interpenetrations as residuals.

To estimate the parallel efficiency of the OpenMP implementation, the speedup is defined as:  $S_p = T_1/T_p$ , where  $T_1$  ( $T_p$ ) denotes the run time with the parallel algorithm executed on 1 processor (respectively,  $p$  processors). The efficiency is defined as  $E_p = S_p/p$  and is expected to be in the interval  $[0, 1]$ .

Table 1 reports the restitution time of each run, as well as the time profile on different parts of the simulation. The numerical solve with NLGS represents on the average 75 % of total time. The contact detection phase is approximately 22 %, and the remaining parts 3 %. Up to now, only the resolution phase is parallelized. The 4 processor run is roughly twice as fast as the sequential run.

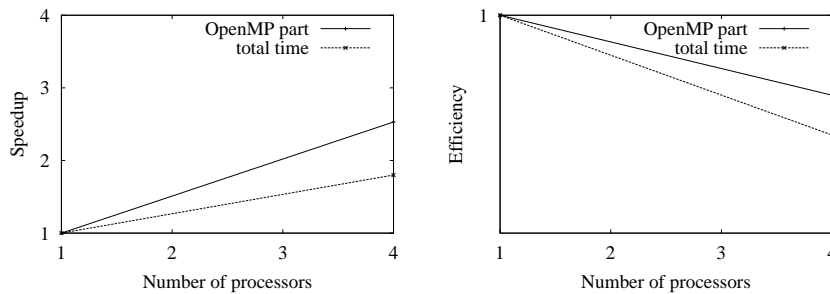


Number of processors	restitution time / min	
	1	4
Total time	14019	7860
Contact detection	2473	2458
NLGS solve	11290	5086
Other parts*	256	316
NLGS solve details		
Treatment of 7 subdomains (in parallel)	10250	4048
Treatment of the global interface (sequential)	1040	1038

\* predictions, updates...

**Table 1.** Profiling computation time, using 1 and 4 processors

The speedup and efficiency plots on Figure 4 illustrates the better performances of the ‘subdomain’ part (OpenMP part) when compared to the total restitution time (respectively 2.53 and 1.8 for the speedup, 63.3% and 45% for the efficiency). The sequential treatment of the interface, as well as the other parts of the code, impairs the parallel performances. The second source for this loss of efficiency is the load unbalance of the processors (7 subdomains of equivalent sizes for 4 processors).



**Figure 4.** Parallel implementation performance with 4 processors and 7 subdomains

#### 4. Asynchronous algorithm

In the previous algorithm, the sequential treatment of the global interface allows to synchronize all the subdomain information. In such a case, the numerical results are always identical for a given splitting of the domain, whatever the number of processors is. For a parallelization point of view, this synchronization increases idle time for the processors, and reduces the efficiency, especially when one wishes to use a large number of processors and subdomains.

Asynchronous algorithms would therefore lead to a higher parallel efficiency. They are usually iterative algorithms where the update of the components of the iterated vector solution is performed on-the-fly as soon as computed.

#### 4.1. Principles of the algorithm

The OpenMP implementation of the asynchronous version is very close to the previous one: the only modification lies in the the global interface which is dealt with as any subdomain in the DDM loop, and has no more a specific treatment. From an algebraic point of view, this is the transformation of a block-Gauss-Seidel splitting of the reference problem [2] of the form:

$$\begin{bmatrix} v_E \\ v_{E'} \\ v_\Gamma \end{bmatrix} - \begin{bmatrix} W_E^L & 0 & 0 \\ 0 & W_{E'}^L & 0 \\ W_{\Gamma E} & W_{\Gamma E'} & W_\Gamma^L \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix} = \begin{bmatrix} v_E^d \\ v_{E'}^d \\ v_\Gamma^d \end{bmatrix} + \\ + \begin{bmatrix} W_E - W_E^L & 0 & W_{E\Gamma} \\ 0 & W_{E'} - W_{E'}^L & W_{E'\Gamma} \\ 0 & 0 & W_\Gamma - W_\Gamma^L \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix}$$

(where superscript  $L$  denotes the diagonal and lower part) for the synchronous algorithm, into a block-Jacobi (for the subdomains) / Gauss-Seidel (for the inner subdomain problems) splitting:

$$\begin{bmatrix} v_E \\ v_{E'} \\ v_\Gamma \end{bmatrix} - \begin{bmatrix} W_E^L & 0 & 0 \\ 0 & W_{E'}^L & 0 \\ 0 & 0 & W_\Gamma^L \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix} = \begin{bmatrix} v_E^d \\ v_{E'}^d \\ v_\Gamma^d \end{bmatrix} + \\ + \begin{bmatrix} W_E - W_E^L & 0 & W_{E\Gamma} \\ 0 & W_{E'} - W_{E'}^L & W_{E'\Gamma} \\ W_{\Gamma E} & W_{\Gamma E'} & W_\Gamma - W_\Gamma^L \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix}$$

Following (Renouf, 2004; Chau, 2005), this type of algorithm may lead to several issues:

- Since the order of processing on each contact is continuously changing, the produced admissible solution changes for each run on the same problem. Comparison of two solutions is therefore somehow difficult. Nevertheless the macroscopic global behavior should be similar ;
- The asynchronous writing on the solution storage vector may lead to conflicts when simultaneous writings occur. In such a case, one or more computed values are discarded.

#### 4.2. Algorithms comparison

The same ballasted track testbed is used and a solution is produced with 3 versions of the code: (i) version 1 is the standard code, without domain decomposition and without parallelization; (ii) version 2 is the synchronous algorithm parallelized with OpenMP ; (iii) version 3 is the asynchronous version with the same domain decomposition, and parallelized with OpenMP. Simulations are performed on a 8 Gb Dual-Core station with 4 processors.

The corresponding test case is a smaller version of the previous one. Namely, it involves 2272 time steps for a physical duration of  $T = 1.8175$  s, 3 subdomains and a global interface are used, the average number of contacts are 9 800 and 22 800 for the two subdomains and 3 000 for the interface. For the iteration counts,  $n_{\text{DDM}} = 500$  DDM iterations and  $n_{\text{NLGS}} = n = m = 1$  NLGS iteration are prescribed.

The obtained solutions have been compared with their compactness and their inertial parameter evolutions. Though slightly different, their trends are similar. The asynchronous solution does exhibit a small overshoot for the compactness, which does not exceed 1.7 % the other ones. This allows to conclude that sound solutions are obtained in each case. Indeed, the residual for the non penetration constraint is similar, and sometimes lower, for the asynchronous case when compared to the synchronous algorithm.

Simulation costs are reported in Table 2. Even on a small number of processors, the asynchronous algorithm is more efficient than the synchronous version. The overall efficiency is nevertheless still small (1.71/3) due to a relatively small problem size. This efficiency, as well as the improvement of the asynchronous version, are expected to be improved with the simultaneous increase of the number of subdomains and problem size.

**Table 2.** Simulation time for the 3 implementation versions

	Restitution time / min		
	Version 1 Reference	Version 2 Synchronous alg.	Version 3 Asynchronous alg.
Number of processors	1	3	3
Total time	923	621	607
Contact detection	193	175	172
NLGS solve	708	424	414
Other parts*	22	22	21

\* predictions, updates...

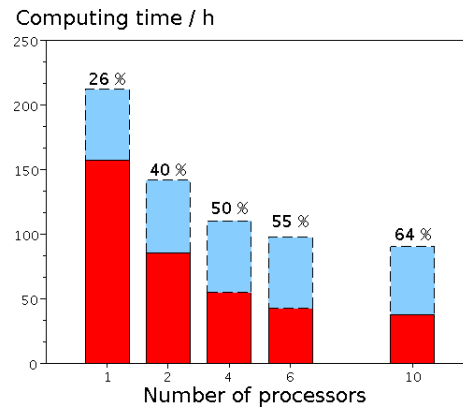
#### 5. Conclusions

The proposed approach combines a domain decomposition method with a shared-memory parallelization technique. Such a strategy allows to reduce significantly the

computational time for solving industrial problems. The use of 4 processors leads to a gain of 1659 minutes (about 4.3 days). However the parallel efficiency is weak with a speedup equal to 1.8 in comparison with the optimal speedup.

In order to optimize the parallel performance some propositions may be formulated. The balance between the processors may be easily improved by choosing a number of subdomains equal to a multiple of the number of processors. Likewise the geometrical partitioning has to be optimized to get a nearly constant number of contacts per subdomain. The size of the interface may be reduced in choosing compact subdomains. Anyway the number of subdomains is limited both by the use of a shared-memory architecture and the need to restrict the size of the interface to the size of one subdomain. Finally a substantial gain may be expected from the parallelization of the contact detection as suggested by the analysis of the computational time spent in the sequential part compared to the parallel part (Figure 5).

In spite of its simplicity, the present strategy is relevant to deal with large-scale strongly nonlinear industrial problems requiring robustness and a multiparametric study for improving maintenance processes.



**Figure 5.** Evolution of the elapsed time for the full computation and for the parallel part with respect the number of subdomains

**Acknowledgement.** The authors wish to thank Robert Perales for his contribution to the numerical results. The first author acknowledges funding from the Association Nationale de la Recherche et de la Technologie.

## 6. References

Azéma E., Etude numérique des matériaux granulaires à grains polyédriques : rhéologie quasi-statique, dynamique vibratoire, application au procédé de bourrage du ballast, Ph.D. Thesis, Université Montpellier 2, 2007.

- Azéma E., Radjaï F., Saussine G., “ Quasistatic rheology, force transmission and fabric properties of a packing of irregular polyhedral particles”, *Mechanics of Materials*, vol. 41, p. 721-741, 2009.
- Cambou B., Jean. M., *Micromécanique des matériaux granulaires*, Traité MIM – Mécanique et ingénierie des matériaux, Hermes Science Europe Ltd, 2001.
- Champaney L., Dureisseix D., *A mixed domain decomposition approach*, Mesh Partitioning Techniques and Domain Decomposition Methods, Civil-Comp Press / Saxe-Coburg Publications, chapter 12, p. 293-320, 2007.
- Chau M., Algorithmes Parallèles asynchrones pour la simulation numérique, Ph.D. Thesis, Institut National Polytechnique de Toulouse, 2005.
- Dubois F., Renouf M., “ Numerical strategies and software architecture dedicated to the modelling of dynamical systems in interaction. Application to multibody dynamics”, *Multibody 2007*, ECCOMAS Thematic Conference, Milano, Italy, june, 2007.
- Hoang T. M. P., Alart P., Dureisseix D., Saussine G., “ A Domain Decomposition method for granular dynamics using discrete elements and application to railway ballast”, *Annals of Solid and Structural Mechanics*, vol. 2, n° 2-4, p. 87-98, 2011a.
- Hoang T. M. P., Saussine G., Dureisseix D., Alart P., “ Behaviour of a portion of railway track under maintenance operation”, *9th World Congress on Railway Research – WCRR2011*, Lille, France, may, 2011b.
- Iceta D., Simulation numérique de la dynamique des systèmes discrets par décomposition de domaine et application aux milieux granulaires, Ph.D. Thesis, Université Montpellier 2, 2010.
- Iceta D., Dureisseix D., Alart P., “ Mixed versus impulse-oriented Domain Decomposition Method for granular dynamics”, *European Journal of Computational Mechanics*, vol. 18, n° 5-6, p. 429-443, 2009.
- Jean M., “ The non-smooth contact dynamics method”, *Computer Methods in Applied Mechanics and Engineering*, vol. 177, p. 235-257, 1999.
- Moreau J. J., “ Numerical aspects of sweeping process”, *Computer Methods in Applied Mechanics and Engineering*, vol. 177, p. 329-349, 1999.
- Nineb S., Alart P., Dureisseix D., “ Domain decomposition approach for nonsmooth discrete problems, example of a tensegrity structure”, *Computers and Structures*, vol. 85, n° 9, p. 499-511, 2007.
- Paderno C., Comportement du ballast sous l’action du bourrage et du trafic ferroviaire, Thèse de doctorat, EPFL, 2010.
- Perales R., Saussine G., Milesi N., Radjai F., “ Tamping Process Optimization”, *Euromech conference ESMC2009*, Lisbonne, Portugal, p. 7-11, 2009.
- Renouf M., Optimisation numérique et calcul parallele pour l’étude des milieux divisés bi- et tridimensionnels, Ph.D. Thesis, Université Montpellier 2, 2004.
- Saussine G., Contribution à la modélisation de granulats tridimensionnels : Application au ballast, Ph.D. Thesis, Université Montpellier 2, 2004.
- Solomon B., *Railway maintenance*, MBI Publishing Company, USA, 2001.

**Ri<sub>2</sub><sup>1</sup> dacteur responsable : GUILLAUME LAURENT**

**SERVICE İ<sub>2</sub><sup>1</sup> DITORIAL –  
HERMES-LAVOISIER  
14 rue de Provigny, F-94236 Cachan cedex  
Tİ<sub>2</sub><sup>1</sup>l. : 01-47-40-67-67  
E-mail : [revues@lavoisier.fr](mailto:revues@lavoisier.fr)  
Serveur web : <http://www.revuesonline.com>**

**ANNEXE POUR LE SERVICE FABRICATION  
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER  
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER  
LE FICHER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL**

**1. ARTICLE POUR LA REVUE :**

*European Journal of Computational Mechanics. Volume 0 – n 0/2011*

**2. AUTEURS :**

*T. M. P. Hoang<sup>\*,\*\*\*</sup> — G. Saussine<sup>\*</sup> — D. Dureisseix<sup>\*\*</sup> — P. Alart<sup>\*\*\*</sup>*

**3. TITRE DE L'ARTICLE :**

*Domain decomposition with discrete element simulations using shared-memory parallel computing for railways applications*

**4. TITRE ABRÉGIÉ  $\frac{1}{2}$  GÏ  $\frac{1}{2}$  POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :**

*DDM for DEM parallel simulations*

**5. DATE DE CETTE VERSION :**

*December 5, 2011*

**6. COORDONNÉES  $\frac{1}{2}$  DES AUTEURS :**

– adresse postale :

<sup>\*</sup> Société Nationale des Chemins de Fer (SNCF), Innovation and Research Division, 45 rue de Londres, F-75379 Paris Cedex 08

{thi-minh-phuong.hoang, gilles.saussine}@sncf.fr

<sup>\*\*</sup> Laboratoire de Mécanique des Contacts et des Structures (LaM-CoS), INSA Lyon / CNRS UMR 5259, 18-20 rue des Sciences, F-69621 Villeurbanne Cedex

david.dureisseix@insa-lyon.fr

<sup>\*\*\*</sup> Laboratoire de Mécanique et Génie Civil (LMGC), Université Montpellier 2 / CNRS UMR 5508, CC048, Place Eugène Bataillon, F-34095 Montpellier Cedex 5

pierre.alart@univ-montp2.fr

– tél  $\frac{1}{2}$  li  $\frac{1}{2}$  phone : 00 00 00 00 00

– tél  $\frac{1}{2}$  li  $\frac{1}{2}$  copie : 00 00 00 00 00

– e-mail : guillaume.laurent@ens2m.fr

**7. LOGICIEL UTILISÉ  $\frac{1}{2}$  POUR LA PRÉ  $\frac{1}{2}$  PARATION DE CET ARTICLE :**

**L<sup>A</sup>T<sub>E</sub>X**, avec le fichier de style article-hermes.cls,  
version 1.23 du 17/11/2005.

**8. FORMULAIRE DE COPYRIGHT :**

Retourner le formulaire de copyright signé  $\frac{1}{2}$  par les auteurs,  
ü  $\frac{1}{2}$  li  $\frac{1}{2}$  charg  $\frac{1}{2}$  sur : <http://www.revuesonline.com>