Contact on multiprocessor environment: from multicontact problems to multiscale approach

Pierre Alart

▶ **To cite this version:**

Pierre Alart. Contact on multiprocessor environment: from multicontact problems to multiscale approach. P. Wriggers & T.A. Laursen. Computational contact mechanics, 498, Springer, pp.163-217, 2007, CISM Courses and Lectures, n°498, 10.1007/978-3-211-77298-0_5 . hal-00574731

HAL Id: hal-00574731
https://hal.science/hal-00574731v1

Submitted on 21 Oct 2024

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.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Contact on Multiprocessor Environment: from Multicontact Problems to Multiscale Approaches

Pierre Alart[*‡]

[*] Laboratory of Mechanical and Civil Engineering, University of Montpellier 2, Montpellier, France

[‡] CNRS, National Center of Scientific Research, France

Abstract This course is devoted to the recent developments in the numerical treatment of large multicontact problems requiring multiprocessor computers to get admissible computer time simulations. Contact conditions lead to non smooth mathematical formulations of steady-state and dynamical problems arising from structural and granular mechanics. Specific solvers, as the Non Linear Gauss Seidel algorithm and the Conjugate Projected Gradient method, have been developed and may be adapted to a parallel treatment. The domain decomposition methods allow to deal with large-scale mechanical problems and take advantage of the multiprocessor architecture of powerful computers. Their efficiency is proved for linear problems. Two different strategy for inserting the contact treatment are detailled and compared: the Newton-Schur approach and the FETI-C method. A multiscale description is finally coupled with a substructuring technique to tackle multicontact problems with diffuse non smoothness.

1 When is it relevant to use a multiprocessor environment

1.1 Large scale contact/multicontact problems

A mechanical problem must be large enough to require a multiprocessor environment. "Large" means that the number of variables to handle is very important. When these variables are independent between them, a simple distribution of the computations on different processors is efficient; that means that the problem is easily split into many independent subproblems. When the variables are coupled by the equations, the problem may still be split into subproblems, but the crucial point for efficiency of parallel treatment is the exchange of informations between them, that is the connections between the processors and the associated memories if the memory is distributed.

Parallel computing strategies have been used in structural mechanics to tackle highly heterogenous structures as fiber-reinforced composite (Ladevèze et al. (2001)) or steel-elastomer composite structure (Lene and Rey (2001)). At these scales it is not easy to distinguish between structure and material. Material science resorts also to parallel computing for composites with complex behaviour components (Feyel and Chaboche (2000)).

In structural and material mechanics using parallel computing, the contact conditions may be involved in different ways. When the contact occurs on the boundary of a large scale domain previously dealt with a multiprocessor technique, no specific parallel strategy is to develop. For instance the contact may be treated in a outer iteration loop in a sequential way, an inner iteration loop being used for solving the material problem requiring a parallel technique. Low level modifications are necessary for such a weak coupling.

When the contact occurs in many different areas, we refer to *multicontact* problems. Then it may be interesting to define a specific multiprocessor strategy to couple the contact conditions with the other equations. Ladevèze et al. (2002) verify their strategy on a micro-cracked medium. Alart et al. (2004) use a multiprocessor approach to tackle cellular materials submitted to large deformations until selfcontact. Champaney et al. (1997) present several applications concerning assemblies of pieces.

When the contact is the main constitutive law of the system, we refer to *diffuse non smoothness* through the whole structure or domain. The typical example is the granular medium viewed as a collection of rigid bodies in interaction by contact and friction. But the tensegrity structures may be considered as diffuse non smooth systems and are studied specifically in this lecture. Granular media are often large scale system with several millions grains and have complex behaviours from a solid state to a dilute gas via a fluid flow. The simulation of such evolutive problems requires parallel techniques to be performed in a reasonable computing time. Several stages of the simulation software may benefit from a parallel treatment, the equilibrium or motion equations, but also the contact detection. A study on the time consuming of the different stages is presented in the next subsection. The two following subsections attempt to classify some multicontact problems according to theirs mathematical features. The last subsection is devoted to a general presentation of a range of discrete non smooth systems.

The second section is dedicated to the adaptation of two classical contact solvers to the parallel computing. In the third section the extension of the domain decomposition approaches to multicontact problems is presented. The fourth section closes the course by a presentation of a domain decomposition method for diffuse non smooth systems with a multiscale enrichment.

1.2 CPU time consuming for granular systems

A recent study has identified the main CPU time consuming parts of a code devoted to the simulation of the behaviour of granular systems (Renouf et al. (2004)). Three computational efforts are evaluated: the contact detection, the contact solver and the convergence test which should be inserted in the previous solver part. But this identification strongly depends on geometry and intrinsic properties of the sample, and we show a relationship between the CPU time consuming rates of the different parts of the code and the mechanical properties of the sample. The granular medium can be considered as a gaz (mixing), as a liquid (avalanche, rotative drum, granular flows) or a solid

(quasi-static evolution, compaction, shear test) according to the process. Three different examples have been chosen to illustrate the main fields of application such as a mixing, a free surface compaction and a rotative drum (see Figure 1). Each simulation takes into account one thousand "poly-disperse" disks, with elastic shocks for mixing and inelastic shocks for the other simulations (Cambou and Jean (2001)).

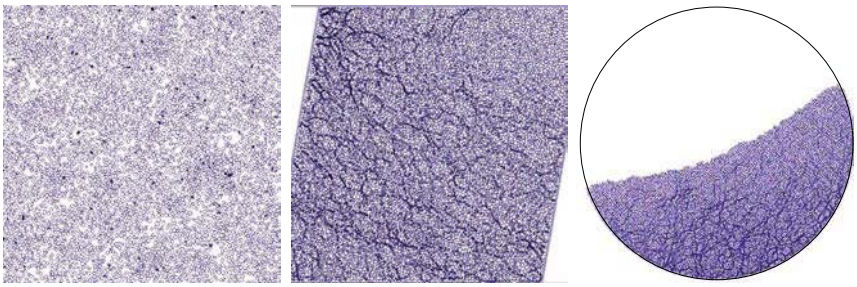

Figure 1. Three granular simulations: mixing, compaction, drum.

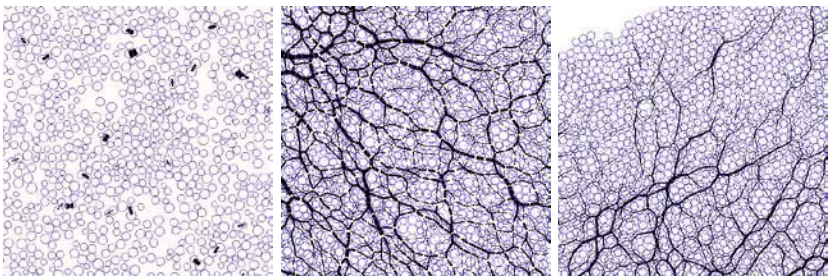

Figure 2. Force networks in three different simulations: mixing, compaction, drum.

Figure 2 shows the contact network in each case. This network does not exist in the mixing case, because of the permanent turbulence of the material: each particle moves in ballistic flight between two impacts. The contact is essentially characterized by binary collisions which do not interact between them. An implicit integration scheme leads to a single iteration to solve the non smooth solver like in an explicit scheme. In the two other cases which involve dense materials, the force network takes up the whole domain with different intensities due to the gravity and the dynamical solicitation. The drum has the more complex behaviour; the bottom of the drum behaves like a solid with stable force network whereas the top flows out as a fluid with regular avalanches. The force network changes continuously with a week coupling between the contacts. Such a process with such an intermediate behaviour between fluid and solid is intensively studied by physicists and mechanicians of granular media (Rajchenbach (2000), Rajchenbach (1990), Bonamy et al. (2002), Renouf et al. (2005)).

Table 1. Repartition of elapsed time taken by subroutine (%).

Problem	solver	convergence	detection
mixing	18.6%	2.9%	47.44%
compaction	84.68%	2.43%	5.82%
drum	85.68%	2.52%	1.89%

For each simulation we identify the CPU time consumed by the *detection* of the contacts and the solution of the contact conditions (determination of impulses and relative velocities satisfying the contact laws). The solution subroutine is itself split into the *solver* part and the *convergence* test. The percentage of elapsed time given in Table 1 confirms this argument: the solver of the non linear contact equations consumes the major part of the CPU time for the two last examples, although the detection of the pairs is the most expensive for the mixing case.

However the contact detection is a present research field for the discrete element approaches (Liu and Lemos (2001)), specially when the geometry becomes complex (Nezami et al. (2004), Song et al. (2006)), as well as for the finite element methods (Li et al. (2001)). Such a research is crucial for fast dynamical applications using explicit integration schemes. But we do not tackle this topic in this lecture.

If we want to deal with a wide range of applications from the equilibrium of a structure with contact to a strongly shaken granular system, the main effort from a computational point of view concerns the solver of a large number of non smooth coupled equations.

1.3 Mathematical formulations of multicontact problems

The situations where the contact occurs are very diverse and concern many fields in engineering science: metal forming, damage in materials, mechanism, crash, robotics, ... This fact explains in part the variety of the numerical algorithms to solve the relations modelling the contact conditions. According to the context in which the numerical tools are developed the formulation itself of the contact interactions may be very different. At first the contact involves inequalities instead of equalities more suitable for computation. Concise formulations useful for the mathematical analysis refer to the Convex Analysis, specially to the subdifferential calculus of convex functions. The reader is referred to some books or papers: Moreau (1974), Kikuchi and Oden (1982), Klarbring (1986), Wriggers (1995), Moreau (1994), Glocker and Pfeiffer (1996) and Brogliato et al. (2002). Such a mathematical background may be not known to deal with contact in numerical softwares, but this leads to very specific numerical procedures strongly dependant of the problem.

In this section we focus our attention on two extreme typical problems involving both unilateral constraints but leading to different mathematical formulations, themselves leading to adopt different efficient numerical strategies, specially in a multiprocessor

context. The first one involves a deformable structure and refers to a structural type problem and the second concerns a collection of rigid bodies and is granular type.

Static structural problem. The so called structural type problem consists in determining the equilibrium of an elastic body in grazing contact with a rigid foundation. As a first characteristic of this case the contact operates as a *boundary condition*. We consider directly the discrete case after applying a classical finite element method: n_n notes the total number of nodes, n_c the number of potential contact nodes. The equilibrium equations consist of a linear system (in a first approximation): stiffness matrix $\mathbb{K}$, generalized displacement $\mathbf{u}$, contact reactions $\mathbf{r}$ on the nodes of the contact area, the external forces $\mathbf{f}$,

$$\mathbb{K}\mathbf{u} - \mathbb{H}\mathbf{r} = \mathbf{f}. \tag{1.1}$$

The matrix $\mathbb{H}$ passes from the local frames to the global one for all potential contact nodes. We have to add the contact conditions written here with two inequalities and a complementarity condition between the contact reactions and the normal components of the displacements on the potential contact surface.

$$\mathbb{H}^t\mathbf{u} \geq 0,\ \mathbf{r} \geq 0,\ \mathbf{r}.\mathbb{H}^t\mathbf{u} = 0 \quad \Leftrightarrow \quad 0 \leq \mathbb{H}^t\mathbf{u} \perp \mathbf{r} \geq 0. \tag{1.2}$$

Frictional conditions may complete this system without changing the main features of the problem except the non symmetry which may be induced by a Coulomb type law. So the number of contacts n_c is small in comparison with the total number of nodes noted n_n: $\mathbb{K}$ is a $2n_n \times 2n_n$ or $3n_n \times 3n_n$ matrix according to the bidimensional or three-dimensional modelling; $\mathbb{H}$ is a $2n_n \times n_c$ or $3n_n \times n_c$ matrix. Finally the *stiffness matrix is not easily invertible* and it is convenient to conserve the global variable $\mathbf{u}$ as a main unknown of the problem. All the more so since *other non linearities* in addition to contact may occur in such structural problem: the large strains, in cellular materials for instance, or a non linear behaviour, as the plasticity in metal forming. To solve the system (1.1) (1.2) with eventual other non linearities, a Newton like method appears as a general non linear solver even if it has to be extended to non differentiable equations (Alart and Curnier (1991), Alart (1997)).

Dynamical granular problem. The situation is quite different for the granular type problem. Following the approach of Moreau (1998), the *main object of the computation is the velocity* and a time stepping method is used as time integrator. The reader is referred to Moreau (1999) and Brogliato et al. (2002) for a discussion about the different integrators. For comparison with the previous case we consider the problem on a single time step; we have to predict the velocity distribution in a collection of rigid bodies at the end of this step. The contact operates now as an *interaction law* between particles. The dynamical equations may be easily written at the contact points using the local frame (see below) by considering *the local variables*: the relative velocity $\mathbf{v}$ between the two particles passing at the contact point at this time and the impulsion $\mathbf{r}$. The system to solve is then the dynamics reduced to contacts,

$$\mathbb{W}\mathbf{r} - \mathbf{v} = \mathbf{b}. \tag{1.3}$$

The right-hand side of (1.3) takes into account the external forces and impulsions and the relative velocity at the end of the previous step. The contact condition may be written using the velocities instead of the displacements,

$$\mathbf{r} \geq 0,\ \mathbf{v} \geq 0,\ \mathbf{r}.\mathbf{v} = 0 \quad \Leftrightarrow \quad 0 \ \leq \ \mathbf{r} \perp \mathbf{v} \ \geq 0. \tag{1.4}$$

The unilateral contact conditions (1.4) may be replaced by more general interaction laws involving only local variables, indiced by α from 1 to the number of contacts n_c,

$$law_\alpha[\mathbf{v}_\alpha, \mathbf{r}_\alpha] = true, \alpha = 1, n_c. \tag{1.5}$$

In dense granulates, *the number of contacts is larger than the number of bodies*. Since *the mass matrix is diagonal*, it is easily invertible and the reduced system with the Delassus matrix $\mathbb{W}$ may be considered to be solved, $\mathbb{W} = \mathbb{H}^t\mathbb{M}^{-1}\mathbb{H}$. But the $\mathbb{W}$ matrix may be singular giving rise to "wedging" effects and indeterminacy of the impulsions. It is then convenient to handle the local variables and to carry out an iteration technique which consists in treating a succession of single-contact problems, according to the approach of Moreau (1999). Such a procedure may be interpreted as a Gauss-Seidel method; some convergence results may be obtained in special situations (Jourdan et al. (1998)). The denomination of "non smooth contact dynamics" (NSCD) algorithm is used in order to be distinguished from the "molecular dynamics" method based on a smoothing of the contact conditions (Cundall (1971), Drake and Walton (1995)).

The two problems presented above have specific features and led us to mention two very different algorithms to solve similar contact equations : a Newton like method for structural type problems and a Gauss-Seidel like one for granular type ones. Other methods can be used.

The structural type problem benefits from the techniques developed in the domain of finite elements and specially to solve linear problems. The Newton method consists of a succession of linear systems which may be solved by a domain decomposition method (DDM) well suited to parallel architecture computers. Besides most papers in the literature use the domain decomposition method for introducing a parallel treatment in contact computational mechanics; in these cases the DDM linear solver is extended to the (non linear, non smooth) contact conditions using a dual approach (Rebel et al. (2000), Dostal et al. (1998), Dostal et al. (2000), Dureisseix and Farhat (2001)) or a mixed one (Champaney et al. (1999), Ladevèze et al. (2002)). The section 3 is dedicated to the presentation of these approaches.

On the contrary we cannot easily extend some well tried techniques to the treatment of a granular type problem. Some numerical experiments based on a domain decomposition strategy have been performed leading to difficult load balancing procedures in a dynamical process (Breitkopf and Jean (1999), Owen D.R.J. and Peric (2000)). Moreover for granular type problems the nonsmoothness is not only located in some interfaces but is diffuse through the whole domain. However a domain decomposition method is proposed in the section 4 related to a multiscale approach. But a first attempt, developped in the section 2, consists in carrying out directly a parallel treatment of the non smooth solvers of the system (1.3) (1.5).

To illustrate the concepts presented above we develop in the next section a general discrete mechanical model which allows to pass continuously from a structural like problem to a granular like one.

Table 2. General notations for a system of bars and cables.

b, c	Bar and cable subscripts
τ^0, τ_b^0, τ_c^0 such that $B^t\tau^0 = 0$	Self balanced prestress
e^0, e_b^0, e_c^0	Related prestrain
τ, τ_b, τ_c	Internal tensions, in bars, in cables
$e = Bu$, $e_b = B_b u$, $e_c = B_c u$	Length variations (strain admissibility)
$\eta = -\dot{e} = -B\dot{u} = -Bv$	Relative velocities (strain rate admissibility)
$\pi = \int_{t^-}^{t^+} \tau d\nu$	Average impulsions
k_b, k_c	Local stiffnesses (in tension for cables)
$\lambda_c = -(e_c + e_c^0) + k_c^{-1}\tau_c$	Corrected length variations in cables
$\lambda_c^+ = \eta_c^+ + k_c^{-1}h^{-2}\pi_c - h^{-1}(e_c^- + e_c^0)$	Corrected relative velocities
F, F^d	Internal and external nodal forces
u, U^d	Unknown and prescribed nodal displacements
B^t, B	Link to node and node to link mappings
$K_b = B_b^t k_b B_b$, $K_c = B_c^t k_c B_c$	Bar network and cable network stiffnesses
$K = K_b + K_c$	Global stiffness
M	Mass matrix
$W = BM^{-1}B^t = \begin{bmatrix} W_{bb} & W_{bc} \\ W_{cb} & W_{cc} \end{bmatrix}$	Delassus operator
$\bar{W} = W + h^{-2}diag(k_b^{-1}, k_c^{-1})$	Corrected Delassus operator

1.4 From a truss to a granulate via the tensegrity.

We discuss in this section the static and dynamical modellings of a structure composed with bars and more or less cables to illustrate the concepts introduced above. Such a discrete structure may be described as a set of nodes and links between them, the non smoothness only occuring in the constitutive relations of the links. When the dynamics is considered, the masses are concentrated in the nodes. In the set of nodes Ω, we distinguish the subset Γ_u of the nodes where the displacement is prescribed to clamp the structure to the support. Three configurations are to be considered: the current one Ω_1 for which the tensions and displacements are unknown, the prestressed configuration Ω_0 before applying additional external loading and the relaxed configuration Ω_{-1} for which the selfstresses are virtually vanished. The three configurations are assumed to be close enough to preserve the principle of small perturbations and the prestresses are assumed to be given. In Table 2, the main notations are introduced.

Basic equations. For smooth motions the dynamical equation involves the time-derivative of the velocities. Since shocks are expected, it is more convenient to write this equation as a *measure differential equation* (Moreau (1998), Jean (1999)),

$$M dv + B^t \tau d\nu = F^d dt. \tag{1.6}$$

where dt is a Lebesgue measure, dv is a differential measure representing the acceleration, $d\nu$ a non-negative real measure relative to which dv happens to possess a density function, and τ is a representative of local density of tension forces. The *balance equation* consists in neglecting the inertial term and the measures.

$$-F + F^d = 0 \quad with \quad F := B^t \tau. \tag{1.7}$$

A dual (or reduced to links) formulation of the dynamics may be preferred using the Delassus operator W,

$$d\eta - W \tau d\nu = -\bar{F}^d dt = -BM^{-1} F^d dt. \tag{1.8}$$

When a time discretisation is performed an elementary subinterval $[t^-, t^+]$ of length h is considered. Since discontinuous velocities are expected, high order integration schemes are not necessary and even troublesome; first-order schemes are enough when many shocks may occur simultaneously. We consider here a fully implicit scheme underlining the impulsion π over the time step as the product of h by an average tension τ^+ considered at the end of step,

$$\eta^+ - \eta^- - W\pi = -h\bar{F}^d \quad with \quad \pi = h\tau^+. \tag{1.9}$$

For statics, the *strain admissibility* equation connects the nodal displacements to length variations of the links; for dynamics the *strain rate admissibility* equation connects the nodal velocities to relative velocities of the ends of the links (see the Table 2).

Constitutive laws. The behaviour relation for a bar indiced by α involves a local stiffness between the tension and the length variation taking into account the prestress,

$$\tau_\alpha = k_\alpha (e_\alpha + e^0_\alpha). \tag{1.10}$$

The dynamical version of this relation involves the relative velocity η^+_α and the impulsion π_α,

$$\pi_\alpha = h^2 k_\alpha (-\eta^+_\alpha + \frac{1}{h}(e^-_\alpha + e^0_\alpha)). \tag{1.11}$$

An *inextensible cable* may be modelled with complementary conditions summarized as follows,

$$0 \leq -e_\alpha \ \perp \ \tau_\alpha \geq 0. \tag{1.12}$$

According to the approach of Moreau (1998), a dynamical discrete version is derived involving complementarity conditions between relative velocity and impulsion,

$$\begin{cases} if -e^-_\alpha > 0 \ \ then \ \ \tau^+_\alpha = 0, \\ if -e^-_\alpha \leq 0 \ \ then \ \ 0 \leq \eta^+_\alpha \perp \pi_\alpha \geq 0. \end{cases} \tag{1.13}$$

An integration lemma given by Moreau (1998) proves that the iterates verifying (1.13) tend to verify (1.12) when the time step h tends to zero.

The modelling of *extensible cables* is a combination of a bar and an inextensible cable; the behaviour law takes the form of a piecewise linear function. But we can easily prove that this relation is equivalent to a complementarity condition between the tension and a *corrected length variation* introduced in the Table 2,

$$\tau_\alpha = \left\{ \begin{array}{cc} k_\alpha(e_\alpha + e_\alpha^0) & if \quad e_\alpha + e_\alpha^0 > 0 \\ 0 & if \quad e_\alpha + e_\alpha^0 \leq 0 \end{array} \right. \quad \Longleftrightarrow \quad 0 \leq \lambda_\alpha \ \perp \ \tau_\alpha \geq 0. \tag{1.14}$$

According to previous developments the dynamical discrete version links the impulsion to a *corrected relative velocity* defined in the Table 2,

$$\tau_\alpha^+ = \left\{ \begin{array}{cc} k_\alpha(e_\alpha^- - h\eta_\alpha^+ + e_\alpha^0) & if \quad e_\alpha^- - h\eta_\alpha^+ + e_\alpha^0 > 0 \\ 0 & if \quad e_\alpha^- - h\eta_\alpha^+ + e_\alpha^0 \leq 0 \end{array} \right. \quad \Longleftrightarrow \quad 0 \leq \lambda_\alpha^+ \ \perp \ \pi_\alpha \geq 0. \tag{1.15}$$

With these ingredients we can postulate some problems with different mathematical features.

Truss. If the system is only composed by bars, the equilibrium of the so obtained truss is classically characterized by a linear system with the nodal displacements as unknowns,

$$Ku = F^d - B^t k e^0. \tag{1.16}$$

The (smooth) dynamical behaviour is governed by a system of second-order differential equations,

$$M\ddot{u} + Ku = F^d - B^t k e^0. \tag{1.17}$$

If some inextensible cables are added - for instance to hang up the structure to the support because it is too flexible and too heavy - the equilibrium depends on the tension in these cables according to a few complementarity conditions,

$$\left\{ \begin{array}{l} K_b u + B_c^t \tau_c = F^d - B_b^t k_b e_b^0 \\ 0 \leq -B_c u \ \perp \ \tau_c \geq 0. \end{array} \right. \tag{1.18}$$

It is then a structural type problem as defined in (1.1); the global stiffness matrix of the truss of bars is invertible and the system (1.18) is equivalent to minimize a lower bounded quadratic bulk energy under convex constraints i.e. a well-posed problem.

Net and granulate. If a lot of bars are replaced by inextensible cables (to make lighter the structure for instance) the matrix K_b may not to be invertible and the problem is no more well-posed. The dynamical problem may be solved more easily and takes the form of a linear complementarity problem,

$$\left\{ \begin{array}{l} \hat{W}_c \pi_c - \eta_c^+ = \hat{F}_c^d \\ 0 \leq \eta_c^+ \perp \pi_c \geq 0. \end{array} \right. \quad with \quad \begin{array}{l} \hat{W}_c = W_{cc} - W_{cb}\bar{W}_{bb}^{-1} W_{bc} \\ \hat{F}_c^d = -\eta_c^- + h\bar{F}_c^d + W_{cb}\bar{W}_{bb}^{-1}[\eta_b^- - h\bar{F}_b^d - \frac{1}{h}(e_b^- + e_b^0)] \end{array} \tag{1.19}$$

We can imagine to replace all the bars by cables - think to a catenary arch model of Gaudi (see Figure 3) or to a fishing net. The dynamical discrete behaviour then derives from the previous system with $\hat{W}_{cc}$ equal to W and this system is similar to the one issued from the modelling of a granular system with frictionless contact between grains i.e. a granular type problem (1.3) (1.4).

Figure 3. Catenary arch model (Gaudi museum) and the Needle Tawer (tensegrity).

Tensegrity. Generally the stiffness of the cables is weaker than the one of the bars. It is then convenient to consider extensible cables instead of inextensible ones. Such a structure is a selfstressed tensegrity system if the set of compressed components is discontinuous and the set of tensioned components is continuous (Motro (2003)). The matrix K_b is then singular with a large kernel composed of the rigid modes of all the bars. Since the global stiffness of the associated truss is invertible, the equilibrium of the system may be characterized by a linear complementarity problem involving the tensions in the cables and the corrected length variations defined in the Table 2,

$$\left\{ \begin{array}{l} A_c\lambda_c - \tau_c = -k_cB_cK^{-1}F^d - \tau_c^0 \\ 0 \leq \lambda_c \perp \tau_c \geq 0. \end{array} \right. \quad with \quad A_c = k_c - k_cB_cK^{-1}B_c^tk_c. \tag{1.20}$$

The one step dynamical discrete problem is still a LCP in considering the corrected relative velocities λ_c^+,

$$\left\{ \begin{array}{l} \tilde{W}_c\pi_c - \lambda_c^+ = \tilde{F}_c^d \\ 0 \leq \lambda_c^+ \perp \pi_c \geq 0. \end{array} \right. \quad with \quad \begin{array}{l} \tilde{W}_c = \bar{W}_{cc} - W_{cb}\bar{W}_{bb}^{-1}W_{bc} \\ \tilde{F}_c^d = \hat{F}_c^d + \frac{1}{h}(e_c^- + e_c^0) \end{array} \tag{1.21}$$

2 Contact solvers and parallel treatment.

Some non smooth direct solvers exist, like the Lemke algorithm for LCP problems, but such methods are devoted to small systems and have not been improved to be implemented on multiprocessor computers. For large-scale problems the iterative algorithms are often chosen; the efficiency is related to the mathematical properties of the problem and to acceleration procedures as relaxation or preconditioners. The adaptation to a parallel treatment is facilitated by the simple features of some elementary stages of the algorithms : small elementary non smooth problems but large linear problems, matrix-vector products. Indeed such stages are easy to treat on a multiprocessor computer.

2.1 Stationnary iterative methods and multithreading.

It can be taken advantage of the structure of the problem (1.3)(1.5) to extend a stationnary iterative algorithm from a linear system to a non smooth one. Indeed the nonsmoothness is restricted to a diagonal operator, while the coupling between the contacts involves the non diagonal part of the $\mathbb{W}$ matrix. We focus our attention on the Gauss-Seidel method.

Non Linear Gauss Seidel algorithm. The solution is made through a contact-by-contact like non linear Gauss-Seidel method. So we consider the contact α and suppose that the others are fixed. To reach the unknowns $(\mathbf{v}_\alpha, \mathbf{r}_\alpha)$, the iterative scheme is defined as follows (iteration k+1) :

$$\left\{ \begin{array}{l} \mathbb{W}_{\alpha\alpha}\mathbf{r}_\alpha^{k+1} - \mathbf{v}_\alpha^{k+1} = \mathbf{b}_\alpha - \sum_{\beta<\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\beta^{k+1} - \sum_{\beta>\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\beta^{k} = \bar{\mathbf{b}}_\alpha \\ law_\alpha[\mathbf{v}_\alpha^{k+1}, \mathbf{r}_\alpha^{k+1}] = true \end{array} \right. \tag{2.1}$$

Such an algorithm appears as a generic solver to solve general systems modelling the behaviour of a collection of particles or bodies interacting between them via various laws. The method is robust enough to converge toward a solution even if the interaction laws do not assure the uniqueness of a global solution. The indeterminacy may be also due to the properties of the $\mathbb{W}$ matrix i.e. related to the associated linear problem, as in some hyperstatic trusses of bars.

Implementation. The implementation is very simple if the $\mathbb{W}$ matrix is available with a low cost in term of storage memory. Even if we have only to store the non zero block matrices $\mathbb{W}_{\alpha\beta}$, such a storage may be important for large problems. The algorithm is schematized in the Table 3. This implementation is called SDL.

If the storage of the $\mathbb{W}_{\alpha\beta}$ matrices is too important, the right-hand side of the first equation in (2.1) may be computed without storing the matrices $\mathbb{W}_{\alpha\beta}$ previously. This step depends of course on the problem type and how the matrix $\mathbb{W}$ is built. In a finite element context with gathering of the elementary matrices, the splitting of $\mathbb{W}$ into $\mathbb{W}_{\alpha\beta}$ is not so easy to perform. For granular type softwares (as LMGC90 Jean (1999)), in which we have to pass often from the global level of the bodies to the local level of the contacts, the procedure is natural. The Table 4 provides in this case the additional tasks

Table 3. Non Linear Gauss Seidel algorithm with storage of the matrices (SDL).

(o) Evaluating all the matrices $\mathbb{W}_{\alpha\beta}$
- $k = k + 1$ (NLGS iteration)
  - $\alpha = \alpha + 1$ (Contact index)
    - (a) Evaluating the right-hand side
      $\bar{\mathbf{b}}_\alpha = \mathbf{b}_\alpha - \sum_{\beta<\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\beta^{k+1} - \sum_{\beta>\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\beta^{k}$
    - (b) Solving the local problem (2.1),
  - Convergence test for $k = 0...k_{max}$

to insert in the scheme. From a mechanical point of view, in the right-hand side of the first equation in (2.1), computed at the step (a), the terms involving the $\mathbb{W}_{\alpha\beta}$ matrices represent the contribution of the stresses $\mathbf{R}_j$ and $\mathbf{R}_l$ applied to the two bodies concerned with the α contact, indiced here by j and l. The matrix $\mathbb{H}_\alpha$ passes from the local frame of the α contact to the global one of the grains. The step (o) is reduced to the computation of the only block diagonal matrices $\mathbb{W}_{\alpha\alpha}$ (step (o')). After solving the local contact problem the stresses in the two bodies are updated at step (ii). This implementation is called ELG.

Table 4. Non Linear Gauss Seidel algorithm without storage of the matrices (ELG).

(o') Evaluating the block diagonal matrices $\mathbb{W}_{\alpha\alpha}$
- $k = k + 1$ (NLGS iteration)
  - $\alpha = \alpha + 1$ (Contact index)
    - (i) Identifying the contacting bodies ($\alpha = jl$)
      Computing an auxiliary value
      $\mathbf{b}_\alpha^{aux} = \mathbb{H}_\alpha^T[(\mathbb{M}_j)^{-1}\mathbf{R}_j^k - (\mathbb{M}_l)^{-1}\mathbf{R}_l^k]$
    - (a) Evaluating the right-hand side
      $\bar{\mathbf{b}}_\alpha = \mathbf{b}_\alpha - \mathbf{b}_\alpha^{aux} + \mathbb{W}_{\alpha\alpha}\mathbf{r}_\alpha^k$
    - (b) Solving the local problem (2.1),
    - (ii) Updating the stresses on bodies j and l, $\mathbf{R}_j$ and $\mathbf{R}_l$
      $$\begin{bmatrix} \mathbf{R}_j \\ \mathbf{R}_l \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{R}_j \\ \mathbf{R}_l \end{bmatrix}^{k} + \mathbb{H}_\alpha(\mathbf{r}_\alpha^{k+1} - \mathbf{r}_\alpha^k)$$
  - Convergence test for $k = 0...k_{max}$

Before analyzing the behaviour of the two implementations in a multiprocessor environment, it is instructive to study the additional cost of the second implementation for a sequential treatment. We consider a depositing process of particles in a box under gravity with an increasing number of particles. The relevant parameter is the average number of contacts $< n_c >$ successively equal to 1 070, 4 100, 8 200, 17 100, 33 500,

70 400 and 137 040. The Figure 4 shows the evolution of two ratios; $< it >$ denotes the average number of ELG iterations over the average number of SDL iterations; it underlines the eventual perturbations due to roundoff errors cumulated in extra computations. Ts is the average ELG iteration time over the average SDL iteration time; it gives the additional cost in CPU time due to extra computations.

$$< it >= \frac{average\ ELG\ iteration\ number}{average\ SDL\ iteration\ number} \qquad Ts = \frac{average\ ELG\ iteration\ time}{average\ SDL\ iteration\ time} \quad (2.2)$$

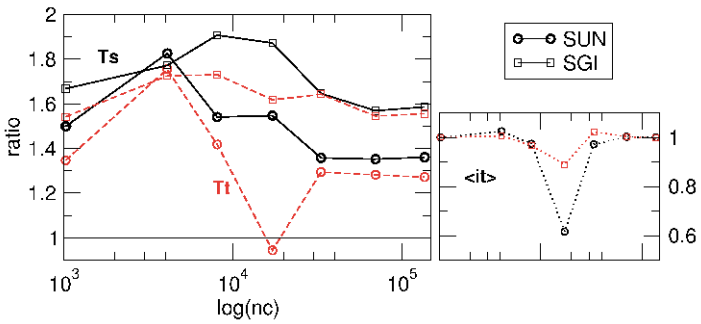

Figure 4. Extra cost of ELG implementation with respect to SDL one in terms of iteration number and time consuming.

Two types of computer are tested. In the Figure 4 le ratio $< it >$ is almost constant equal to 1 except for one simulation $< n_c >= 17\ 100$ for which a crisis occurs at the end of the process where the iteration number blows up with the SDL implementation. But this situation is exceptional : a dynamical crisis may be delayed by the use of an implementation or an other, but it is not erased as it can be viewed in the middle of the process described in the Figure 5 in terms of iteration number. The CPU time extra cost is 1.5 on average with a slight decrease with respect to the number of contacts (cf Figure 4).

Multithreading. The approach developed here want to be the simplest one in using multiprocessor techniques available on shared memory computers. Indeed the distributed memory architecture requires the splitting of the data into different memories in such away this splitting reduces at best the informations to exchange between the processors and theirs memories. Such a strategy is inherent to domain decomposition methods that are presented in a next section and that impose important developments. Moreover the solver cannot be separated from the storage of the whole problem.

Otherwise a Gauss-Seidel type algorithm is essentially sequential because it is a "line-by-line" solver. From a mechanical point of view and using the terminology of granular problems, an impulsion in a contact between two bodies has to be updated in

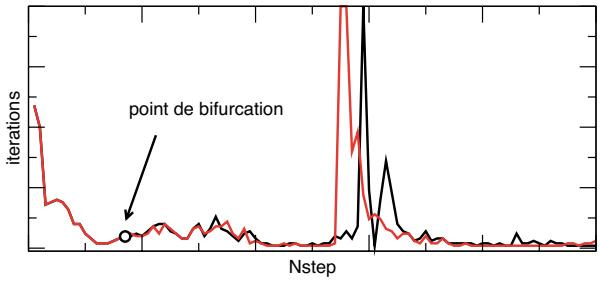

Figure 5. Iteration number versus time steps for the two implementations.

Table 5. Inserting *OpenMP* directives in the NLGS scheme (SDL).

(o) Evaluating all the matrices $\mathbb{W}_{\alpha\beta}$
$k = k + 1$ (NLGS iteration)
!$OMP PARALLEL PRIVATE(...) SHARED(...) ...
!$OMP DO ...
$\alpha = \alpha + 1$ (Contact index)
(a) Evaluating the right-hand side
$\bar{\mathbf{b}}_\alpha = \mathbf{b}_\alpha - \sum_{\beta<\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\alpha^{k+1} - \sum_{\beta>\alpha} \mathbb{W}_{\alpha\beta}\mathbf{r}_\alpha^{k}$
(b) Solving the local problem (2.1),
!$OMP END DO
!$OMP END PARALLEL
Convergence test for $k = 0...k_{max}$

the memory as soon as it has been computed. A shared memory architecture is then more practical. We propose to parallelize the NLGS algorithm itself, independently of any geometric or topologic information. Technically this is performed using OpenMP (http:\\www.openmp.org, Gondet and Lavallee (2000)) directives. It presents major advantages: its use is transparent, and its implementation allows to keep the same source code for parallel or scalar use.

The multithreading procedure consists in splitting the contact loop between P *threads* which may be related to different processors. In the context of shared memory architecture, this splitting is easily performed via OpenMP directives, removing the control of message passing for distributed memory architecture. The only difficulty consists in identifying which variables must be shared between the processors or be private for each one.

This strategy leads to a contact loop renumbering in such a way the sequence of the iterates is modified even if a single processor is used (it is useful to debug for instance). With a multiprocessor computer this renumbering is stochastic and can generate a *race condition* when two processors have to update the stress applied to a same body (this

concerns the ELG implementation). We evaluate the *algorithmic performance* for P processors $Pa(P)$ as the ratio of the average sequential iteration number $< it >_{seq}$ over the average iteration number with P processors $< it >_P$. The parallel performance is given by a *relative speed-up* $S(P)$, taking into account the algorithmic performance, using the time consuming in the solver for a sequential running T_{seq} and for P processors $T(P)$,

$$Pa(P) = \frac{< it >_{seq}}{< it >_P} \qquad S(P) = Pa(P)\ \frac{T_{seq}}{T(P)}. \tag{2.3}$$

Numerical tests. In this part we discuss the results obtained on different simulations: a free surface compaction and a biaxial test. We put poly-disperse disks under gravity (for 95% average radius equal to $0.01m$ and 5% equal to $0.02m$) in a box. For all disks, the mass density is $580kg.m^{-3}$. After depositing, the velocity of the lateral walls is governed by the following equation :

$$|v_x| = \frac{1}{60}(1 - cos(\frac{\pi.t}{30})).$$

The process is performed using 10000 time steps ($h = 6.10^{-2}\ s$). The compaction is performed considering three situations.

The biaxial test consists in imposing a biaxial deformation to a square box: a constant pressure is applied on the lateral wall since a constant velocity is prescribed above. The process is carried out until 10% deformation. The samples have 1 016 and 10 251 disks. In the Figure 6 the gain in terms of time using the SDL implementation decreases with the number of processors for the two problems while the iteration number is weakly perturbed.

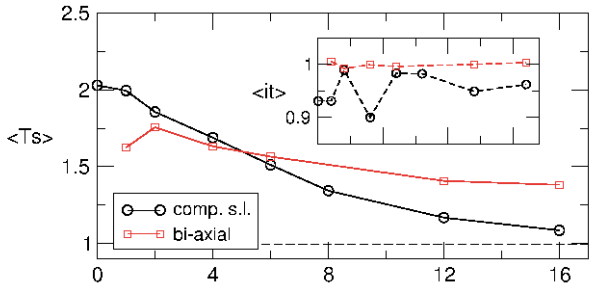

Figure 6. Evolution of the extra cost of the ELG implementation versus the number of processors. In insert the evolution of the ratio of iteration number.

The parallel performances of the ELG implementation are better than these of the SDL one (cf Figure 7). The speed-up may be greater than P, but it is an artefact, called *superlinear behaviour* due to the management of the memory in multiprocessor architecture.

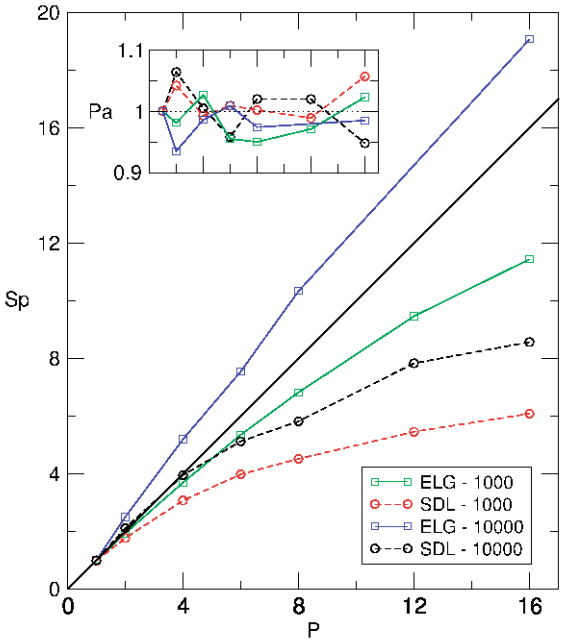

Figure 7. Speed-up for two samples and the two implementations performed on *SGI Origin 3800*.

3D extension. Generally the solution of the local contact problem (2.1) requires an iterative algorithm as the generalized Newton method because frictional contact law involve non smooth projection operators on a non polyedral set $\mathcal{C}(\mu r_{\alpha,n})$,

$$\left\{\begin{array}{l} \mathbb{W}_{\alpha\alpha}\mathbf{r}_\alpha^{k+1} - \mathbf{v}_\alpha^{k+1} = \bar{\mathbf{b}}_\alpha \\ r_{\alpha,n} - proj_{\mathbb{R}^+}(r_{\alpha,n} - \rho_n \mathbf{v}_{\alpha,n}) = 0, \\ \mathbf{r}_{\alpha,t} - proj_{\mathcal{C}(\mu r_{\alpha,n})}(\mathbf{r}_{\alpha,t} - \rho_t \mathbf{v}_{\alpha,t}) = \mathbf{0}. \end{array}\right. \tag{2.4}$$

Such an iterative procedure may be avoided for granular system involving spheres because the $\mathbb{W}_{\alpha\alpha}$ matrix is then diagonal. A study with two samples (Figure 8) shows that the ELG implementation is now more attractive than the SDL one and the advantage increases with the number of processors. Such a behaviour is quite different than in a 2D modelling (cf Figure 4). Moreover the number of iterations is much more perturbed by the implementation type. In the Figure 9 the speed-up is analyzed on a depositing process under gravity with two samples (1 317 and 3 530 spheres). The multiprocessor behaviour of the ELG implementation is still better than the SDL one, but the speed-up moves away from the optimal curve when the processor number increases. Moreover the algorithmic performance of the ELG implementation is strongly penalized with an extra iteration number around 40% (cf insert in the Figure 9).

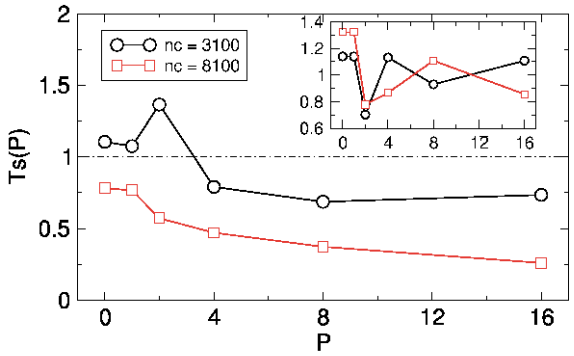

Figure 8. Evolution of the extra cost of the ELG implementation versus the number of processors. In insert the evolution of ration of iteration number (3D case).

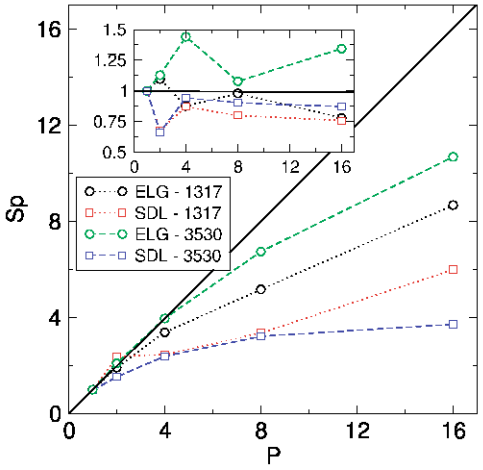

Figure 9. Speed-up for two samples and the two implementations performed on *SGI Origin 3800* (3D case).

2.2 Projected conjugate gradient algorithms.

Conjugate gradient methods have been used to solve frictionless contact problems in the eighties (Dilintas et al. (1988); May (1986)) in the context of finite element modelling. An extension for frictional contact is proposed by Raous and Barbarin (1992), but it is

restricted to structural problems using the displacement field as the main unknown; moreover a fixed point loop is introduced for satisfying the Coulomb law, and the friction law is regularized to be inserted as a non quadratic but smooth additional term in the energy to minimize. We propose in the following a single loop algorithm very close to a standard conjugate gradient method with local adaptations consisting in iterate corrections and gradient projections. This approach is motivated by granular type problems that may be expressed as a *box constrained quadratic problem* (Dostal (1997)). Duality based methods may also transform a structural problem in such a problem. The frictional laws lead to quasi-optimisation formulations and require specific treatments developped in a second section. In the following we consider a *standard constrained quadratic problem*,

$$\mathbf{r} \in \underset{\tilde{\mathbf{r}} \in \mathcal{C}}{argmin}\ \frac{1}{2}\tilde{\mathbf{r}}.\mathbb{W}\tilde{\mathbf{r}} - \mathbf{b}.\tilde{\mathbf{r}}. \tag{2.5}$$

It is interesting to specify the geometric features of the constraint set $\mathcal{C}$ according to the kind of the problem. If the unilateral contact is only accounted for, the set $\mathcal{C}$ is the non negative cone of $\mathbb{R}^{n_c}$ or the non negative orthant. If friction is only considered (the contact is assumed to be maintained and the normal component of the contact reaction known), the convex set is an hyper rectangle depending on a parameter: the contact by contact Tresca threshold s_α. For a *Tresca like frictional contact problem* in a bidimensional modelling the set $\mathcal{C}$ is the cartesian product of infinite half-bands in $\mathbb{R}^2$.

$$\mathcal{C}(s) = \prod_{\alpha=1}^{n_c}\ \mathbb{R}^+ \times [-s_\alpha, s_\alpha] \tag{2.6}$$

For these three cases the set $\mathcal{C}$ is a polyhedral convex set, and even box type. For a three-dimensional Tresca like frictional contact problem the set $\mathcal{C}$ is the cartesian product of infinite half-cylinders in $\mathbb{R}^3$ and it is no more polyhedral.

The classical Coulomb law links the friction threshold to the normal component of the contact force (or impulsion) via a friction coefficient μ. The coupled frictional contact problem is not an optimization problem anymore. Formally it is always possible to formulate a "quasi"-optimization problem (in reference to more classical quasi-variational inequalities which derive from it) for which the constraint set depends on the normal components of the solution as a parameter; only the *granular type frictional contact problem* is given,

$$\mathbf{r} \in \underset{\tilde{\mathbf{r}} \in \mathcal{C}(\mu r_n)}{argmin}\ \frac{1}{2}\tilde{\mathbf{r}}.\mathbb{W}\tilde{\mathbf{r}} - \mathbf{b}.\tilde{\mathbf{r}}. \tag{2.7}$$

Frictionless case. The frictionless granular type problem (1.3) (1.4) is so reformulated as a *cone constrained quadratic problem*,

$$\mathbf{r} \in \underset{\tilde{\mathbf{r}} \geq \mathbf{0}}{argmin}\ \frac{1}{2}\tilde{\mathbf{r}}.\mathbb{W}\tilde{\mathbf{r}} - \mathbf{b}.\tilde{\mathbf{r}}. \tag{2.8}$$

We define a conjugate gradient type approach by conjugating the previous descent direction $\mathbf{p}^{k-1}$ with the current gradient $\mathbf{u}^k$ after projecting them on the tangent cone

Table 6. Conjugate Projected Gradient algorithm.

$\mathbf{r}^0$, $\mathbf{u}^0$ and $\mathbf{p}^0$ given
$k \rightarrow k+1$
$Update \;\; \mathbf{r}^{k+\frac{1}{2}} = \mathbf{r}^k + \alpha^{k+1}\mathbf{p}^k \quad$ with $\alpha^{k+1} = \frac{\mathbf{u}^k.\mathbf{p}^k}{\mathbf{p}^k.\mathbb{W}\mathbf{p}^k}$
$Project\ iterate \;\; \mathbf{r}^{k+1} = proj_{\mathcal{C}}(\mathbf{r}^{k+\frac{1}{2}})$
$Compute\ residual \;\; \mathbf{u}^{k+1} = \mathbf{b} - \mathbb{W}\mathbf{r}^{k+1}$
$Precondition \;\; \mathbf{u}^{k+1} \leftarrow \mathbb{P}\mathbf{u}^{k+1}$
$Project\ gradient \;\; \mathbf{w}^{k+1} = proj(\mathbf{u}^{k+1}; T_{\mathcal{C}}(\mathbf{r}^{k+1}))$
$Project\ gradient \;\; \mathbf{z}^{k+1} = proj(\mathbf{p}^k; T_{\mathcal{C}}(\mathbf{r}^{k+1}))$
$Conjugate \;\; \mathbf{p}^{k+1} = \mathbf{w}^{k+1} + \beta^{k+1}\mathbf{z}^{k+1} \quad$ with $\beta^{k+1} = \frac{\mathbf{w}^{k+1}.\mathbb{W}\mathbf{p}^k}{\mathbf{p}^k.\mathbb{W}\mathbf{p}^k}$

$T_{\mathcal{C}}(\mathbf{r}^{k+1})$, and so defining a Conjugate Projected Gradient (CPG) algorithm (cf table 6).

The conjugating process is a priori fully efficient if the iterates stay during several iterations in the same set of active constraints, that is to say on the same facet or edge of the constraint set. An interesting case is the set of active constraints to which the solution belongs; this set corresponds in contact mechanics to the set of the solution statuses contact by contact. After this set is found the Gauss-Seidel algorithm is slow to reach the solution; we can hope in this case a better behavior of the conjugate gradient algorithm; numerical tests have to confirm or to invalidate this prediction. Such a strategy was proposed by May (1986) and Dilintas et al. (1988) for frictionless contact problems, but our formulation is more general and synthetic. Indeed Dilintas et al. (1988) fixes the set of active constraints to perform the CPG algorithm until convergence before updating the active constraints.

Such an algorithm does not benefit from a theoretical convergence result but is simple to carry out and may be easily treated by parallel computing because the extra instructions in comparison with a conjugate gradient linear solver are projections (iterates and gradients) only performed component-wise,

$$\mathbf{r}^{k+1} = proj_{\mathcal{C}}(\mathbf{r}^{k+\frac{1}{2}}) \Longleftrightarrow \quad for \;\; \alpha = 1, n_c, \;\; r_\alpha^{k+1} = max(r_\alpha^{k+\frac{1}{2}}, 0). \tag{2.9}$$

$$\mathbf{w}^{k+1} = proj(\mathbf{u}^{k+1}; T_{\mathcal{C}}(\mathbf{r}^{k+1})) \Longleftrightarrow \tag{2.10}$$

$$for\ \alpha = 1, n_c,\ if\ r_\alpha^{k+1} = 0\ then\ w_\alpha^{k+1} = max(u_\alpha^{k+1}, 0)\ else\ w_\alpha^{k+1} = u_\alpha^{k+1}. \tag{2.11}$$

Frictional case. For a Coulomb frictional contact law, the Tresca like frictional contact problem may be viewed as an intermediate problem in a numerical solution strategy. A classical approach consists in carrying out a fixed point algorithm on the friction threshold. But we formulate in the Table 7 directly a diagonalized version (with a single loop) which may be justified in (Renouf and Alart (2005)). The modifications of the previous algorithm concern the correction of the iterate and the projection of the gradients. They are presented contact by contact (locally) and restricted to the two-dimensional modelling.

Table 7. Conjugate Projected Gradient Algorithm (*CPG*), frictional contact version.

Initialization of $\mathbf{u}^0 = \mathbf{b} - \mathbb{W}\mathbf{r}^0, \mathbf{w}^0 = \mathbf{u}^0,\ \mathbf{p}^0 = \mathbf{u}^0$
$k = k + 1$

$\mathbf{r}^{k+\frac{1}{2}} = \mathbf{r}^k + \alpha^{k+1}\mathbf{p}^k$ with $\alpha^{k+1} = \dfrac{\mathbf{u}^k.\mathbf{p}^k}{\mathbf{p}^k.\mathbb{W}\mathbf{p}^k}$

Correction of the iterate : $\mathbf{r}^{k+\frac{1}{2}} \rightarrow \mathbf{r}^{k+1}$

$\mathbf{u}^{k+1} = \mathbf{b} - \mathbb{W}\mathbf{r}^{k+1}$ (preconditioner: $\mathbf{u}^{k+1} = \mathbb{P}(\mathbf{b} - \mathbb{W}\mathbf{r}^{l,k+1})$)

Convergence criterion : if true then $\mathbf{r}^{\text{conv}} = \mathbf{r}^{k+1}$

Projection of gradients: $\mathbf{w}^{k+1} = proj(\mathbf{u}^{\mathbf{k+1}}; T_{\mathcal{C}^{k+1}}(\mathbf{r}^{k+1}))$
$\mathbf{z}^{k+1} = proj(\mathbf{p}^k; T_{\mathcal{C}^{k+1}}(\mathbf{r}^{k+1}))$

$\mathbf{p}^{k+1} = \mathbf{w}^{k+1} + \beta^{k+1}\mathbf{z}^{k+1}$ with $\beta^{k+1} = -\dfrac{\mathbf{w}^{k+1}.\mathbb{W}\mathbf{p}^k}{\mathbf{p}^k.\mathbb{W}\mathbf{p}^k}$

The *correction of the iterate* $\mathbf{r}_\alpha^{k+\frac{1}{2}}$ is then carried out on a new local convex set, related to a new local threshold $s_\alpha^{k+1} = max(0, \mu r_{\alpha,n}^{k+\frac{1}{2}})$, if the iterate does not belong to it. But when the iterate is inside $\mathcal{C}_\alpha^{k+1}$, the situation is quite complicated. A strict application of a projection procedure would lead to confirm $\mathbf{r}_\alpha^{k+\frac{1}{2}}$ as $\mathbf{r}_\alpha^{k+1}$ and so consider a "stick" status for this contact, which is not optimal if the previous status was "slip". Indeed, assume that the solution status is "slip" (backward or forward), the iterate may oscillate between "slip" and "stick" because of the conical form of the Coulomb criterion. For taking advantage of the conjugating process it is more convenient to keep the descent directions in the same subspace corresponding to the "slip" status. Consequently if the current local descent direction $\mathbf{p}_\alpha^k$ has no tangential component, the local contact status, which was then "slip" previously, has to be maintained even if the iterate is inside the friction cone and inside its approximated cylindrical set $\mathcal{C}_\alpha^{k+1}$; such a situation is shown with the case (*b*) in the Figure 10. The status provides the pertinent information on the iterate at which the tangent cone $T_\mathcal{C}$ has to be evaluated.

Since the approximating convex set is well defined the *projection of the gradients* on the tangent cone is standard. But it is instructive to illustrate these procedures according to the previous contact status first stored in memory (cf the Figure 11).

In order to preserve a parallel treatment "contact by contact" a simple diagonal preconditioner is only considered. We refer the reader to the paper of Renouf and Alart (2005) for details about pre-conditioning and convergence criteria.

Sequential numerical tests. In this section the *CPG* algorithm is essentially compared with the *NLGS* method. All simulations are performed with the LMGC90 software Dubois and Jean (2003) dedicated to multicontact problems, very useful for granular materials. The presentation is restricted to the analysis on a single step in a simulation of a granular system. One step is isolated at the end of a depositing process of particles in a box in the field of gravity. This amounts to determine the distribution of the contact forces in a equilibrium state of a granular packing. As the number of contacts is

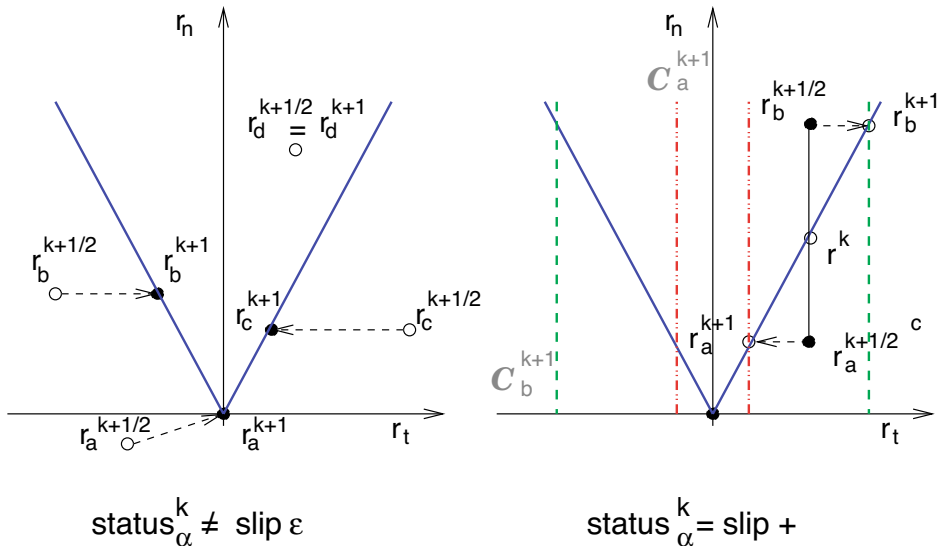

Figure 10. Iterate correction (some situations according to the previous status).

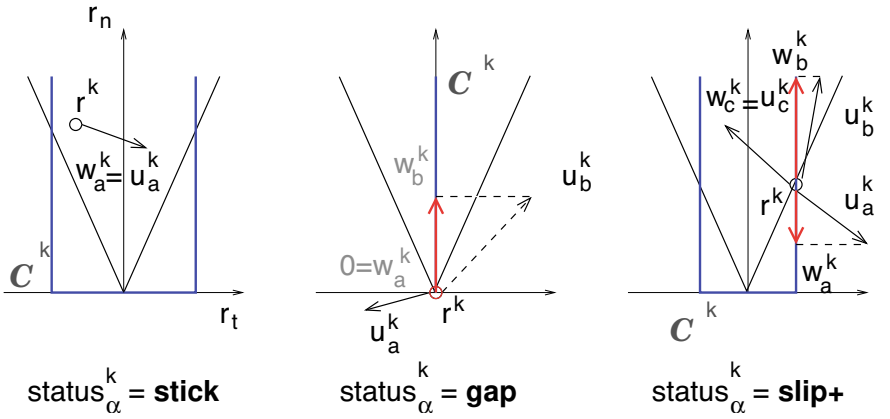

Figure 11. Gradient projection (some situations according to the previous status)

an important parameter of the study several numerical samples are created in the same box ($1m \times 1m$) with different sizes of circular particles. Thus the capacity of the box goes from 1 000 to 33 000 disks. Four algorithms are tested: Non Linear Gauss-Seidel (*NLGS*), Projected Gradient without conjugating (PG), Conjugate Projected Gradient (*CPG*), Preconditioned Conjugate Projected Gradient (*PCPG*). For frictionless contact (Figure 12a) the better behavior of conjugate gradient methods is not surprising but has to be confirmed for frictional contact problems with the adaptations proposed above.
The sample for the Figures 12 and 13 comprises 15 000 contacts. The conjugating process reveals to be necessary for a good behavior of gradient method in comparison with the Gauss-Seidel one. The *NLGS* solver provides a smooth convergence since the *CPG* methods converge faster but with large perturbations and a final drop. The large oscillations with the gradient methods can be attributed to the corrections of the iterate specific to the Coulomb frictional contact law as proved by the smoother curves obtained with a zero friction coefficient (Figure 12).We verify in the Figure 13 the increase of the oscillations with a stronger friction coefficient equal to 0.4. In this example the pre-conditioner improves significantly the algorithm. Two parameters are relevant to appreciate the performance of the gradient methods: the size of the system characterized by the number of contacts n_c and the required accuracy ϵ.

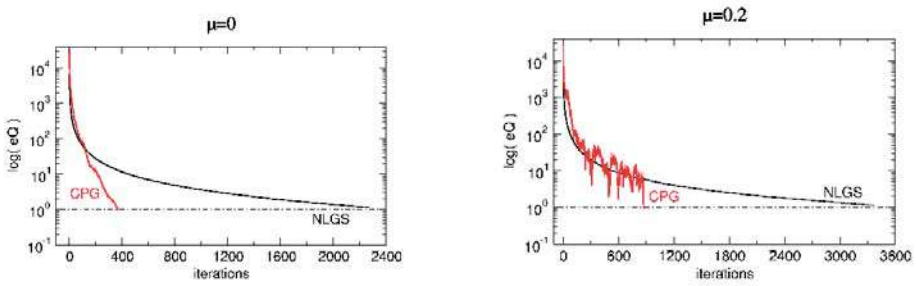

Figure 12. Convergence behavior of CPG/NLGS ($n_c = 15000$) for two friction coefficients.

We present in the Figure 14 the evolution of the gain of iterations from the *NLGS* to the *CPG* or *PCPG* (δ =(number of *NLGS* iterations) / (number of (P)*CPG* iterations)) according to the number of contacts and for the three required accuracies. The gain is at least equal to 2.98 ($n_c = 15\ 249$ and $\epsilon = 1.66\ 10^{-5}$) and may reach 9.3 for *CPG* ($n_c = 15\ 249$ and $\epsilon = 1.66\ 10^{-4}$) and even 9.47 for *PCPG* ($n_c = 28\ 014$ and $\epsilon = 1.66\ 10^{-6}$). Due to the erratic convergence behavior underlined in the Figure 13, it is not simple to conclude below 40 000 contacts. Beyond this value the gain seems to stabilize around 4 for the *CPG* algorithm and 7 for the preconditioned one. Finally the gain is all the more important since the required accuracy is high.

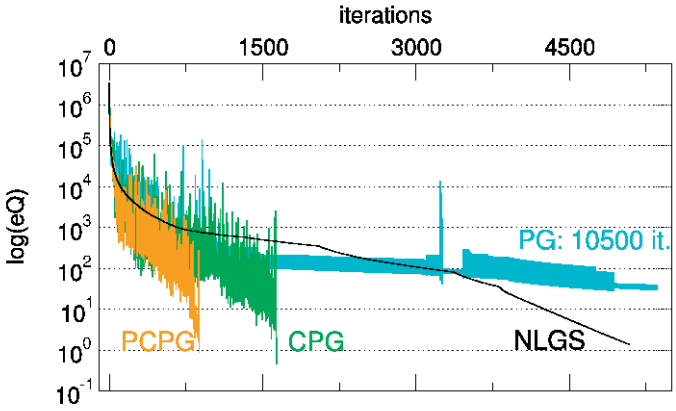

Figure 13. Convergence behavior of the fourth algorithms ($n_c = 15000$) for $\mu = 0.4$.

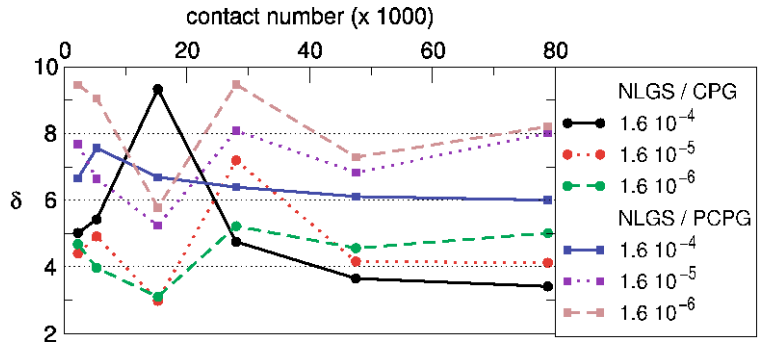

Figure 14. Gain of iterations, δ, of the *CPG* and *PCPG* in regard of the *NLGS*.

Multiprocessor numerical tests. Gradient algorithms are intrinsically parallel due to the fact that they are essentially composed of matrix-vector products. The implementation of a parallel version is thus not a difficulty itself. Like for the *NLGS* algorithm Renouf et al. (2004), we opt for a multithreading on shared memory computer to avoid the message passing. The implementation is so easily performed via OpenMP directives (see Gondet and Lavallee (2000)). Although our algorithm is intrinsically parallel, using parallel treatment modifies slightly the solution on a whole process. On one time step sequential and parallel versions give the same solution. But during a process we accumulate numerical errors which can have more visible consequences in a granular material, and this for the reasons evoked previously. Recall that the infinity of solutions makes the simulations very sensitive to the numerical error accumulations; these errors may be due to the parallelization but also to the implementation procedure as proved above on NLGS algorithms. So during the process, the evolution of the iterations number with sequential and parallel treatment diverges after some time steps. We use once again the relative speed-up, $S(P)$, introduced previously. The Figure 15 represents the evolution of $S(P)$ for a simple shear test and free surface compaction.

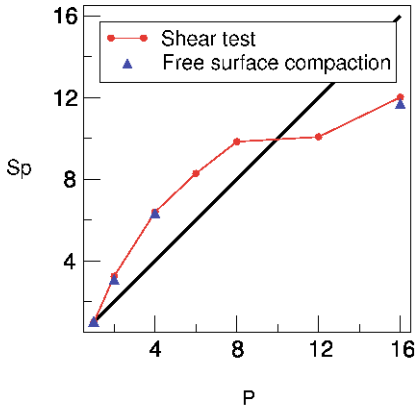

Figure 15. Evolution of the relative Speed-Up. Simulations have been performed on a SGI Origin 3800 using processors R14000/500 Mhz.

The evolution of speed-up is quite curious. This phenomenon is due to the increase of usable size memory. For a sequential computation only 500 Mo are available. This capacity is multiplied by P for a P-parallel simulation. So to the efficiency of parallel computing, we add the benefit of a large memory stack, very useful for large simulation. Consequently, the superlinear speed-up can be explained by this phenomenon until 4 processors. In other word the sequential computation is penalized by the large required memory and should not be taken in reference to evaluate the multiprocessor performances. This illustrates the difficulty to evaluate the parallel performances depending

on many parameters related to the hardware architecture.

3D extensions. The extension to 3D problem is not an easy task due to the non polyedral form of the friction constraints. The projection of the gradients is relevant for polyedral constraints sets because the iterates stay on a facet of active constraints. A first way consists in approximating the Coulomb cone by a pyramid and so recovering a polyedral set (Klarbring (1986)). But a first study shows that the gains in terms of iterations are lower for three-dimensionnal modellings than for bidimensionnal ones (Renouf and Alart (2004)).

3 Domain decomposition approaches

Domain decomposition strategies have been developed in computational structural mechanics to deal with large-scale problems discretized by the finite element method. The domain decomposition methods (DDM), or more generally the substructuring techniques, are efficient methods because they allow to reduce memory storage and calculation time. Moreover these methods take advantage of the new multi-processor generation of computers as they exhibit an intrinsec parallelism with a high granularity. The main component of the domain decomposition algorithm is a numerical solver based on the solution of local independent subproblems on subdomains. In addition, these methods are efficient solvers in a classical monoprocessor environment as well. In a first time, we investigate two domain decomposition approaches introduced for linear systems. In a second time, we discuss two strategies to solve large-scale multicontact problems in using at best the advantages of the two approaches. In the context of these lectures we give some elemental examples developped with the MAPLE software to well understand the technical aspects inherent to these approaches.

3.1 Primal / dual DDM formulations for linear problems.

This section consists of a "parallel" presentation of the primal and dual formulations in a domain decomposition strategy for a better understanding of the insertion of the contact treatment in these approaches. The features of the two methods modify not only the solver algorithm but so the substructuring of the domain itself with respect to the potential contact areas. We present an algebraic version of the DDM without introducing the underlying continuous problem (in elasticity for instance) even if some steps are characterized in reference to continuous problems (Dirichlet or Neumann problem). The primal approach refers to the BDD method (Balancing Domain Decomposition) intensively studied by Le Tallec (1994); the dual approach refers to the FETI method (Finite Element Tearing and Interconnecting) introduced by Farhat and Roux (1991).

Reference problem and substructuring. The reference problem is a linear system of equations involving an unknown displacement vector u_g, a square, symmetric positive definite matrix K_g arising from the finite element discretization of a linear, elliptic, self-

adjoint boundary value problem on a domain Ω, and a right-hand side force vector f_g,

$$K_g u_g = f_g. \tag{3.1}$$

The global domain Ω, which may be viewed as a set of nodes with degrees of freedom, is split into n_s non-overlapping subdomains $\Omega^{(s)}$, $\Omega = \bigcup_{s=1}^{n_s} \Omega^{(s)}$. The global interface Γ is the assembly of the local interfaces defined from the boundaries of each subdomain,

$$\Gamma = \bigcup_{s=1}^{n_s} \Gamma^{(s)} \quad with \quad \Gamma^{(s)} = \partial\Omega^{(s)} \cap \left(\bigcup_{p=1, p\neq s}^{n_s} \partial\Omega^{(p)} \right) - \partial\Omega. \tag{3.2}$$

The vectors u_g and f_g are split into the internal degrees of freedom u_I and f_I and the interface ones u_Γ and f_Γ. For each subdomain a same splitting is introduced.

$$u_g = \left\{ \begin{array}{c} u_I \\ u_\Gamma \end{array} \right\} \quad , \quad f_g = \left\{ \begin{array}{c} f_I \\ f_\Gamma \end{array} \right\} \quad , \quad u^{(s)} = \left\{ \begin{array}{c} u_i^{(s)} \\ u_b^{(s)} \end{array} \right\} \quad , \quad f^{(s)} = \left\{ \begin{array}{c} f_i^{(s)} \\ f_b^{(s)} \end{array} \right\}. \tag{3.3}$$

The primal formulation consists in imposing the displacement continuity on the interface by introducing the *restriction operators* from the global interface to the local ones,

$$R^{(s)} : \Gamma \rightarrow \Gamma^{(s)} \quad , \quad R^{(s)} : u_\Gamma \rightarrow u_b^{(s)} = R^{(s)} u_\Gamma. \tag{3.4}$$

The reference problem (3.1) is then reformulated as follows,

$$\left\{ \begin{array}{l} K_{ii}^{(s)} u_i^{(s)} + K_{ib}^{(s)} R^{(s)} u_\Gamma = f_i^{(s)} \ , \ s = 1, n_s \\ \sum_s R^{(s)t} K_{bi}^{(s)} u_i^{(s)} + \sum_s R^{(s)t} K_{bb}^{(s)} R^{(s)} u_\Gamma = f_\Gamma = \sum_s R^{(s)t} f_b^{(s)} \end{array} \right. \tag{3.5}$$

The dual formulation consists in controlling the displacement continuity on the interface by a signed Boolean matrix that extracts from a subdomain vectors $u^{(s)}$ its signed ($\pm$) restriction (or trace) to the subdomain interface boundary, without reference to the interface components of the global vector u_Γ,

$$B^{(s)} : \Omega^{(s)} \rightarrow \Gamma^{(s)} \quad , \quad B^{(s)} : u^{(s)} \rightarrow \pm\, u_b^{(s)}. \tag{3.6}$$

The reference problem is then replaced by a constrained problem involving all the local unknowns $u^{(s)}, s = 1, n_s$ by introducing a Lagrange multiplier λ associated with the displacement continuity on the interface,

$$\left\{ \begin{array}{l} K^{(s)} u^{(s)} = f^{(s)} - B^{(s)t} \lambda, \ s = 1, n_s \\ \sum_s B^{(s)} u^{(s)} = 0 \end{array} \right. \tag{3.7}$$

Interface problems. To get the interface problem the primal approach uses a block Gaussian elimination of the internal degrees of freedom, easily performed because the matrix $K_{ii}^{(s)}$ is generally invertible,

$$u_i^{(s)} = K_{ii}^{(s)-1} (f_i^{(s)} - K_{ib}^{(s)} R^{(s)} u_\Gamma) \ , \ s = 1, n_s. \tag{3.8}$$

The interface problem of the dual approach requires a dualization process to eliminate the primal variable on each subdomain,

$$u^{(s)} = K^{(s)-1}(f^{(s)} - B^{(s)t}\lambda) \quad , \; s = 1, n_s. \tag{3.9}$$

If the matrix $K^{(s)}$ is not invertible a specific procedure is needful as developped later. But we assume for the present that $u^{(s)}$ is well defined by the equation (3.9).

The primal interface problem involves the interface displacements as unknowns using the global Schur matrix S as the assembly of local Schur matrices $S^{(s)}$,

$$Su_\Gamma \; = \; b, \quad with \quad S = \sum_s R^{(s)t} S^{(s)} R^{(s)}. \tag{3.10}$$

$$S^{(s)} = K_{bb}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} \quad , \quad b = \sum_s R^{(s)t}(f_b^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} f_i^{(s)}). \tag{3.11}$$

The dual interface problem involves the interface multipliers as main unknowns and the F_I matrix characteristic of the FETI method,

$$F_I\lambda \; = \; d, \quad with \quad F_I = \sum_s B^{(s)} K^{(s)-1} B^{(s)t} \quad , \quad d = \sum_s B^{(s)} K^{(s)-1} f^{(s)}. \tag{3.12}$$

With the assumptions on the reference problem the two interface problems are equivalent to the quadratic optimisation problems,

$$\underbrace{Min}_{v_\Gamma} \quad \frac{1}{2}\, v_\Gamma^t S v_\Gamma - b^t v_\Gamma. \tag{3.13}$$

$$\underbrace{Min}_{\mu} \quad \frac{1}{2}\, \mu^t F_I \mu - d^t \mu. \tag{3.14}$$

Local problems. To benefit from a multi-processor environment, the interface matrices have not to be directly computed but the interface problems underline how to deal with the connections between the parallel solution of subproblems performed on different processors. The modern domain decomposition methods solve the interface problem by an iterative preconditionned conjugate gradient type algorithm. Such an algorithm requires as the main consuming part a matrix-vector product. For the primal interface problem the matrix-vector product $g := Sp$ is naturally split into n_s matrix-vector products $g_b^{(s)} := S^{(s)} p_b^{(s)}$ with $p_b^{(s)} := R^{(s)} p$,

$$g_b^{(s)} := K_{bb}^{(s)} p_b^{(s)} - K_{bi}^{(s)} \bar{g}_i^{(s)} \quad with \quad \bar{g}_i^{(s)} := K_{ii}^{(s)-1} K_{ib}^{(s)} p_b^{(s)}. \tag{3.15}$$

The computation of $\bar{g}_i^{(s)}$ amounts to solve a Dirichlet problem on the subdomain $\Omega^{(s)}$ with $-R^{(s)}p$ as the Dirichlet condition,

$$\begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} \bar{g}_i^{(s)} \\ -R^{(s)}p \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \tag{3.16}$$

For the dual interface problem the matrix-vector product $g := F_I p$ is equivalent to solve n_s Neumann problems,

$$\begin{array}{c} g^{(s)} = B^{(s)}\hat{g}^{(s)} \\ \hat{g}^{(s)} = K^{(s)-1}B^{(s)t}p \end{array} \Leftrightarrow \left[\begin{array}{cc} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{array} \right] \left\{ \begin{array}{c} \hat{g}_i^{(s)} \\ \hat{g}_b^{(s)} \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ \hat{p}_b^{(s)} \end{array} \right\} = B^{(s)t}p. \tag{3.17}$$

Coarse system for the dual approach. When the subdomain $\Omega^{(s)}$ has not sufficient Dirichlet boundary conditions to prevent $K^{(s)}$ from being singular ($\Omega^{(s)}$ is called a floating subdomain), the Neumann problem (3.17) has not a unique solution. For elasticity problems on a single domain, such a situation appears when the decomposition of the whole domain is fine enough to provide subdomains without clamp conditions. For multibody systems for which some bodies are equilibrated by contact with the neighbouring bodies such a situation is inherent to the mechanical problem and generated some researches in term of semi-coercive problems (Dostal et al. (1998)). The dualization process (3.9) is completed in considering a generalized inverse $K^{(s)+}$, the rigid body modes of $\Omega^{(s)}$ collected in the kernel $N^{(s)}$ of $K^{(s)}$ and the set of amplitudes $\alpha^{(s)}$ that specifies the contribution of the null space to the solution $u^{(s)}$,

$$u^{(s)} = K^{(s)+}(f^{(s)} - B^{(s)t}\lambda) + N^{(s)}\alpha^{(s)} \tag{3.18}$$

The coefficients $\alpha^{(s)}$ may be determined by requiring that each subdomain problem be mathematically solvable - that is, each floating subdomain be self-equilibrated - which can be written as,

$$N^{(s)t}(f^{(s)} - B^{(s)t}\lambda) = 0. \tag{3.19}$$

The interface problem (3.12) is then replaced by the following one,

$$\left[\begin{array}{cc} F_I & -G_I \\ -G_I^t & 0 \end{array} \right] \left\{ \begin{array}{c} \lambda \\ \alpha \end{array} \right\} = \left\{ \begin{array}{c} d \\ -e \end{array} \right\}, \tag{3.20}$$

where F_I and d are defined in (3.12) replacing $K^{(s)-1}$ by $K^{(s)+}$, and

$$\begin{array}{rl} G_I = \left[B^{(1)}N^{(1)} \ \ ... \ \ B^{(n_s)}N^{(n_s)}\right] & \\ \alpha = \left[\alpha^{(1)t} \ \ ... \ \ \alpha^{(n_s)t}\right]^t & . \\ e = \left[f^{(1)t}N^{(1)} \ \ ... \ \ f^{(n_s)t}N^{(n_s)}\right]^t & \end{array} \tag{3.21}$$

A constrained quadratic optimisation problem may be associated with this last problem useful to deal later with contact constraints,

$$\underbrace{Min}_{G_I^t\mu - e = 0} \quad \frac{1}{2}\,\mu^t F_I \mu - d^t\mu. \tag{3.22}$$

Since the constraints are linear, the Conjugate Projected Gradient algorithm, as formally presented in Table 6 and reformulated in Table 8, requires only the projection of the gradient defined by the following linear operator,

$$P := I - G_I(G_I^t G_I)^{-1}G_I^t. \tag{3.23}$$

The initialisation of the algorithm uses the splitting of the multiplier λ, where λ^0 is a particular solution of $G_I^t \lambda = e$,

$$\lambda = \lambda^0 + P\bar{\lambda}, \quad with \quad \lambda^0 = G_I (G_I^t G_I)^{-1} e. \tag{3.24}$$

This initialisation and the projection of the gradient at each iteration $p := Pg$ requires the solution of a FETI coarse problem concerning only the α coefficients,

$$p := Pg = g - G_I (G_I^t G_I)^{-1} G_I^t g, \quad where \quad (G_I^t G_I)\alpha = \beta = G_I^t g. \tag{3.25}$$

Preconditioning. Even without preconditioning the interface systems have a better condition number - for instance $cond(S) = \mathcal{O}\left(\frac{1}{H^2}\left(1 + \frac{H}{h}\right)\right)$ - than the reference problem, cond $(K_g) = \mathcal{O}\left(\frac{1}{h^2}\right)$), where H is the subdomain size and h the mesh size. But the conjugate gradient type algorithms are overall efficient when a preconditionner is added.

The Neumann-Neumann preconditioner for the Schur approach consists in approximating the inverse of the sum $S = \sum_{s=1}^{n_s} R^{(s)t} S^{(s)} R^{(s)}$ by a weighted sum of the local Schur inverse matrices $S^{(s)-1}$,

$$\bar{S}^{-1} = \sum_s R^{(s)t} D^{(s)} S^{(s)-1} D^{(s)} R^{(s)}, \tag{3.26}$$

where $D^{(s)}$ is a diagonal weighting matrix verifying $\sum_{n_s}^{s} R^{(s)t} D^{(s)} R^{(s)} = I_\Gamma$. For instance if the domain is homogeneous $D^{(s)}$ may store in each of its entries the inverse of multiplicity of the associated interface degree of freedom, that is, the inverse of the number of subdomains to which the interface d.o.f. belongs. For heterogeneous problems others choices are possible.

The preconditioner of the FETI method is built in a same way,

$$\bar{F}_I^{-1} = D \left(\sum_s B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & A_{bb}^{(s)} \end{bmatrix} B^{(s)t} \right) D, \tag{3.27}$$

where the matrix $A_{bb}^{(s)}$ may be chosen equal to $S_{bb}^{(s)}$ (Dirichlet preconditioner) or equal to $K_{bb}^{(s)}$ (lumped preconditioner).
The preconditioning step must have nice parallel properties for implementation. The preconditioning step of the Schur method $z := \bar{S}^{-1} g$ amounts to solve n_s Neumann problems with $\hat{g}_b^{(s)} = D^{(s)} R^{(s)} g$ as the Neumann condition,

$$\hat{z}_b^{(s)} := S^{(s)-1} \hat{g}_b^{(s)} \quad \Leftrightarrow \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} \hat{z}_i^{(s)} \\ \hat{z}_b^{(s)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \hat{g}_b^{(s)} \end{Bmatrix} = \hat{g}^{(s)}. \tag{3.28}$$

The preconditioning step of the FETI method $z := \bar{F}_I^{-1} g$ is split into n_s matrix-vector products $\check{z}_b^{(s)} := A_{bb}^{(s)} \check{g}_b^{(s)}$ with $\check{g}^{(s)} := B^{(s)t} D g$. With the lumped preconditioner $A_{bb}^{(s)}$ is simply replaced by $K_{bb}^{(s)}$. With the Dirichlet preconditioner, we have,

$$\check{z}_b^{(s)} := K_{bb}^{(s)} \check{g}_b^{(s)} - K_{bi}^{(s)} \tilde{z}_i^{(s)} \quad with \quad \tilde{z}_i^{(s)} := K_{ii}^{(s)-1} K_{ib}^{(s)} \check{g}_b^{(s)}. \tag{3.29}$$

The computation of $\tilde{z}_i^{(s)}$ amounts to solve a Dirichlet problem on the subdomain $\Omega^{(s)}$ with $-\check{g}_b^{(s)} = -B^{(s)t}Dg$ as the Dirichlet condition,

$$\begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} \tilde{z}_i^{(s)} \\ -\check{g}_b^{(s)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}. \tag{3.30}$$

Coarse system for the Schur approach. Since the Neumann problem of the Schur preconditioner is not always well-posed, this preconditioner is not optimal. The condition number is then: cond $(\bar{S}^{-1}S) = \mathcal{O}\left(\frac{1}{H^2}\left(1 + \left[\log\left(\frac{H}{h}\right)\right]^2\right)\right)$. This preconditioner is not optimal as it does not scale well with the number of subdomains. To improve it, Mandel (1993) proposes a strategy, called *the balancing method*, similar to the one of the FETI standard iteration. The preconditioning procedure with coarse variables may be written,

$$z := \bar{S}^{-1}g = \sum_s R^{(s)t}D^{(s)}S^{(s)+}D^{(s)}R^{(s)}g + G\gamma = \tilde{z} + G\gamma, \tag{3.31}$$

where $S^{(s)+}$ is a generalized inverse of $S^{(s)}$ and G contains the rigid body modes of all the floating subdomains and γ the set of amplitudes of them,

$$\begin{array}{c} G = \left[R^{(1)t}D^{(1)}N^{(1)} \quad ... \quad R^{(n_s)t}D^{(n_s)}N^{(n_s)}\right] \\ \gamma = \left[\gamma^{(1)t} \quad ... \quad \gamma^{(n_s)t}\right]^t \end{array}. \tag{3.32}$$

The determination of γ requires the solution of a global coarse system in order to improve the condition number of the conjugate gradient algorithm. This system is defined by the following optimization problem that consists in minimizing a specific norm of the gap between the inverse of the Schur matrix S^{-1} and its preconditioning one $\bar{S}^{-1}$,

$$\underbrace{Min}_{\gamma} \quad g^t(\bar{S}^{-1} - S^{-1})S(\bar{S}^{-1} - S^{-1})g. \tag{3.33}$$

Using the equality $G^t g = 0$, γ is solution of the coarse problem,

$$(G^t SG)\gamma = -G^t S\tilde{z}. \tag{3.34}$$

The preconditioning procedure may be summarized in the following formula replacing (3.26),

$$\bar{S}^{-1} = \left(I - G(G^t SG)^{-1}G^t S\right)\sum_s R^{(s)t}D^{(s)}S^{(s)+}D^{(s)}R^{(s)}. \tag{3.35}$$

The implementation requires two steps with a parallel treatment. The first one consists in computing $\tilde{z}$ using Neumann problems subdomain by subdomain (with regularizing for floating ones) (3.28). The second step is the solution of the coarse problem; but the preparation of the coarse matrix $G^t SG$, and specially the product SG is performed by solving Dirichlet problems with $G^{(s)}$ as the Dirichlet condition (see 3.15). With the balancing method, the condition number is now asymptotically optimal (cond $(\bar{S}^{-1}S) =$

$\mathcal{O}\left(1+\left(\log \frac{H}{h}\right)^2\right)$; that is, the preconditioner is then insensitive to the number of subdomains.

This "parallel" presentation of the primal and dual formulations shows that the two approaches comprise the same basic ingredients: Dirichlet and Neumann local problems and a global coarse problem. These ingredients are not located at the same step of a conjugate gradient type algorithm. Finally the scalability of the two methods is assured by the asymptotical behaviour of the condition number. The insertion of contact conditions in the reference problem introduces a non linear (and even non smooth) component and we have to define a strategy to combine a non linear solver with a domain decomposition method. Two extreme approaches may be developed with different features concerning at once the substructuring procedure and the adaptations of the algorithm.

3.2 Newton-Schur method for contact problems.

The Newton-Schur approach is a general combination of the Newton method as the non linear solver and the primal domain decomposition method as a linear solver. Such a strategy has been used by De Roeck and Le Tallec (1992) for large strain elasticity problems. The Generalized Newton Method coupled with an augmented lagrangean formulation of frictional contact problems (Alart and Curnier (1991)) may be coupled with the Schur method, but this imposes some important adaptations presented below.

Substructuring. The main feature of this non-linear domain decomposition strategy consists in distinguishing the physical contact interfaces from the numerical subdomain interfaces. Contrary to current approaches using dual or mixed methods (Ladevèze et al. (2002), Dostal et al. (1998), Dostal et al. (2000), Schoberl (2001)), we suggest to treat the physical contact interfaces inside the subdomains. Danek et al. (2005) develop a similar approach. This choice is flexible enough to deal with the diffuse non smoothness of the systems presented in the first section, because the decomposition of the mechanical system is not forced to respect the geometry of its components. It makes it possible to balance the size of the subdomains and get an optimal decomposition for parallel efficiency. The Figure 16 gives such a decomposition of a collection of deformable grains. This example is not the most relevant because the two types of interfaces intersect themselves, leading to specific treatment at the inersections. The Figure 17 provides a suitable substructuring of a cellular structure for which each cell is a subdomain, the (self)contact interfaces are the internal wall of the cells and numerical interfaces are in the middle of the walls.

Extra rigid body modes. Since the contact interfaces are inside the subdomains eventual extra rigid body modes may occur when the contact interfaces go through the subdomain. But the number of extra rigid body modes depends on the global contact status on these interfaces associated with the current Newton iteration. The rigid body modes are automatically detected by the local solver but it is interesting to foresee the maximal number. In the Figure 19 for a given global contact status given in the Figure 18, the extra rigid body modes are displayed taking into account some motions of grains, parts of grains or aggregates of parts of grains. The given status involves slip and stick statuses because friction is accounted for even if one needs further adaptations presented

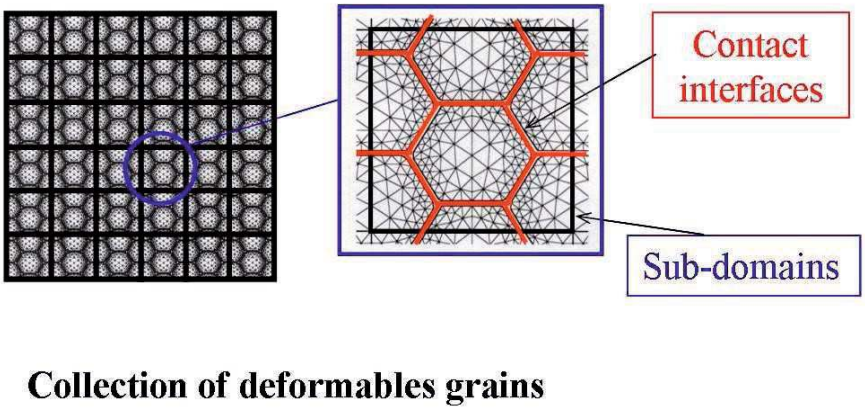

Figure 16. Contact (red) and numerical (black) interfaces for a collection of deformable grains.

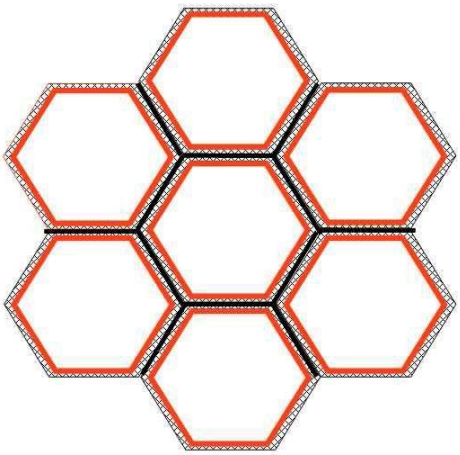

Figure 17. Contact (red) and numerical (black) interfaces for a cellular structure.

below.

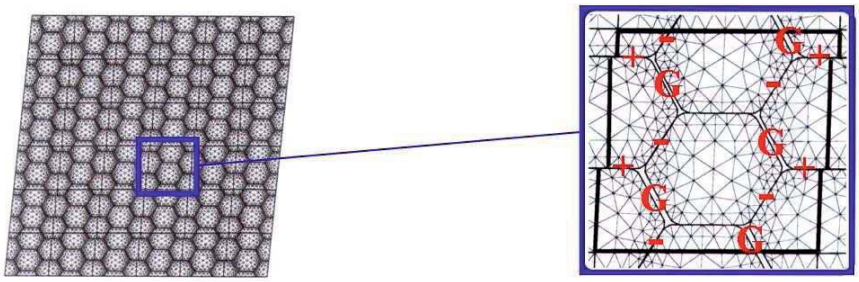

Figure 18. A global contact status on a subdomain: G for gap, + and - for forward slip and backward slip.

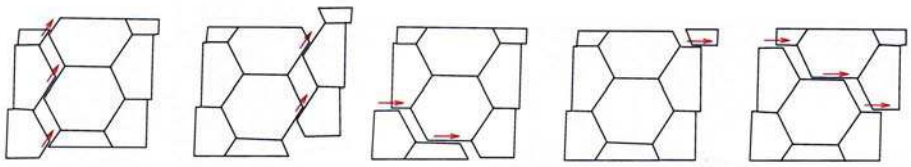

Figure 19. Extra rigid body motions for the global contact status of the previous figure.

A simple frictionless example with MAPLE. We develop in annex a simple frictionless example using the Newton-Schur method. But the interface linear Schur system is solved directly without carrying on a conjugate gradient algorithm and a parallel treatment. The reference problem is displayed in the Figure 20 where some degrees of freedom are underlined: squares for contact nodes, stars for multipliers. A decomposition in three subdomain is proposed; only the second subdomain is concerned with the contact. Some degrees of freedom are mentioned: squares for internal contact nodes, stars for multipliers and black circles for the nodes of the numerical interfaces.

Adaptations for frictional contact. The generalized Newton method coupled with an augmented Lagrangean formulation leads to non symmetric matrices (Alart and Curnier (1991)). The first adaptation consists in replacing the conjugate gradient algorithm by a GMRes algorithm (Saad and Schultz (1986)) that needs a single matrix-vector product as the conjugate gradient and then has the same parallel properties than it. The

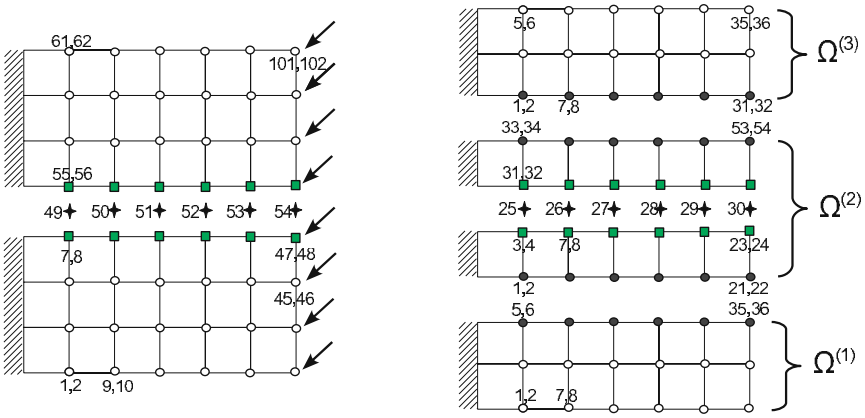

Figure 20. Frictionless contact reference problem and substructuring.

numerical experiments reported in the Figure 21 show that the behaviour of the iterative Schur complement solver is strongly perturbed when friction occurs, that is, when non symmetry is introduced in the tangent matrices. Indeed the minimization problem (3.33) is not well defined for non symmetric problems. The first idea is to replace the matrix S by the symmetrized matrix S^s ($S^s = (S + S^t)/2$). But a better choice is to use a symmetric matrix which has a mechanical meaning : the Schur complement matrix associated with a frictionless contact status by imposing μ equal to zero; this preconditioner is called *specific*.

In order to underline the influence of the non symmetry on the algorithm, we present in the Figure 21 the average number of GMRes iterations with respect to the friction coefficient varying from 0 to 2 for an example of a rolling shutter (for details refer to Ach and Alart (2001), Barboteu et al. (2001)) composed with 16 slats and 15x2 subdomains (26 floating subdomains). If the friction coefficient is zero the problem is symmetric and the different methods are equivalent and efficient. As soon as we introduce a small friction coefficient, the number of GMRes iterations increases quickly because nearly all contact elements have a slip status which introduce the non symmetry in the matrices. For μ equal to 0.2 we have the higher ratio of slip status. For μ larger than 0.2 the ratio of sticking status increases and the non symmetry decreases. For μ greater than 2, we have only stick status and we recover a symmetric problem. The algorithm with the two methods behaves like the evolution of the ratio of slip status, even if the friction coefficient is weak. Notice that the specific balancing method is less sensitive to the non symmetry : it requires twice less iterations than the standard one.

In order to avoid such a behaviour, a general *non symmetric preconditioner* has been proposed by Alart et al. (2000b) and Alart et al. (2000a). The presentation is quite technical and uses the concept of additive Schwarz method. In fact it consists in introducing the *dual rigid modes* as the kernel $N_*^{(s)}$ of the transpose matrix $K^{(s)t}$ of $K^{(s)}$ to define

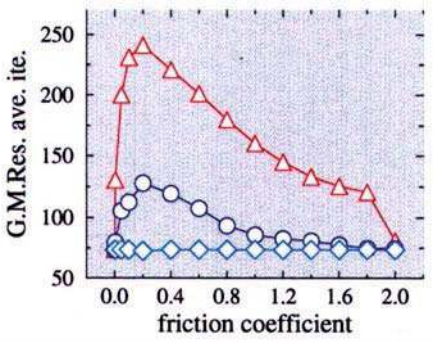

Figure 21. Influence of the friction on the behaviour of some preconditioned GMRes algorithms (standard $\triangle$, specific O, non symmetric $\Diamond$).

the matrix G_* replacing G,

$$G_* = \left[R^{(1)t} D^{(1)} N_*^{(1)} \quad \ldots \quad R^{(n_s)t} D^{(n_s)} N_*^{(n_s)} \right]. \tag{3.36}$$

But the regularizing of the local Neumann problem is performed using the standard rigid modes previously stored in the kernel $N^{(s)}$ of $K^{(s)}$; for instance the regularized matrix of the Neumann problem may be defined by $K^{(s)+} = (K^{(s)} + N^{(s)} N^{(s)t})^{-1}$. The preconditioner may be then summarized in the following formula,

$$\bar{S}^{-1} = \left(I - G_* (G_*^t S G_*)^{-1} G_*^t S \right) \sum_s R^{(s)t} D^{(s)} S^{(s)+} D^{(s)} R^{(s)}. \tag{3.37}$$

With this preconditioner we recover a behaviour of the GMRes algorithm independant of the friction coefficient as underlined in the Figure 21.

In a multiprocessor environment it is essential to analyze the scalability properties of the different Neumann-Neumann preconditioners, that is the convergence is not damaged by the number of subdomains. For the rolling shutters (Figure 22), we can verify that for a problem without friction ($\mu = 0$, symmetric problem), the 2-level Neumann-Neumann preconditioner has the expected behaviour (curve $*$). But with friction, the standard procedure leads to a high increase of the number of iterations (curve $\triangle$) with the number of subdomains. The results are even worse than without coarse solver (curve $\square$). The specific preconditioner (curve $\circ$) improves the convergence but is not optimal. On the other hand, the 2-level non symmetric Neumann-Neumann preconditioner (curve $\diamond$) leads to a full recovery of the numerical scalability properties obtained with a symmetric problem.

For the collection of deformable grains (Figure 23) the good behaviour of the non symmetric preconditionner is confirmed when the number of floating subdomains increases. This non symmetric procedure is more efficient than the standard and specific balancing method specially in presence of shear. Indeed, the friction (and then the non symmetry) plays a more important role in shear than in compression (Figure 23).

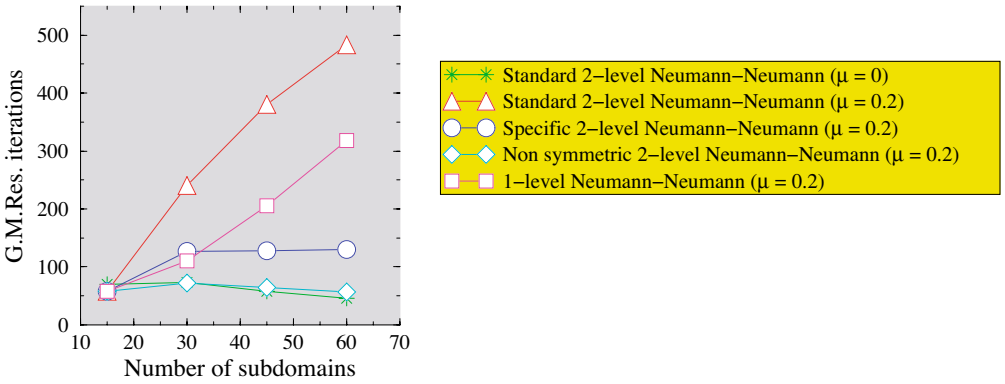

Figure 22. Numerical scalability of the preconditioners (rolling shutters).

To conclude this section subsequent works have shown the efficiency of the approach to tackle more complex problems. Alart et al. (2004) have coupled an arc-length technique with the Newton-Schur method to deal with large deformations in finite elasticity and multiple selfcontact in cellular media. Barboteu (2005) proposes an extension of the previous preconditioner to elastodynamical finite deformations.

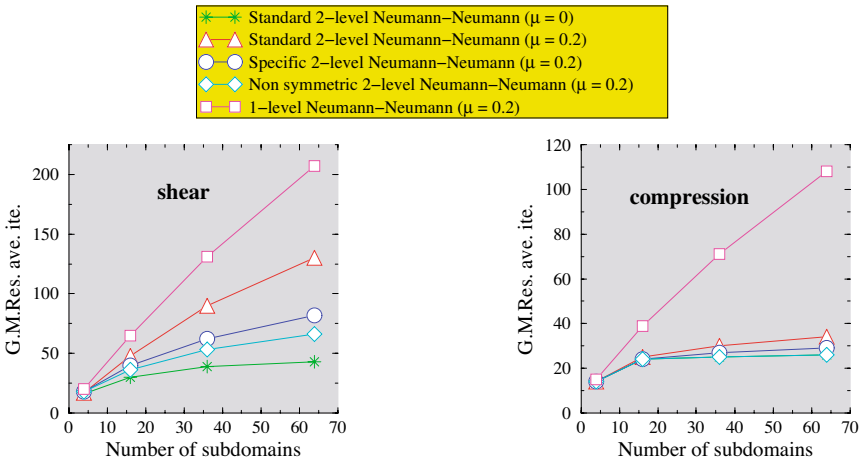

Figure 23. Numerical scalability of the preconditioners (deformable grains).

3.3 FETI-C method for contact problems.

The dual approach of the FETI method is based on Lagrange multipliers for enforcing the continuity conditions of the displacements on the interfaces. It is then natural to try to extend the FETI method to contact conditions on these interfaces. We present here the method detailled in Dureisseix and Farhat (2001) because the strategy consists in defining a single loop algorithm - called "one-shot iterative procedure". It is quite different from the Newton-Schur algorithm presented previously and from the other approaches based on the FETI method. For instance Dostal et al. (1998) and Dostal et al. (2000) propose two levels of iterations, the first one aimed at satisfying the contact conditions, and the second one at at satisfying the equilibrium equations. In the outer-iterations, an active zone of contact is updated by a mathematical programming technique. In the inner-iterations, a minimal subregion of the previously predicted area of contact is frozen and a state of equilibrium is sought after.

We try to compare the algorithm proposed by Dureisseix and Farhat (2001) with the projected conjugate gradient of the section 2 even if the context is quite different (structural versus granular problems).

Substructuring. The first consequence of the choice of a dual method concerns the substructuring. The number of subdomains must be larger than the given number of (elastic) bodies and foundations - that is, each body itself may be decomposed into several subdomains. In other words the minimal number of subdomains is imposed by the number of bodies. This leads to distinguish between two types of subdomain interfaces: the perfect (or numerical) interfaces, and potential contact interfaces. For a given subdomain $\Omega^{(s)}$ the associated interface is splitted into a perfect one and a contact one,

$$\Gamma^{(s)} = \Gamma_P^{(s)} \cup \Gamma_C^{(s)}. \tag{3.38}$$

For each subdomain the trace operator $B^{(s)}$ must be replaced by a new operator still noted $B^{(s)}$ defined as follows,

$$B^{(s)} : \Omega^{(s)} \rightarrow \Gamma^{(s)} \quad , \quad B^{(s)} : u_M^{(s)} \rightarrow \left\{ \begin{array}{ll} \pm u_M^{(s)} & if \quad M \in \Gamma_P^{(s)} \\ n_M^{(s)t} u_M^{(s)} & if \quad M \in \Gamma_C^{(s)} \\ 0 & if \quad M \notin \Gamma^{(s)} \end{array} \right. \tag{3.39}$$

where M is a generic node of $\Omega^{(s)}$ and $n_M^{(s)}$ an oriented normal to the interface at M - at the facing node M' on the neighbouring subdomain q the normal is $n_{M'}^{(q)} = -n_M^{(s)}$. Note that $B^{(s)}$ is no more a Boolean matrix, except when the contact interface is parallel to the reference axis. This operator may be split according to the two types of interface, $\breve{B}^{(s)} : \Omega^{(s)} \rightarrow \Gamma_P^{(s)}$ restricted to perfect interfaces, $\hat{B}^{(s)} : \Omega^{(s)} \rightarrow \Gamma_C^{(s)}$ restricted to contact interfaces. The crosspoints between four subdomains (in 2D discretisation) require specific treatments. For perfect interfaces we need usually six Lagrange multipliers to connect all the bodies. For four contact interfaces only four Lagrange multipliers have only to be introduced. (cf. Figure 24).

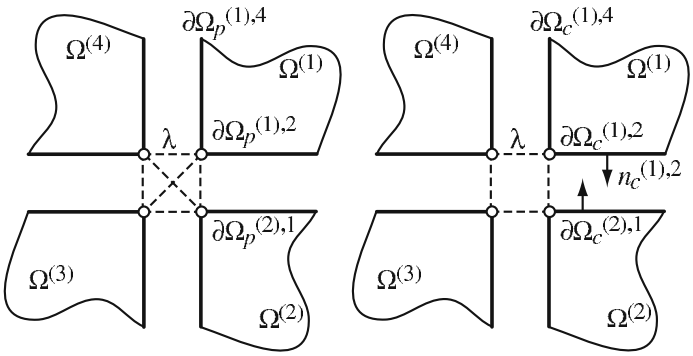

Figure 24. Corner Lagrange multipliers: perfect interfaces (left), contact interfaces (right), Dureisseix and Farhat (2001) .

Frictionless interface problems. The constrained problem replacing the system (3.7) involves both equality and inequality constraints,

$$\left\{ \begin{array}{l} K^{(s)}u^{(s)} = f^{(s)} - \breve{B}^{(s)t}\breve{\lambda} - \hat{B}^{(s)t}\hat{\lambda},\ s = 1, n_s \\ \sum_s \breve{B}^{(s)}u^{(s)} = 0 \\ \sum_s \hat{B}^{(s)}u^{(s)} \geq 0\ ,\ \hat{\lambda} \geq 0\ ,\ \hat{\lambda}^t \sum_s \hat{B}^{(s)}u^{(s)} = 0 \end{array} \right. \tag{3.40}$$

For the sake of simplicity no initial clearance is taken into account in this formulation; the Lagrange multipliers $\hat{\lambda}$ are interpreted as contact forces restricted to its normal components. The constrained quadratic dual problem extending (3.14) associated with this formulation extending is then,

$$\underbrace{Min}_{\breve{\mu}\ ,\ \hat{\mu}\geq 0} \quad \frac{1}{2}\ \breve{\mu}^t \breve{F}_I \breve{\mu} - \breve{d}^t\breve{\mu} + \frac{1}{2}\ \hat{\mu}^t \hat{F}_I \hat{\mu} - \hat{d}^t\hat{\mu}. \tag{3.41}$$

In concatenating all the multipliers in a single vector, $\mu := [\breve{\mu}^t, \hat{\mu}^t]^t$, the minimisation problem is formally simplified,

$$\underbrace{Min}_{\mu\geq 0} \quad \frac{1}{2}\ \mu^t F_I \mu - d^t\mu. \tag{3.42}$$

The FETI-C non linear solution algorithm. Although they use a preconditioned conjugate projected gradient algorithm for solving the FETI problem with equality constraints, Dureisseix and Farhat (2001) introduce the treatment of the inequality constraints, not with the projection on constraints set or on its tangent cone, but with the concept of active set. This leads to a detailled but complex presentation. We follow here this presentation but we try to compare it with the algorithm presented in the Table 6, to identify the common steps and the differences. Dureisseix and Farhat (2001) define

the set of active contact interfaces $\tilde{\Gamma}_C$ depending on the current value of the multiplier $\hat{\lambda}_M$ at node M,

$$M \in \tilde{\Gamma}_C \quad if \quad \hat{\lambda}_M > 0. \tag{3.43}$$

An "active" trace operator $\tilde{B}^{(s)}$ is then defined depending on the current $\hat{\lambda}$,

$$\tilde{B}^{(s)} : u_M^{(s)} \rightarrow \begin{cases} \pm u_M^{(s)} & if \quad M \in \Gamma_P^{(s)} \\ n_M^{(s)t} u_M^{(s)} & if \quad M \in \tilde{\Gamma}_C^{(s)} \\ 0 & if \quad M \in \Omega^{(s)} \cup (\Gamma_C^{(s)} / \tilde{\Gamma}_C^{(s)}) \end{cases} \tag{3.44}$$

It follows the different modified "active" definitions,

$$\begin{gathered} \tilde{G}_I = \left[\tilde{B}^{(1)} N^{(1)} \quad ... \quad \tilde{B}^{(n_s)} N^{(n_s)} \right] \\ \tilde{P} := I - \tilde{G}_I (\tilde{G}_I^t \tilde{G}_I)^{-1} \tilde{G}_I^t. \end{gathered} \tag{3.45}$$

and curiously,

$$\tilde{F}_I = \sum_s B^{(s)} K^{(s)+} \tilde{B}^{(s)t}. \tag{3.46}$$

Dureisseix and Farhat (2001) note that the operator $\tilde{F}_I$ is unsymmetric, but it follows from the previous definitions that the set of admissible Lagrange multipliers on which $\tilde{F}_I$ operates is such that,

$$\tilde{B}^t \lambda = B^t \lambda. \tag{3.47}$$

The resulting FETI-C solver for frictionless contact problems is summarized in the Table 8. It is interesting to compare this algorithm with the one of the Table 6. The projection procedure and the "primal planing" (so called in Avery et al. (2004)) correspond to the projection of the residual on the tangent cone to the constraints set. The re-projection procedure does not exist in the Table 6 because the projection of the gradient occurs after the preconditioning. On the other hand, in the table 6, the projection operates on the residual and on the previous descent direction, since the projection concerns only the residual in the Table 8. Conjugating of the gradients and updating of the iterate are similar. The "dual planing" procedure corresponds to the projection of the iterate on the constraints set, but this is an iterative procedure to enforce the self-equilibrium of each subdomain simultaneously with the projection that may modify the coarse problem. This procedure is described in the Table 9.
Note that the step 2.a of the planing procedure incorporates a coarse problem modifying the active contact zone. Experience shows that in general, the planing procedure converges after a small number of subiterations. The complexity of this algorithm is due to the choice of a single loop algorithm to deal with the inequality constraints issued from contact and with the equality constraints deriving from the floating subdomains. The approach developed above has been recently extended to frictional contact.

Table 8. FETI-C algorithm.

$$
\text{Initialize} \left\{ \begin{array}{l} \tilde{\Gamma}_C^0 = \Gamma_C \\ \bar{\lambda}^0 = G_I (G_I^t G_I)^{-1} e \\ \lambda^0 = Planing(\bar{\lambda}^0) \\ \tilde{P}^0 = \tilde{P}(\tilde{\Gamma}_C^1) \\ r^0 = d - \tilde{F}_I^1 \lambda^0 \end{array} \right.
$$

$$
\left[\begin{array}{l} Iterate \;\; k = 1, 2, ... \;\; until \;\; convergence \\ Project \;\; w^{k-1/2} = \tilde{P}^{(k-1)t} r^{k-1} \\ Primal \;\; planing \;\; w^{k-1} = \left\{ \begin{array}{ll} w^{k-1/2} on & \Gamma_P \\ < w^{k-1/2} >_+ & on \;\; \tilde{\Gamma}_C^k \end{array} \right. \\ Precondition \;\; z^{k-1} = \tilde{\bar{F}}_I^{-1} w^{k-1} \\ Re - project \;\; y^{k-1} = \tilde{P}^{(k-1)} z^{k-1} \\ Conjugate \;\; p^k = y^{k-1} + \zeta^k p^{k-1} \;\; with \;\; \zeta^k = \frac{y^{k-1}.w^{k-1}}{y^{k-2}.w^{k-2}} \\ Update \;\; \lambda^{k-1/2} = \lambda^{k-1} + \eta^k p^k \;\; with \;\; \eta^k = \frac{p^{k-1}.w^{k-1}}{p^k.\tilde{F}_I^k p^k} \\ Dual \;\; planing \;\; \lambda^k = Planing(\lambda^{k-1/2}) \\ Compute \; residual \;\; r^k = \left\{ \begin{array}{l} r^{k-1} - \eta^k \tilde{F}_I^k p^k if \; no \; status \; change \\ d - \tilde{F}_I^{k+1} \lambda^k \;\; if \;\; status \;\; change \end{array} \right. \end{array} \right.
$$

Table 9. Dual planing procedure.

$$
\begin{array}{l} Iterate \;\; n = 1, 2, ... \;\; until \;\; convergence \\ 1.a \;\; Project: \;\; \lambda^n \leftarrow \left\{ \begin{array}{ll} \lambda^n & on \;\; \Gamma_P \\ < \lambda^n >_+ & on \;\; \Gamma_C \end{array} \right. \\ 1.b \;\; Active \; contact: \;\; determine \; \tilde{\Gamma}_C^n \;\; from \; \lambda^n \;\; and \; update \tilde{G}_I^n \\ 1.c \;\; Test \; equilibrium: \; if \;\; \tilde{G}_I^{nT} \lambda^n = e \;\; end \\ 2.a \;\; Self - equilibrium: \;\; \lambda^n \leftarrow (I - \tilde{G}_I^n (\tilde{G}_I^{nT} \tilde{G}_I^n)^{-1} \tilde{G}_I^{nT}) \lambda^n + \tilde{G}_I^n (\tilde{G}_I^{nT} \tilde{G}_I^n)^{-1} e \\ 2.b \;\; Test \; positiveness: \;\; if \; \exists M \in \Gamma_C \; \lambda_M^n < 0 \; go \; to \; 1.a \end{array}
$$

4 Multiscale approach for diffuse non smoothness

Domain decomposition methods, as presented above, provide a general framework to the multiscale numerical analysis of structures. Indeed, substructuring appears as a naturally multiscale numerical strategy where each substructure is related to an intermediate scale between the basic components inside the subdomains and the global structure. Moreover multiscale approaches can enrich the substructuring in incorporating either a micro-macro splitting of some variables (Ladevèze and Dureisseix (2000)) or even an homogenization procedure (Ladevèze et al. (2001)). By this way the domain decomposition approaches are not only efficient numerical methods to solve large-scale problems but may also provide a *numerically homogenized behavior* of each substructure, which is useful from a mechanical point of view. We present in this section a first attempt to extend the LATIN (Large Time Increment) Ladevèze and Dureisseix (2000) micro-macro approach, initially developed for continuous media (Ladevèze and Dureisseix (2000)), to strongly nonsmooth discrete systems. This strategy is applied to the tensegrity structure presented in section 1 whose the equilibrium is characterized by the LCP expressed in (1.20). For details refer to Nineb et al. (2007).

4.1 Substructuring strategy

The first step of the problem reformulation consists of a decomposition of the structure into substructures and interfaces (see Figure 25). Each of these components possesses its own variables and equations. We can proceed in two ways: either nodes are distributed among substructures, and interfaces are links joining a substructure to another, or links are distributed among substructures, and interfaces are nodes joining a substructure to another. Only this last case is considered herein (see Figure 25).

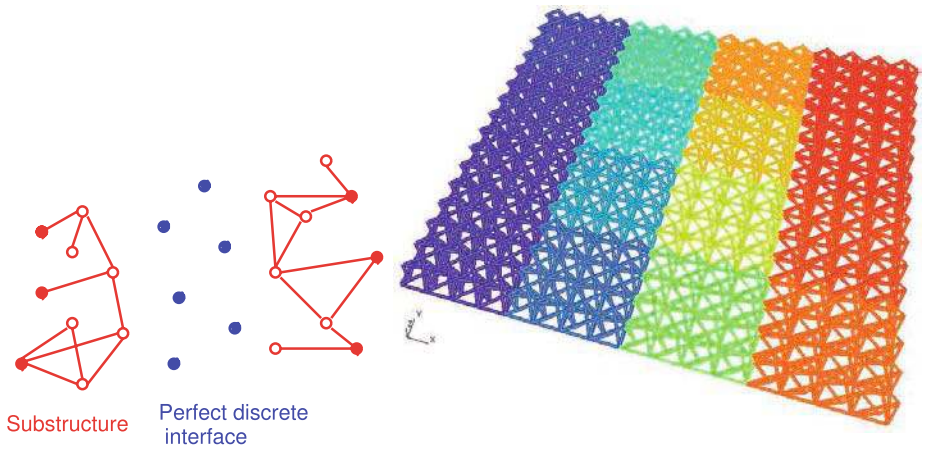

Figure 25. Discrete interface (left) and substructuring of a tensegrity grid (right) with 16 subdomains.

Table 10. Substructuring notations

F_E, F_E^d	Internal and external nodal forces to substructure E
U_E, U_E^d	Internal and external nodal displacements to substructure E
$\Gamma_{EE'}$	Interface between substructures E and E'
Γ	Global interface
$C_{E\Gamma}$	Boolean mapping matrix of substructure boundary dofs
$F_{E\Gamma}$	Forces of interface Γ acting on substructure E
$F_{EE'}$	Forces of interface $\Gamma_{EE'}$ acting on substructure E
$U_{EE'}$	Displacement of $\Gamma_{EE'}$ nodes connected to E
$U_{E\Gamma}$	Displacement on substructure E boundary

A substructure E is submitted to the action of its neighbouring interfaces $\Gamma_{EE'}$: forces $F_{EE'}$ and displacements $U_{EE'}$. Extended to all the interfaces local to the substructure E, the assembling of the previous fields are denoted with $F_{E\Gamma}$ and $U_{E\Gamma}$. An interface $\Gamma_{EE'}$ transfers the forces $F_{EE'}$ and the displacements $U_{EE'}$ on each of its sides.

The solution $s = \bigcup_E s_E$ with $s_E = (e_E, t_E, U_{EE'}, F_{EE'})$ of the reference problem must satisfy to:

- the balance equation (local version of (1.7):

$$-F_E + F_E^d + C_{E\Gamma}^t F_{E\Gamma} = 0 \tag{4.1}$$

  where $F_E = B_E^t \tau_E$;
- the strain admissibility:

$$\left\{ \begin{array}{l} U_{E\Gamma} = C_{E\Gamma} U_E \\ e_E = B_E U_E \\ U_E \mid_{\Gamma_u} = U_E^d \end{array} \right. \tag{4.2}$$

- the constitutive relations (1.10), (1.14);
- the interface behavior:
  force balance:

$$F_{EE'} + F_{E'E} = 0 \tag{4.3}$$

  continuity of displacements:

$$U_{EE'} = U_{E'E} \tag{4.4}$$

The interfaces exhibit a perfect behavior because of the continuity of the displacements and because the nonsmoothness is localized within the substructures. This modelling choice is identical to the one of Barboteu et al. (2001) and somehow the dual of the one proposed by Ladevèze et al. (2002) where the nonlinearity (contact in crack) is isolated in the interfaces. This substructuring strategy can be easily extended to the dynamical behaviour and even to granular materials where the nodes are the mass centers of the grains and the links are the punctual contacts. But some relations have to be modified according to a velocity–impulse formulation.

4.2 A micro-macro LATIN method

In order to take into account the multiscale aspect of the behavior of a large scale tensegrity structure, especially when it is designed as an assembly of identical modules, a suited numerical strategy has to be settled. A multilevel domain decomposition approach can tackle this task, when the so-called coarse space is related to an homogenized model of the structure as considered by Farhat and Roux (1991). We choose herein to follow the approach proposed by Ladevèze and Dureisseix (2000) and completed by Ladevèze et al. (2001) that was designed for continuum media, and to extend it to discrete systems.

Table 11. Micro-macro notations

m, M	Denote micro and Macro subscripts
l	Search direction parameter per substructure
d	Search direction parameter per interface
$f_{EE'}$	Macro generalized forces of interface $\Gamma_{EE'}$ acting on substructure E
$f_{E\Gamma}$	Macro generalized forces of interface Γ acting on substructure E
$u_{EE'}$	Macro generalized displacements of $\Gamma_{EE'}$ nodes connected to E
$u_{E\Gamma}$	Macro generalized displacements on substructure E boundary
$R^t_{EE'}$	Projector onto generalized macro space

Micro-macro description. Once a substructuring has been performed, the first step is to describe the media from a microscopic and a macroscopic point of view. These descriptions arise from the mechanical fields lying on each interface $\Gamma_{EE'}$ independently. Both displacement $U_{EE'}$ and forces $F_{EE'}$ are split into two parts: the macro part, denoted with a superscript M, and the additional micro part, denoted with a superscript m. Therefore, one gets

$$F_{EE'} = F^M_{EE'} + F^m_{EE'} \qquad \text{and} \qquad U_{EE'} = U^M_{EE'} + U^m_{EE'} \tag{4.5}$$

The micro and macro spaces must be uniquely defined: they should be "orthogonal" in a way an orthogonal projector can be used on each subspace. The energy splitting is used in such a way:

$$\sum_{EE'} (F_{EE'})^t \; U_{EE'} = \sum_{EE'} (F^M_{EE'})^t \; U^M_{EE'} + (F^m_{EE'})^t \; U^m_{EE'} \tag{4.6}$$

Superscript t denotes the L_2 transposition: this energy measure is specific to discrete systems.

Macro representation. The macro part lies in a small sized subspace, therefore the macro fields are described with few parameters; a basis of macro fields can be chosen as: $F^M_{EE'} = R_{EE'} f_{EE'}$ where $f_{EE'}$ stores the macro parameters for forces and $R_{EE'}$ is the set of basis vectors. For sake of simplicity, the corresponding macro parameters for the

displacement field, $u_{EE'}$ is chosen such that the same vector basis holds:

$$U^M_{EE'} = R_{EE'} u_{EE'} \qquad \text{and} \qquad F^M_{EE'} = R_{EE'} f_{EE'} \tag{4.7}$$

The macro parameters can be selected depending on the problem one has to model; herein, we choose the generalized averages of fields $F_{EE'}$ and $U_{EE'}$ on $\Gamma_{EE'}$, up to the order 1. For 3D analysis, we get:

- for $f_{EE'}$, the 3 resultants of forces, the 3 moments, the 2 tensions and the shear in the interface plane, and the dilatation in the interface plane;
- for $u_{EE'}$, the corresponding generalized averages are the 3 mean translations, the 3 mean rotations, the 2 stretchings and the 'shearing' in the interface plane, and the expansion in the interface plane.

Therefore, 10 parameters are needed per interface for generalized macro force fields, as well as for the generalized macro displacement fields. As an additional simplification, the basis $R_{EE'}$ is orthonormalized:

$$R^t_{EE'}\ R_{EE'} = \mathbf{1}$$

to get

$$(F^M_{EE'})^t\ U^M_{EE'} = f^t_{EE'} R^t_{EE'} R_{EE'} u_{EE'} = f^t_{EE'} u_{EE'}$$

The consequences when using (4.6) are the expressions of the orthogonal projector onto the macro space:

$$f_{EE'} = R^t_{EE'}\ F_{EE'} \qquad \text{and} \qquad u_{EE'} = R^t_{EE'}\ U_{EE'} \tag{4.8}$$

and the orthogonality: $R^t_{EE'}\ U^m_{EE'} = 0 = R^t_{EE'}\ F^m_{EE'}$.

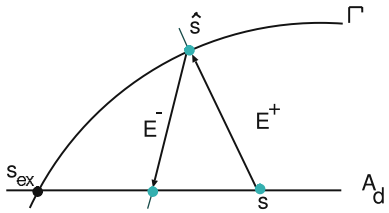

Figure 26. LATIN method principles

LATIN method. The LATIN method, introduced by Ladevèze (1999), is used as a solver to find the solution of the previous problem. Briefly, it consists of several steps:

- the equations are split into two sets: (i) the admissible set $\mathbf{A_d}$ with balance equations per substructure (4.1) and strain compatibility per substructure (4.2), and (ii) the constitutive relation set $\mathbf{\Gamma}$ with link behaviors (4.3) and interface behavior (4.4). In the present case, $\mathbf{\Gamma}$ is not differentiable due to the diffuse non smoothness of cable behavior;

- an iterative procedure producing alternatively a solution in $\mathbf{A_d}$, and a solution in $\mathbf{\Gamma}$, using search directions $\mathbf{E}+$ and $\mathbf{E}-$, see Figure 26. The solution of the problem, $s_{\text{ex}} = \{(e_E, \tau_E)\ ;\ (U_{EE'}, F_{EE'})\}$ is the intersection of $\mathbf{A_d}$ and $\mathbf{\Gamma}$.
Building $\hat{s}$ once s is known, is the so-called *local stage*; it involves positive scalar search direction parameters d and l:

$$(\hat{\tau}_E - \tau_E) + l(\hat{e}_E - e_E) = 0 \tag{4.9}$$

and

$$(\hat{F}_{EE'} - F_{EE'}) - d(\hat{U}_{EE'} - U_{EE'}) = 0 \tag{4.10}$$

Building s once $\hat{s}$ is known, is the so-called *linear stage*: it involves the same parameters as soon as the directions $\mathbf{E}+$ and $\mathbf{E}-$ are conjugate: $(\tau_E - \hat{\tau}_E) - l(e_E - \hat{e}_E) = 0$ and $(F_{EE'} - \hat{F}_{EE'}) + d(U_{EE'} - \hat{U}_{EE'}) = 0$.
Such a choice for the search directions ensures the convergence of the method, see Ladevèze (1999).

4.3 Local stage — local non smooth solvers

As previously mentioned, this stage consists in finding $\hat{s}$ once s is known from the previous stage. It leads to local and linear problems on (perfect) interfaces $\Gamma_{EE'}$, and diffuse non smooth local linear complementary problems (LCP) on substructures Ω_E.

The perfect behavior of the interface $\Gamma_{EE'}$, coupled with search directions using the parameters d_m and d_M, leads to an explicit expression (a similar expression is obtained for the macro part),

$$\begin{cases} \hat{U}^m_{E'E} = \hat{U}^m_{EE'} = \dfrac{1}{2}[(U^m_{EE'} + U^m_{E'E}) - \dfrac{1}{d^m}(F^m_{EE'} + F^m_{E'E})] \\ \hat{F}^m_{E'E} = -\hat{F}^m_{EE'} = \dfrac{1}{2}[(F^m_{E'E} - F^m_{EE'}) - d^m(U^m_{E'E} - U^m_{EE'})] \end{cases}$$

For substructure Ω_E, using a scalar search direction parameter l, the problem is independent for cables and bars. For bars, $(\hat{e}_b, \hat{\tau}_b)$ must verify the constitutive relation (1.10) and the search direction (4.9). One gets:

$$\begin{cases} \hat{e}_b = (k_b + l)^{-1}(le_b + \tau_b - k_b e^0_b) \\ \hat{\tau}_b = k_b(\hat{e}_b + e^0_b) \end{cases}$$

For cables, $(\hat{e}_c, \hat{\tau}_c)$ must verify the constitutive relation (1.14). The resulting problem is a LCP once the change of variables is used, to get:

$$\begin{cases} \hat{\lambda}_c - (k_c^{-1} + l^{-1})\hat{\tau}_c = \tilde{\lambda} \\ 0 \le \hat{\tau}_c \perp \hat{\lambda}_c \ge 0 \end{cases}$$

where $\tilde{\lambda} = -[l^{-1}\tau_c + e_c + k_c^{-1}t^0_c]$ is known at this stage. Its solution is

$$\begin{cases} \hat{\lambda}_c = <\tilde{\lambda}>_+ \\ \hat{\tau}_c = -(k_c^{-1} + l^{-1}) <\tilde{\lambda}>_- \end{cases}$$

Note that this is the simplest choice for the local stage. Subsequent work will deal with a version where admissibility per substructure is enforced at the local stage, leading to a LCP coupling bars and cables, and for which there exists efficient solvers. Such a version is under development, in the same spirit as to improve direct linear global solvers with domain decomposition techniques by maintaining them for the smaller problems of independent substructures, and coupling them with iterative procedures on interface only, see Le Tallec (1994) and Farhat and Roux (1991). Indeed, this would replace global LCP solvers on the whole problem with local LCP solvers per substructure, as the one used in Section 2.

4.4 Linear stage — macro homogenized problem

This stage is similar to the one described in Ladevèze et al. (2001). Once $\hat{s}$ is known from the previous stage, the linear stage consists in finding s that satisfies the balance equations and strain admissibility per substructure (4.1), (4.2). Considering the macro part, an additional constraint is to enforce macro continuity of the fields at interfaces: with a force-oriented approach, this constraint is to prescribe, on each interface $\Gamma_{EE'}$,

$$F^M_{EE'} + F^M_{E'E} = 0 \tag{4.11}$$

To avoid an overconstrained problem, the search direction requires a weak formulation. The presentation of this part is quite technical and uses the same ingredients than the approach developed for heterogenous structures (see Ladevèze et al. (2001)), because the interfaces are also perfect without perturbation due to the non smooth relations isolated inside the subdomains. Nevertheless, due to the discrete nature of the problem, an algebraic formulation needs to be derived, that is detailled by Nineb et al. (2007). This leads to a macro homogenized problem enriching the coarse problem of the domain decomposition methods according to the macro representation introduced in Section 4.2.

4.5 A first test on a tensegrity system

Problem settings. We consider as a test case a tensegrity grid obtained with the duplication of a self stressed elementary module Quirant et al. (2003). Such a module is composed of 8 nodes, 12 cables and 4 bars, see Figure 27 on the left. The characteristic parameters of this module are given in Table 12. Prestressing is such that this module is selfstressed, i.e. $Bt^0 = 0$. Therefore, any assembly of such modules will be self balanced automatically.The tested tensegrity grid possesses $16 \times 16 = 256$ modules, it is split into $4 \times 4 = 16$ substructures (each containing 16 modules) and 24 strong interfaces, see Figure 25.

As a comparison point of view, a similar substructuring of a continuum media plate would lead to 16 substructures and only 24 interfaces. As boundary conditions, the lower nodes on two opposite edges are clamped, and a uniform vertical force field $F^d = 40$ N is prescribed on every node.

Choice of search directions. There are two parameters for this approach that define the search directions: d^m and l. For the parameter l, we used the simplest choice: l

Table 12. Characteristic parameters

$H = 0.5$ m	Module height
$L = 1$ m	Module length
$S_c = 0.5\ 10^{-4}$ m^2	Cable section
$E_c = 10^{11}$ Pa	Cable Young modulus
$S_b = 2.8\ 10^{-4}$ m^2	Bar section
$E_b = 2\ 10^{11}$ Pa	Bar Young modulus
$t_c^0 = 2000$ N	Lower cables prestress
$t_c^0 = \sqrt{2} \times 2000$ N	Upper cables prestress
$t_c^0 = \sqrt{(1 + 4\frac{H^2}{L^2})} \times 2000$ N	Bracing cables prestress
$t_b^0 = -\sqrt{(5 + 4\frac{H^2}{L^2})} \times 2000$ N	Bar prestress

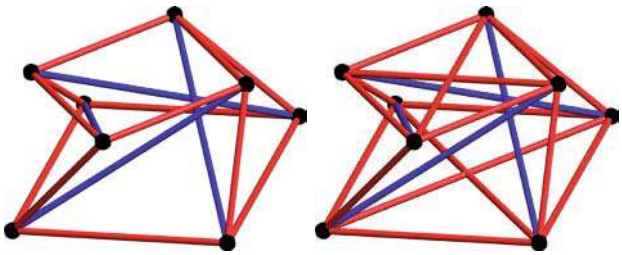

Figure 27. Used modules (initial module on the left and modified module on the right)

is the stiffness of the links of the underlying networks of cables and bars, i.e. $l = k_c$ for the cables, and $l = k_b$ for the bars. Concerning the parameter d^m at the interfaces, a global stiffness of a module has been computed, and the corresponding values of an equivalent stiffness arising from the interfaces alone has been chosen. This procedure has the advantage of being automatic, though the obtained value is not exactly the optimal one. In the present case, it leads to $d^m = 3.92\ 10^6$ N/m. For all of the subsequent simulations, these values have been selected once for all, and not been changed in all of the following.

Numerical results. For the considered test case, the loading level is close to, but less than its ultimate limit value for which there is a lack of stability of the whole structure (similar to global buckling in continuum media and corresponding herein to mechanism occurring).

We consider several loading amplitudes αF_d with $0 \leq \alpha \leq 1$. For such values of α, the simulation is performed in one step from the reference configuration Ω_0 to the current one Ω_α, without time stepping. This can be done for the considered problem because it is not an evolution-type one: the final solution does not depend on the loading path.

Figure 28 shows the values of τ_c on each of the 3072 cables, for the different values of α, sorted by increasing values. For $\alpha = 0$, one recovers the values τ_c^0 in the cables (cf Table 12). Obviously, as α increases, the stress redistribution is larger and larger: the number of slack cables increases (as well as the maximum value of internal tensions τ_c) to reach about 14 % of the whole set of cables when $\alpha = 1$. For this value of loading, the structure is still within its stable domain for which it still possesses a stiffness reserve. Such simulations are useful to check the integrity of such a structure under extreme loading conditions above normal service usage for which, in general, one assesses that no cable slackens; if this is the case, the strength of the structure could be endangered when the load decreases again and when slacken cables suddenly reload: the rapid change in local apparent stiffness lead to dynamical loadings that can damage the nodes. Figure 29 shows the non linear evolution of this fraction of slack cables when the loading increases.

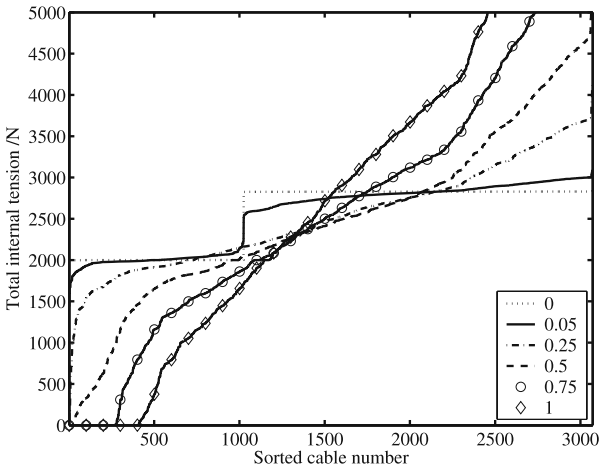

Figure 28. Internal tension in cables for different values of the loading parameter α

Concerning the convergence of the algorithm, Figures 30 reports the evolution of the error e along the iterations, for several values of the loading parameter α. For small values of α, no cable slackens and the convergence is very similar to the one of the linear case. When α increases, i.e. when the number of slack cables increases, the convergence is affected and soon exhibits two different rates: an initial one, and an asymptotic one for small values of the energy error (here, about 10^{-4}). Such a behavior has to be investigated thoroughly, for instance with a max norm of the error, its projection on the eigenmodes...

Moreover, the proposed approach does not exhibit a constant convergence rate with respect to the non smoothness ratio, defined as the fraction of slack cables of Figure 29. Indeed, when the number of slack cables increases, the tangent stiffness of numerous links can be largely changed. A first improvement of the approach would consist in adapting the search direction l to the local status of cables, but it leads to re-factorizations of

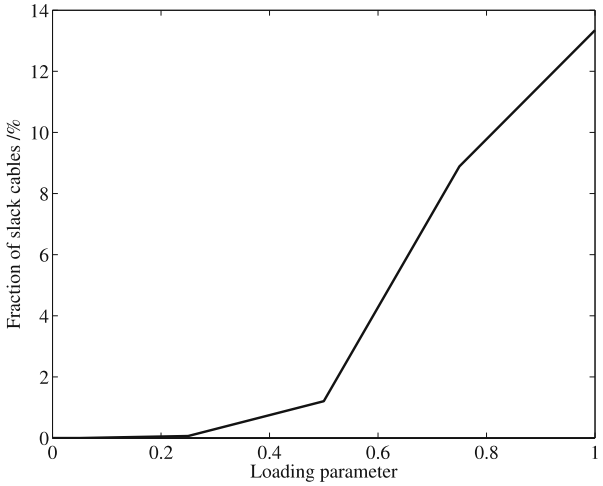

Figure 29. Fraction of slack cables vs loading parameter α

the subdomain stiffness matrices, which is a costly part of the algorithm. Doing so only few times during iterations could recover a constant convergence rate along iterations, but probably not along different values of α. A second possibility, mentioned above and which is under development, is to use a global search direction per subdomain, adding local admissibility to the local stage problem. In this case, one gets a global LCP problem per subdomain.

Comparison with a monodomain approach. To compare the methods, we use the same algorithm on the same test problem with a unique domain, and therefore without any multiscale feature. In this case, the LATIN approach becomes equivalent to an augmented Lagrangian approach; this is also the case for unilateral conditions arising from frictionless contact, see Alart et al. (2006). The evolution of the error with respect to the number of iterations is shown on Figure 31 for a loading parameter $\alpha = 1$. Obviously, the monodomain exhibits a higher convergence rate, which is not surprising if one consider the case of a linear problem, for which the monodomain method is a direct method, while domain decomposition methods require iterations to converge. Therefore, one needs a cost evaluation to compare the two approaches.

Each approach consists of two distinct phases: (i) the initialization phase where the costs are dominated by factorizations of the stiffness matrices, (ii) the iteration phase where, for each iteration, the cost is related on one hand to the local stage for solving the links behaviors, and on the other hand, for solving linear problems (with forward and backward substitutions); during this second phase, the cost is usually dominated by the linear solves and the local stage has the same cost for both approaches.

For the initialization phase, the monodomain approach requires the factorization of

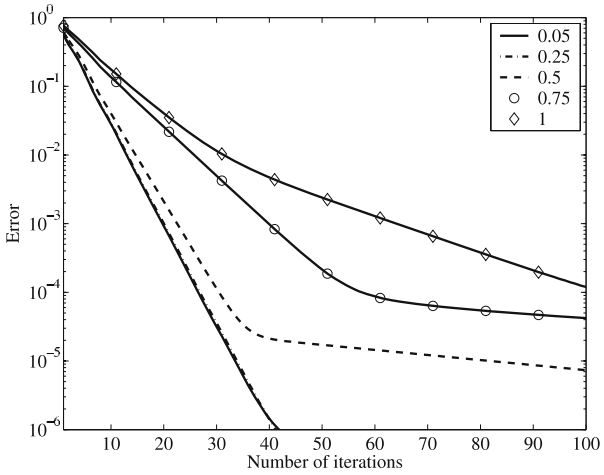

Figure 30. Evolutions of the error with respect to a reference solution, for different values of the loading parameter α

the global stiffness matrix, while the multiscale approach requires the factorization of all the local substructure matrices, as well as the macroscopic problem matrix. For the same reasons as the domain decomposition methods outperform direct solvers, the cost of initialization is always higher for the monodomain approach.

For the iteration phase, the cost of one linear solve for the monodomain approach is (in terms of floating point operations) nl, where n is the number of dofs, and l is the bandwidth of the global stiffness matrix. Considering the proposed test case, $n = 2500$ and $l = 200$ (using a reverse Cuthill-McKee renumbering scheme). The number of substructures is $N_S = 16$, the number of dofs for the local substructure stiffness matrix is $n_S = 195$ and its bandwith $l_S = 50$; finally, the number of dofs for the coarse problem is $N = 216$ and its bandwidth $L = 60$. The ratio of the costs for 1 iteration of the monodomain approach and 1 iteration of the multiscale approach (on a monoprocessor machine) is

$$a_{\text{iter}} \approx \frac{nl}{N_S n_S l_S + NL} = 2.9$$

The ratio of convergence rates obtained on Figure 31 to reach an error threshold of 1 % is also close to 3: during the iteration process, the two algorithms are of the same efficiency. The potential gain therefore lies in the initialization phase, for which the factorization cost ratio is

$$a_{\text{init}} \approx \frac{\frac{1}{2}nl^2}{\frac{1}{2}N_S n_S l_S^2 + \frac{1}{2}NL^2} = 11.6$$

When the problem size increases, if the convergence rate is maintained, we can expect an increase in the efficiency. Indeed, the factorization costs do not increase in the same manner for each approach (typically, $n_S \approx n/N_S$, and $l_S \approx l/\sqrt{N_S}$).

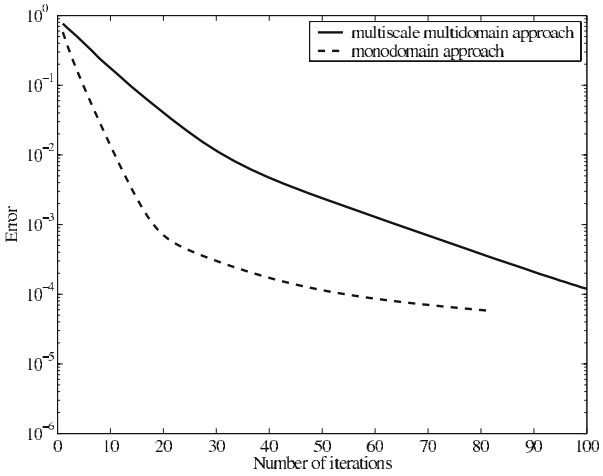

Figure 31. Comparison of error evolutions for different approaches

5 Conclusion

The contact treatment in a multiprocessor environment depends on the problem in which the contact occurs, essentially on the ratio of degrees of freedom concerned by non smooth contact conditions.

If the contactless problem associated with the contact problem requires itself a parallel computing strategy, the multiprocessor approach to deal with contact has to adapt the previous strategy. But the choice of a strategy is a modelling choice depending on the problem itself. For instance the contact area may be located inside the subdomain (section 3.2) or on the interfaces between the subdomains (section 3.3).

The parallel computing may be carried out at different levels in a software architecture. It may consist in extending linear solvers to non linear and even non smooth ones: Gauss Seidel to Non Linear Gauss Seidel (section 2.1), Conjugate Gradient to Projected Conjugate Gradient (section 2.2), FETI to FETI-C (section 3.3). But the contact treatment may be also taken into account in a more general parallel computing strategy to solve non linear large scale problems: Newton-Schur method (section 3.2), LATIN method (section 4).

Finally we can predict that the panel of strategies could still increase with the variety of large scale mechanical problems to tackle.

Bibliography

K. Ach and P. Alart. Numerical simulation of a multi-jointed structure. In *Phylosophical transactions of the Royal Society of London*, volume A 359, pages 2557–2573, 2001.

P. Alart. Méthode de Newton généralisée en mécanique du contact. *J. Math. Pures Appl.*, 76:83–108, 1997.

P. Alart and A. Curnier. A mixed formulation for frictionnal contact problems prone to Newton like solution methods. *Comp. Meth. Appl. Mech. Engrg.*, 92:353–375, 1991.

P. Alart, M. Barboteu, P. Le Tallec, and M. Vidrascu. Méthode de Scharwz additive avec solveur grossier pour problèmes non symétriques. *C. R. Acad. Sci. Paris*, 331: 399–404, 2000a.

P. Alart, M. Barboteu, P. Le Tallec, and M. Vidrascu. An iterative Schwarz method for non symmetric problems. In N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, editors, *Thirteenth International Conference on Domain Decomposition Methods*. http://www.ddm.org, 2000b.

P. Alart, M. Barboteu, and J. Gril. A numerical modelling of non linear 2d-frictional multicontact problems: application to post-buckling in cellular media. *Computational Mechanics*, 34:298–309, 2004.

P. Alart, D. Dureisseix, and M. Renouf. Using nonsmooth analysis for numerical simulation of contact mechanics. In Alart, Maisonneuve, and Rockafellar, editors, *Nonsmooth Mechanics and Analysis: Theoretical and Numerical Advances*, chapter 17, pages 195–208. Springer, 2006.

P. Avery, G. Rebel, M. Lesoinne, and C. Farhat. A numerically scalable dual-primal substructuring method for the solution of contact problem - part i: the frictionless case. *Comp. Meth. Appl. Mech. Engrg.*, 193:2403–2426, 2004.

M. Barboteu. Construction du preconditionneur neumann-neumann de decomposition de domaine de niveau 2 pour des problemes elastodynamiques en grandes deformations. *C.R. Acad. Sci. Paris*, I 340:171–176, 2005.

M. Barboteu, P. Alart, and M. Vidrascu. A domain decomposition strategy for non-classical frictional multi-contact problems. *Comp. Meth. Appl. Mech. Engrg*, 190: 4785–4803, 2001.

D. Bonamy, F. Daviaud, and L. Laurent. Experimental study of granular surface flows via a fast camera : a continuous description. *Phys. Fluids*, 14(5):1666–1673, 2002.

P. Breitkopf and M. Jean. Modélisation parallèle des matériaux granulaires. In *4ème colloque national en calcul des structures-Giens*, 1999.

B. Brogliato, AA. ten Dam, L. Paoli, F. Génot, and M. Abadie. Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *Appl. Mech. Rev.*, 55 (2):107–149, 2002.

B. Cambou and M. Jean. *Micromécanique des matériaux granulaires*. Hermès Science, 2001.

L. Champaney, J.Y. Cognard, D. Dureisseix, and P. Ladeveze. Large scale applications on parallel computers of a mixed domain decomposition method. *Computational Mechanics*, 19(4):253–262, 1997.

L. Champaney, J.-Y. Cognard, D. Dureisseix, and P. Ladevèze. Modular analysis of assemblages of 3D structures with unilateral contact conditions. *Comp. Struct.*, 73 (1-5):249–266, 1999.

P. A. Cundall. A computer model for simulating progressive large scale movements of blocky rock systems. In *Proceedings of the symposium of the international society of rock mechanics*, volume 1, pages 132–150, 1971.

J. Danek, I. Hlavacek, and J. Nedoma. Domain decomposition for generalized unilateral semi-coercive contact problem with given friction in elasticity. *Math. and Comp. in Simulation*, 68:271–300, 2005.

Y.-H. De Roeck and M. Le Tallec, P. an Vidrascu. A domain-decomposed solver for non linear elasticity. *Comp. Meth. Appl. Mech. Engrg.*, 99:187–207, 1992.

G. Dilintas, P. Laurent-Gengoux, and D. Trystram. A conjugate projected gradient method with preconditioning for unilateral contact problems. *Comp. Struct.*, 29(4): 675–680, 1988.

Z. Dostal. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim.*, 7(3):871–887, 1997.

Z. Dostal, A. Friedlander, and S. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemp. Math.*, 218:82–93, 1998. AMS.

Z. Dostal, F.A.M. Gomes Neto, and S.A. Santos. Solution of contact problems by FETI domain decomposition with natural coarse space projection. *Comput. Meth. Appl. Mech. Engrg.*, 190(13-14):1611–1627, 2000.

T.-J. Drake and O.-R. Walton. Comparison of experimental and simulated grain flows. *J. Appl. Mech.*, 62:131–135, 1995.

F. Dubois and M. Jean. LMGC90 une plateforme de développement dédiée à la modélisation des problèmes d'interaction. In *Actes du sixième colloque national en calcul des structures*, volume 1, pages 111–118. CSMA-AFM-LMS, 2003.

D. Dureisseix and C. Farhat. A numerically scalable domain decomposition method for the solution of frictionless contact problems. *Int. J. Num. Meth. Engrg*, 50:2643–2666, 2001.

C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32:1205–1227, 1991.

F. Feyel and J.L. Chaboche. FE^2 multiscale approach for modeling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Comput. Meth. Appl. Mech. Engrg.*, 183:309–330, 2000.

C. Glocker and F. Pfeiffer. *Multibody dynamics with unilateral contacts*. John Wiley and Sons, 1996.

E. Gondet and P.-F. Lavallee. *Cours OpenMP*. IDRIS, septembre 2000.

M. Jean. The non-smooth contact dynamics method. *Comp. Meth. Appl. Mech. Engrg*, 177:235–257, 1999.

F. Jourdan, P. Alart, and M. Jean. A Gauss-Seidel like algorithm to solve frictional contact problem. *Comp. Meth. Appl. Mech. Engrg*, 155:31–47, 1998.

N. Kikuchi and J.-T. Oden. *Contact problems in elasticity; a study of variational inequalities and Finite Element Methods*. SIAM,Philadelphia, 1982.

A. Klarbring. A mathematical programming approach to three-dimensional contact problems with friction. *Comput. Methods Appl. Mech. Engrg.*, 58:175–200, 1986.

P. Ladevèze. *Nonlinear Computational Structural Mechanics — New Approaches and Non-Incremental Methods of Calculation*. Springer Verlag, 1999.

P. Ladevèze and D. Dureisseix. A micro / macro approach for parallel computing of heterogeneous structures. *International Journal for Computational Civil and Structural Engineering*, 1:18–28, 2000.

P. Ladevèze, O. Loiseau, and D. Dureisseix. A micro-macro and parallel computational strategy for highly heterogeneous structures. *International Journal for Numerical Methods in Engineering*, 52(1–2):121–138, 2001.

P. Ladevèze, A. Nouy, and O. Loiseau. A multiscale computational approach for contact problems. *Comput. Meth. Appl. Mech. Engrg.*, 43:4869–4891, 2002.

P. Le Tallec. Domain decomposition methods in computational mechanics. *Comput. Mech. Adv.*, 1:121–220, 1994.

F. Lene and C. Rey. Some strategies to compute elastomeric lamified composite structures. *Composite Structure*, 54:231–241, 2001.

S. Li, D. Quian, D. Zhao, and T. Belytschko. A mesh free detection algorithm. *Comp. Meth. Appl. Mech. Engrg.*, 31:3271–3292, 2001.

X.L. Liu and J.V. Lemos. Procedure for contact detection in discrete element analysis. *Adv. Eng. Software*, 32:402–415, 2001.

J. Mandel. Balancing domain decomposition. *Communications Appl. Num. Meth.*, 9: 233–241, 1993.

H.-O. May. The conjugate gradient method for unilateral problems. *Comp. Struct.*, 12 (4):595–598, 1986.

J.-J. Moreau. On unilateral constraints, friction and plasticity. In G. Capriz and eds. G. Stampacchia, editors, *New Variational Techniques in Mathematical Physics (C.I.M.E. II ciclo 1973, Edizioni Cremonese, Roma, 1974)*, pages 173–322, 1974.

J.-J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J.-J. Moreau and eds. P.-D. Panagiotopoulos, editors, *Non Smooth Mechanics and Applications, CISM Courses and Lectures*, volume 302 (Springer-Verlag, Wien, New York), pages 1–82, 1998.

J-J. Moreau. Some numerical methods in multibody dynamics: application to granular materials. *Eur. J. Mech. A Solids*, 13(4-suppl.):93–114, 1994.

J.J. Moreau. Numerical aspects of sweeping process. *Comp. Meth. Appl. Mech. Engrg*, 177:329–349, 1999.

R. Motro. *Tensegrity*. Hermes Science Publishing, London, 2003.

E. Nezami, Y. Hashash, D. Zhao, and F. Ghaboussi. A fast detection algorithm for 3d discrete element method. *Comp. and Geotechnics*, 31:575–587, 2004.

S. Nineb, P. Alart, and D. Dureisseix. Domain decomposition approach for nonsmooth discrete problems, example of a tensegrity structure. *Computers and Structures*, 85 (9):499–511, 2007.

Han K. Owen D.R.J., Feng Y.T. and Peric. Dynamic domain decomposition and load balancing in parallel simulation of finite/discrete elements. In *European Congress on Computational Methods in Applied sciences and Engineering*. ECCOMAS, 2000.

J. Quirant, M.N. Kazi-Aoual, and R. Motro. Designing tensegrity systems: the case of a double layer grid. *Engineering Structures*, 25(9):1121–1130, 2003.

J. Rajchenbach. Flow in powders: From discrete avalanches to continuous regime. *Phys. Rev. Lett.*, 65(18):2221–2224, 1990.

J. Rajchenbach. Granular flows. *Adv. Phys.*, 49(2):229–256, 2000.

M. Raous and S. Barbarin. Conjugate gradient for frictional contact. In Edt. A. Curnier, editor, *Proc. Contact Mechanics Int. Symp.*, pages 423–432. PPUR, 1992.

G. Rebel, K.C. Park, and C.A. Felippa. A contact-impact formulation based on localised Lagrange multipliers. Technical Report Report no. cu-cas-00-18, Center for Aerospace Structures, University of Colorado at Boulder, 2000.

M. Renouf and P. Alart. Conjugate gradient type algorithms for frictional multicontact problems: applications to granular materials. *Comput. Methods Appl. Mech. Engrg.*, 194:2019–2041, 2005.

M. Renouf and P. Alart. Gradient type algorithms for 2d/3d frictionless/frictional multicontact problems. In P. Neittaanmaki, MT. Rossi, K. Mijava, and O. Pironneau, editors, *ECCOMAS 2004*, 2004.

M. Renouf, Dubois F., and P. Alart. A parallel version of the Non Smooth Contact Dynamics algorithm applied to the simulation of granular media. *J. Comput. Appl. Math.*, 168:375–382, 2004.

M. Renouf, D. Bonamy, P. Alart, and F. Dubois. Numerical simulation of 2d steady granular flows in rotative drum: on surface flow rheolog. *Phys. Fluids*, 17(10):103303, 2005.

Y. Saad and M. Schultz. Gmres : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

J. Schoberl. Efficient contac solvers based on domain decomposition techniques. *Computers and Mathematics*, 42:1217–1228, 2001.

Y. Song, R. Turton, and F. Kayihan. Contact detection algorithm for dem simulations of tablet-shaped particles. *Powder Technology*, 161:32–40, 2006.

P. Wriggers. Finite element algorithms for contact problems. *Arch. Comput. Meth. Engrg.*, 2:1–49, 1995.