European Journal of Computational Mechanics
Vol. 21, Nos. 3–6, June–December 2012, 242–253

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

T.M.P. Hoang[a,b]*, G. Saussine[a], D. Dureisseix[c] and P. Alart[b]

[a]*Société Nationale des Chemins de Fer (SNCF), Innovation and Research Division, 45 rue de Londres, Paris Cedex 08 F-75379, France;* [b]*Laboratoire de Mécanique et Génie Civil (LMGC), Université Montpellier 2/CNRS UMR 5508, CC048, place Eugène Bataillon, Montpellier Cedex 5 F-34095, France;* [c]*Laboratoire de Mécanique des Contacts et des Structures (LaMCoS), INSA Lyon/CNRS UMR 5259, 18–20 rue des Sciences, Villeurbanne Cedex F-69621, France*

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 behaviour. To reduce computational costs, we study the use of two strategies: domain decomposition methods and shared-memory parallelisation with OpenMP. An example of a maintenance process, the tamping, on a portion of railway track with seven sleepers, is simulated.

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; parallelisation; LMGC90; ballast; maintenance

Mots-clés: non smooth contact dynamics; parallélisation; LMGC90; ballast; maintenance

1. Motivations: tamping process optimisation

The degradation of ballasted railway under commercial exploitation requires frequent and costly maintenance operations. One of these operations is the tamping process (Azéma, 2007; Azéma, Radjaï, & Saussine, 2009; Paderno, 2010; 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 behaviour of the ballast during tamping is necessary to propose optimisations of this maintenance process (Perales, Saussine, Milesi, & Radjai, 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,

*Corresponding author. Email: tmp.hoang@gmail.com

ISSN 1779-7179 print/ISSN 1958-5829 online

http://dx.doi.org/10.1080/17797179.2012.714723
http://www.tandfonline.com

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. A domain decomposition method (DDM) coupled with a shared-memory parallelisation 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 seven 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 NSCD approach developed by Moreau and Jean (Cambou & Jean, 2001; Jean, 1999; Moreau, 1999) 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 NCSD 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 non-smooth 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,

$$M(V - V^i) = R^d + R \tag{1}$$

where M denotes the mass and inertia matrix. V^i and V are the generalised velocities of the grains at the instants t_i and t_{i+1} (the subscript $i+1$ is omitted for the following). They contain the translational velocities and the rotation rates in the inertia basis of the grains. R^d represents the external impulsions and R are the impulsions 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 impulsion r. The local variables are defined according to a local-to-global (or contact-to-grain) mapping H and its transpose H^{T}: $\mathrm{v} = H^{\mathrm{T}}\ V$ and $R = Hr$, (Visseq et al., 2012). Indeed, for the two-grain interaction of the 2D case in Figure 1, and for the sole resultant to simplify, H contains the local basis vectors as:

$$H = \left[\underline{x}_i \quad \underline{y}_i \quad -\underline{x}_j \quad -\underline{y}_j \right]^{\mathrm{T}} [\underline{n} \quad \underline{t}]$$

i.e. the coordinates of the local basis vectors in the grain coordinate basis.

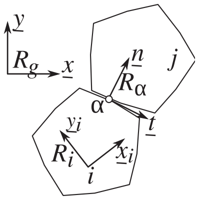

Figure 1. Coordinate basis; R_g: global coordinate basis, R_i: local (related to the grain i) coordinate basis (for grain dynamics) and R_α: local (related to an interaction α between two grains) coordinate basis (for interactions). Details of the local contact frame $(\underline{n}, \underline{t})$ at a contact α between two touching particles i and j.

The dynamics is then given by $v = v^d + Wr$, where $v^d = H^{\mathrm{T}}(M^{-1}R^d + V^i)$, and $W = H^{\mathrm{T}}M^{-1}H$ is the so-called Delassus operator.

Once the velocities of each grain are determined, the position update is performed (the translations are determined with a θ-method integration scheme, while the rigid body finite rotations are determined with the Hughes–Winget scheme).

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^α and the normal contact impulsion r_n^α, once the penetration tested with a predicted gap g^α, for the contact α. The frictional contact law is then,

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

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

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

$$\begin{cases} v = v^d + Wr \\ \mathcal{R}(r, v) = 0 \end{cases}$$

2.1.3. *Non-linear Gauss–Seidel solver*

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

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

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) as shown in Figure 2. Then, two partitioning techniques may be performed:

- *The primal approach* distributes the grains into the subdomains (Figure 3 left). A grain belongs to a subdomain if the coordinates of its mass centre is inside a geometrical box previously defined by a regular grid. The interface consists then of *links* between grains of two neighbouring subdomains (Hoang, Alart, Dureisseix, & Saussine, 2011a, 2011b).
- *The dual approach* distributes the links into the subdomains (Figure 3, 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, Dureisseix, & Alart, 2009).

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 non-smooth solver. This appears clearly when ones examines the algebraic partitioning deriving from the geometric substructuring.

2.2.2. Algebraic partitioning

According to the notations developed in Hoang et al. (2011a), the subscripts E, E' and Γ denote the variables of the subdomain E and E' and of the interface Γ, respectively. For the sake of simplicity, only two subdomains are considered. The uppercase letters refer to

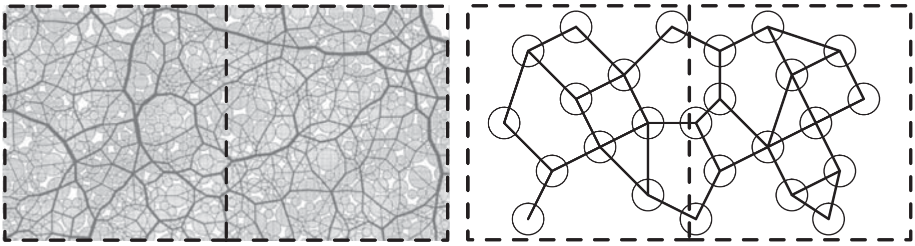

Figure 2. Substructuring with a box-type method (a granular example and the schematic principle).

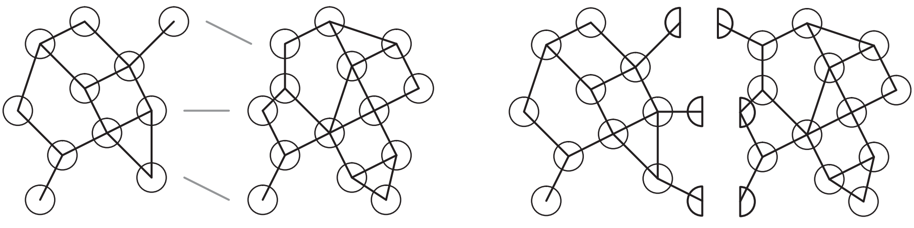

Figure 3. Primal vs. dual approaches.

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 other hand, the quantities related to the interactions between the grains are split as follows: the interactions between two grains of the same subdomain E are tagged with an index E; the interaction between two grains of different subdomains belongs to the interface and are tagged with the index Γ. So, 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 Γ applied to the subdomain E,

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

Indeed, with the previous splitting and the topology of the domain decomposition, one can check that the mapping H also splits in the following form:

$$R = Hr = \begin{bmatrix} R_E \\ R_{E\Gamma} \\ R_{E'} \\ R_{E'\Gamma} \end{bmatrix} = \begin{bmatrix} H_E & 0 & 0 \\ 0 & 0 & H_{E\Gamma} \\ 0 & H_{E'} & 0 \\ 0 & 0 & H_{E'\Gamma} \end{bmatrix} \begin{bmatrix} r_E \\ r_{E'} \\ r_\Gamma \end{bmatrix}$$

The dual relations provide the relative velocities starting from the generalised 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 partitioned system,

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

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 NSCD is a NLGS 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 NLGS method, a DDM 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 and the number of NLGS iterations in the interface m. The number of DDM iterations n_{DDM} may be replaced by a convergence test in a generic version. But as the evolution of this number is too erratic if it is controlled by a convergence test and as a performance analysis is consequently difficult, a high enough fixed n_{DDM} is preferred in this study. The numbers n and m are fixed to 1 in the following examples because a previous study (Hoang et al., 2011a) proves that it is the optimal values for convergence. That corresponds to a standard sequential NLGS algorithm with a special renumbering of the contacts which are dealt first in the subdomains, then on the interface. The emphasised parts (slanted font) are additional tasks to allow the DDM and parallel treatment.

Algorithm 1. DDM–OpenMP–NLGS

```
for $i = 1, 2 \ldots$ do {Loop on time steps}
  Gap prediction and contact detection
  Domain partitioning
  for $E = 1, 2 \ldots n_{\mathrm{SD}}$ do {Parallel loop on subdomains}
    Compute 'free' velocity at each grain of the subdomain E
  end for
  for $j = 1, 2 \ldots n_{\mathrm{DDM}}$ do {Loop on DDM iterations}
    for $E = 1, 2 \ldots n_{\mathrm{SD}}$ do {Parallel loop on subdomains}
      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
  for $E = 1, 2 \ldots n$SD do {Parallel loop on subdomains}
    Compute nodal quantities (update velocities, positions)
  end for
end for
```

3. Numerical results

The targeted application concerns the behaviour of a large slice of the ballasted railway, submitted to the tamping process. The numerical simulation of such a problem requires an unaffordable computational cost. The proposed test in this section relies on the previous solving

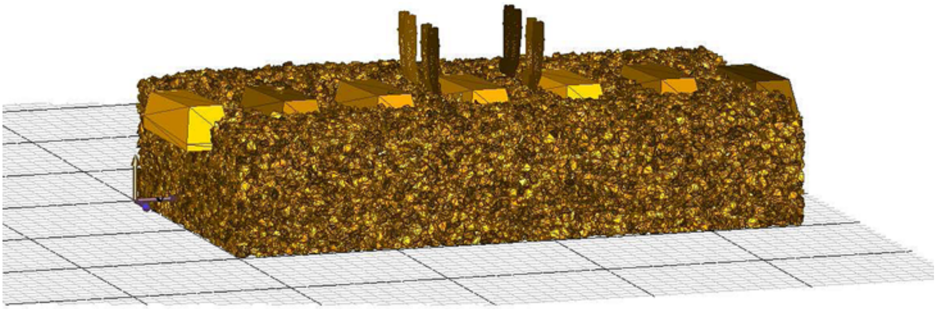

Figure 4. Railway slice with seven sleepers. The tamping cycle is operated on the sleeper # 4 (numbered from left to right).

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 m × 2 m × 0.56 m is modelled as a slice of a track with seven sleepers. It is submitted to a tamping cycle on sleeper # 4. Figure 4 depicts the geometry of this specimen.

The specimen is made of 88,100 polyhedral grains, and approximatively 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 discretised with 8820 time steps,
- $n_{\mathrm{SD}} = 7$ subdomains correspond to the areas under each sleeper,
- $n_{\mathrm{DDM}} = 740$ and
- $n = m = 1$.

The implementation is performed within the LMGC90 platform (Dubois & Renouf, 2007), and a 8 GB RAM 2 dual-core PC is used (two processors with two 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. Indeed, since the granular problems have multi-valued solutions, comparisons of one solution produced by one algorithm to an other one can only rely on the comparisons of global physical parameters, such as compactness. Nevertheless, the accuracy and the physical feasibility of a solution is ensured with low values of numerical residuals, such as interpenetrations (Hoang et al., 2011a). These interpenetrations are also used to fix empirically the number of time steps. Indeed, we have not a relevant stability condition because no elasticity is assumed in the contacts between perfectly rigid grains.

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 core (respectively, p cores). 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 average 75% of total time. The contact detection phase is approximatively 22% and the remaining parts 3%. Up to now, only the resolution phase if parallelised. The four-core run is roughly twice as fast as the sequential run.

Table 1. Profiling computation time, using one and four cores.

	Restitution time/min	
Number of cores	1	4
Total time	14,019	7860
Contact detection	2473	2458
NLGS solve	11,290	5086
Other parts[a]	256	316
NLGS solve details		
Treatment of seven subdomains (in parallel)	10,250	4048
Treatment of the global interface (sequential)	1040	1038

[a]Predictions, updates …

Table 2. Parallel implementation performance with four cores and seven subdomains.

	OpenMP part		Full simulation	
Number of cores	Speedup	Efficiency (%)	Speedup	Efficiency (%)
1	1	100	1	100
4	2.53	63.3	1.78	44.5

The speedup and efficiency shown in Table 2 illustrates the better performances of the ‘subdomain’ part (OpenMP part) when compared to the total restitution time. 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 cores (seven subdomains of equivalent sizes for four cores).

4. Asynchronous algorithm

In the previous algorithm, the sequential treatment of the global interface allows to synchronise 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 cores is. For a parallelisation point of view, this synchronisation increases idle time for the cores and reduces the efficiency, especially when one wishes to use a large number of cores 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 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 (Chau, 2005; Renouf, 2004), 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 behaviour should be similar.
- The asynchronous writing on the solution storage vector may lead to conflicts when simultaneous writings occur. In such a case, the data conflict management in OpenMP is used, and the more recently computed value is conserved, the previous ones are overwritten.

4.2. Algorithms comparison

The same ballasted track testbed is used and a solution is produced with three versions of the code: (i) Version 1 is the standard code, without domain decomposition and without parallelisation; (ii) Version 2 is the synchronous algorithm parallelised with OpenMP and (iii) Version 3 is the asynchronous version with the same domain decomposition, and parallelised with OpenMP. Simulations are performed on a 8 GB Dual-Core station with four cores.

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, three subdomains and a global interface are used, the average number of contacts are 9800 and 22,800 for the two subdomains and 3000 for the interface. For the iteration counts, $n_{\mathrm{DDM}} = 500$ DDM iterations and $n_{\mathrm{NLGS}} = n = m = 1$ 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% of the other ones. This allows us to conclude that sound solutions are obtained in each case.

Table 3. Simulation time for the three implementation versions.

	Restitution (time/min)		
	Version 1 reference	Version 2 synchronous alg.	Version 3 asynchronous alg.
Number of cores	1	3	3
Total time	923	621	607
Contact detection	193	175	172
NLGS solve	708	424	414
Other parts[a]	22	22	21

[a] Predictions, updates …

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 3. Even on a small number of cores, the asynchronous algorithm is more efficient than the synchronous version. The overall efficiency is nevertheless still small (1.71/3); this is due first to the small number of cores and subdomains for a partitioning which is quite balanced, and second, to a relatively small problem size. Indeed, the efficiency grows with the granularity, i.e. the ratio of the parallelised computations (linked to the size of the subdomains) to the volume of data exchange between the cores (linked to the size of the interface). This efficiency, as well as the improvement of the asynchronous version, are therefore expected to be improved with the simultaneous increase of the number of subdomains and problem size.

5. Conclusions

The proposed approach combines a DDM with a shared-memory parallelisation technique. Such a strategy allows to reduce significantly the computational time for solving industrial

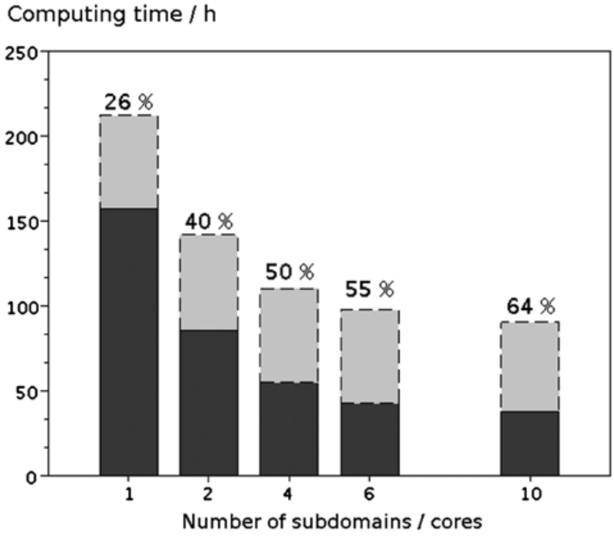

Figure 5. Evolution of the elapsed time with respect the number of subdomains. Grey rectangles: sequential part, black rectangles: parallel part and sum of the two: full computation time. Percentage value is the sequential-to-total time ratio.

problems. The use of four cores leads to a gain of 6159 min (about 4.3 days). However, the parallel efficiency is weak with a speedup equal to 1.78 in comparison with the optimal speedup.

In order to optimise the parallel performance, some propositions may be formulated. The balance between the cores may be easily improved by choosing a number of subdomains equal to a multiple of the number of cores. Likewise, the geometrical partitioning has to be optimised 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 parallelisation 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 non-linear industrial problems requiring robustness and a multi-parametric study for improving maintenance processes.

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 (ANRT).

References

Azéma, E. (2007). *Etude numérique des matériaux granulaires à grains polyédriques: rhéologie quasi-statique, dynamique vibratoire, application au procédé de bourrage du ballast* (*Numerical study of granular material composed by polyedric grains : quasi-static rheology, vibrationnal dynamic and application to tamping process*) (PhD Thesis). Université Montpellier 2.

Azéma, E., Radjaï, F., & Saussine, G. (2009). Quasistatic rheology, force transmission and fabric properties of a packing of irregular polyhedral particles. *Mechanics of Materials, 41*, 721–741.

Cambou, B., & Jean, M. (2001). *Micromécanique des matériaux granulaires*. (*Micro-mechanical of granular materials*) Traité MIM – Mécanique et ingénierie des matériaux. Hermes Science Europe Ltd.

Chau, M. (2005). *Algorithmes Parallèles asynchrones pour la simulation numérique (Parallel asynchronous algorithm for numerical simulation)* (PhD Thesis). Institut National Polytechnique de Toulouse.

Dubois, F., & Renouf, M. (2007, June). *Numerical strategies and software architecture dedicated to the modelling of dynamical systems in interaction. Application to multibody dynamics*. In Multibody 2007, ECCOMAS Thematic Conference, Milano, Italy.

Hoang, T.M.P., Alart, P., Dureisseix, D., & Saussine, G. (2011). A domain decomposition method for granular dynamics using discrete elements and application to railway ballast. *Annals of Solid and Structural Mechanics, 2*(2–4), 87–98.

Hoang, T.M.P., Saussine, G., Dureisseix, D., & Alart, P. (2011, May). Behaviour of a portion of railway track under maintenance operation. In *9th World Congress on Railway Research* – WCRR2011, Lille, France.

Iceta, D. (2010). *Simulation numérique de la dynamique des systèmes discrets par décomposition de domaine et application aux milieux granulaires* (*Numerical simulation of the dynamic of discrete systems using domain decomposition. Granular media application*) (PhD Thesis). Université Montpellier 2.

Iceta, D., Dureisseix, D., & Alart, P. (2009). Mixed versus impulse-oriented domain decomposition method for granular dynamics. *European Journal of Computational Mechanics, 18*(5–6), 429–443.

Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering, 177*, 235–257.

Moreau, J.J. (1999). Numerical aspects of sweeping process. *Computer Methods in Applied Mechanics and Engineering, 177*, 329–349.

Paderno, C. (2010). *Comportement du ballast sous l'action du bourrage et du trafic ferroviaire* (*Ballast behaviour under tamping process and railway traffic*) (Thèse de doctorat). EPFL.

Perales, R., Saussine, G., Milesi, N., & Radjai, F. (2009). *Tamping process optimization*. In Euromech conference ESMC2009 (pp. 7–11). Lisbonne, Portugal.

Renouf, M. (2004). *Optimisation numérique et calcul parallele pour l'étude des milieux divisés bi- et tridimensionnels* (*Numerical Optimisation and Parallel Computing applied to the simulation of 2D/3D discrete element*) (PhD Thesis). Université Montpellier 2.

Saussine, G. (2004) *Contribution* à la *modélisation de granulats tridimensionnels: Application au ballast* (*Contribution for the modelling of three-dimensional granular matter : study of railway ballast*) (PhD Thesis). Université Montpellier 2.

Solomon, B. (2001). *Railway maintenance*. New York: MBI.

Visseq, V., Martin, A., Iceta, D., Azema, E., Dureisseix, D., & Alart, P. (2012). Dense granular dynamics analysis by a domain decomposition approach. *Computational Mechanics, 49*(6), 709–723.