



HAL
open science

A parallel and multiscale strategy for the parametric study of transient dynamic problems with friction

Pierre-Alain Boucard, David Odièvre, Fabrice Gatuingt

► To cite this version:

Pierre-Alain Boucard, David Odièvre, Fabrice Gatuingt. A parallel and multiscale strategy for the parametric study of transient dynamic problems with friction. *International Journal for Numerical Methods in Engineering*, 2011, 88, pp.657-672. 10.1002/nme.3194 . hal-00994263

HAL Id: hal-00994263

<https://hal.science/hal-00994263>

Submitted on 21 May 2014

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.

A parallel and multiscale strategy for the parametric study of transient dynamic problems with friction

P.-A. Boucard^{1*}, D. Odièvre¹ and F. Gatuingt¹

*LMT-Cachan (ENS Cachan/CNRS/Université Paris 6/PRES UniverSud Paris)
61 av. du Président Wilson, F-94230 Cachan, France*

SUMMARY

The objective of this work is to develop an efficient strategy for the parametric study of dynamic problems involving contacts with friction. Our approach is based on the multiscale LATIN method with domain decomposition. This is a mixed method which deals with the forces and velocities at the interfaces between the different subdomains simultaneously. We propose to take advantage of the capability of the multiscale LATIN method, called the *multiparametric strategy*, to reuse the solution of a given problem in order to solve similar problems. This strategy has already been applied successfully to a variety of static problems; here, it is extended to dynamics. First, we present the multiscale strategy in dynamics. Then, we show how the multiparametric strategy can be extended to dynamics. We illustrate the capabilities of the method through an academic 3D example and the simulation of a bolted joint.

KEY WORDS: parametric study; multiscale computational method; transient dynamics; domain decomposition; contact; friction; parallel processing

1. INTRODUCTION

Modeling and simulation play an important role in engineering and design and give rise to multiple problems, particularly in dynamics when dealing with large assemblies with connections. These connections have a significant impact on the dimensioning process because they are subject to highly nonlinear local phenomena (contacts with friction), particularly in rapid transient dynamics problems, whose proper representation requires very fine meshes [1]. In the case of structural assemblies, one's knowledge of the friction coefficients is especially limited. Although the incorporation of a system's parametric uncertainties into a model represents quite a challenge for structural engineers, without such information the structural response could not be analyzed accurately.

It is much more difficult to explore this design space taking these parametric uncertainties into account than to solve the general parametric field problem because one must take into consideration nonlinear structural behavior (in our case, unilateral contact with friction). In order to do that, it is necessary to calculate the response of the structure for each set of values of the design parameters [2]. In this case, the design parameters are the friction coefficients and the prestresses. Upon completion of these calculations, the response parameters (maximum

stresses, reaction forces...) are obtained and associated with the values of the input parameters. Then, the design space can be defined and explored by sampling parameter values within defined limits. In this context, the choice of an efficient computational method is of vital importance. The perturbation method is one of the main approaches used to obtain such structural responses ([2, 3]; [4, 5] for the dynamic case). In recent years, many researchers have focused on the stochastic finite element method, which is capable of dealing with the parametric uncertainties mentioned above [6]. Some of these methods have been developed in dynamics, such as the Neumann method [7] and the spectral method [8]. The POD method [9] can also be used as it allows to build a basis for several sensitivity analysis [10].

The objective of the work presented here is to develop a strategy which is suitable for problems involving multiple resolutions, which we will refer to as *multiparametric problems*. Some multiparametric strategies have been proposed for optimization problems [11, 12]. These lead to an efficient implementation of the multiresolution optimization technique. In this paper, we consider the case of assemblies of elastic structures connected through frictional contacts and dynamic loading conditions. Contact problems are characterized by constraints such as nonpenetration conditions, and an *a priori* unknown active area of contact (the area where contact indeed occurs). These conditions lead to stiff nonlinear systems of equations. There are several possible approaches for the resolution of static contact problems [13, 14, 15]. In most of these approaches, the numerical methods used to enforce the contact constraints are either Lagrange multiplier methods or penalty methods [16]. Penalty methods [17, 18] are closely related to the regularization of the contact constraints and are usually formulated in terms of the displacement variables; therefore, they are primal methods. They enable the treatment of contact as a material behavior. Penalty methods can be subject to various numerical difficulties, especially ill-conditioning, when a too large or too small penalty parameter is introduced. Lagrange multiplier methods are dual methods in which the multipliers, which represent the contact reaction forces, are introduced in order to enforce the nonpenetration conditions exactly. Augmented Lagrange multiplier methods [20, 21, 22, 23] lead to mixed formulations involving both displacement and force unknowns. The numerical resolution schemes underlying both Lagrange multiplier methods and augmented Lagrange multiplier methods are often related to the Uzawa algorithm [24, 25, 26].

In this paper, we develop a specific method in order to deal with the problems arising from the fact that the nonlinearities, as described previously, are localized in the connections. The friction law we use is Coulomb's classical law. Our approach is based on a decomposition of the assembly into substructures and interfaces. The interfaces play the vital role of providing an easy and accurate way to model local nonlinearities, such as contact and friction. The problem is solved in each substructure using the finite element method associated with a time integration scheme. An iterative scheme based on the multiscale LArge Time INcrement (LATIN) method developed at LMT Cachan [27, 28, 29] is used for the global resolution. This approach leads to a very significant reduction in computation cost for dynamic problems [30]. The single-scale LATIN method has already been used in dynamics in [40, 41, 42]. The multiscale LATIN method is a mixed method which deals with both velocities and forces at the interfaces simultaneously and solves a condensed macroscopic problem in order to accelerate the convergence of the numerical scheme. When only static cases are considered, the LATIN method (without the multiscale approach) can be derived by other means, such as in Lions [31] or Glowinski and Le Tallec [32].

More specifically, our objective is to calculate a large number of design configurations, each corresponding to a set of values of all the parameters involved in the mechanical analysis. Normally, a full calculation is needed for each set of parameters. Here, as an alternative, we propose to take advantage of the capabilities of the LATIN method and reuse the solution of one problem (for one set of parameters) to solve similar problems (for the other sets of parameters). This approach has already been used for viscoplastic and large-strain problems in [33, 34, 35] and is now also being used for structural identification problems [36]. It leads to a significant reduction in the total computation cost of determining the design space. The objective of the work presented here is the extension of this method, which has already been largely developed in statics and quasi-statics, to dynamics.

2. THE MULTISCALE LATIN METHOD

This multiscale domain decomposition method consists of three components: the decomposition of the spatial domain, the separation of the scales and a resolution algorithm. The main features of these three components are described below. The details of the method itself can be found in [37].

2.1. Decomposition into substructures and interfaces

An assembly is a set of substructures which communicate with one another through interfaces (see Figures 1(a) and 1(b)). Each interface represents a connection. The substructures and interfaces each have their own variables and equations (admissibility, equilibrium and behavior). The interface associated with two connected substructures Ω_E and $\Omega_{E'}$ is designated by $\Gamma_{EE'}$. Each interface is a mechanical entity with its own variables and behavior, which depend on the type of connection. Many different types of connections, such as frictional contact, can be modeled with this approach. The interface variables consist of two force fields F_E , $F_{E'}$ and two dual velocity fields W_E , $W_{E'}$ (see Figure 1(b)). By convention, F_E and $F_{E'}$ represent the action of the interface on the substructures, and W_E and $W_{E'}$ are the velocities of the substructures viewed from the interface. Thus, the interface concept can be easily extended to the boundary, where the displacements, the velocities or the forces are prescribed. In our multiscale vision, the interfaces play the important role of separating the scales: the definitions of the microscopic and macroscopic fields are related to the interface quantities of the substructured problem and are specified prior to any discretization. Then the interface unknowns are divided into :

$$W_E = W_E^m + W_E^M \quad \text{and} \quad F_E = F_E^m + F_E^M$$

where W^M and W^m denote respectively the macro parts and the micro complements of the velocity field.

A comparison with the coarse problem introduced into the FETI method for the dynamic case by Farhat-Shen-Mandel [45] can be found in [30].

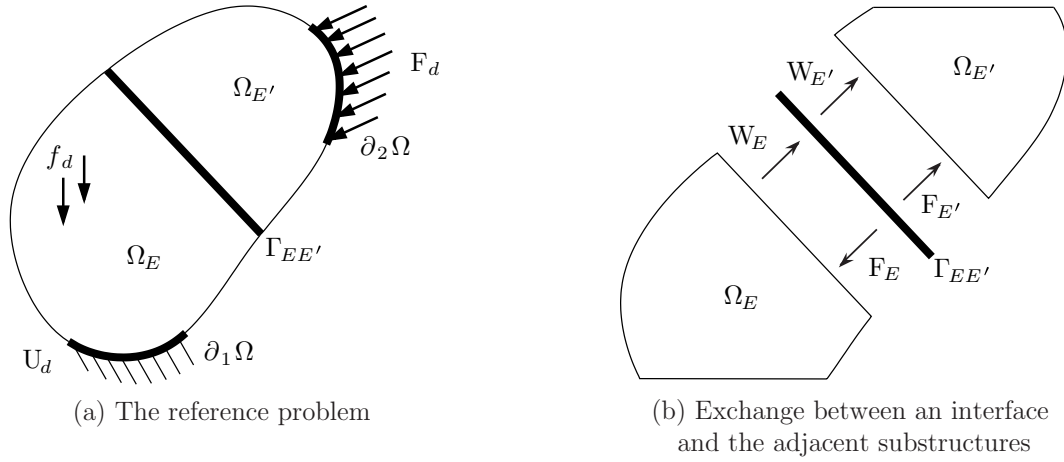


Figure 1. Decomposition of the reference problem into substructures and interfaces

2.2. The substructured problem

2.2.1. The problem within a substructure: Let $u_E(M, t)$ be the displacement field at any point M of substructure Ω_E and at any time t of $[0, T]$, and let $\mathcal{U}^{[0, T]}$ be the associated space. $\mathcal{U}_0^{[0, T]}$ is the virtual space defined by $\mathcal{U}_0^{[0, T]} = \{u \mid u \in \mathcal{U}^{[0, T]}, u = 0 \text{ on } \partial_1 \Omega\}$. $\varepsilon_E(M, t)$ is the strain field. The current state of the substructure is characterized by the stress field $\sigma_E(M, t)$, whose associated space is $\mathcal{S}^{[0, T]}$. The mechanical problem to be solved within each substructure Ω_E is:

Find the evolutions of the displacement field $u_E(M, t)$ and stress field $\sigma_E(M, t)$ such that:

- Kinematic admissibility: $\forall t \in [0, T], u_E \in \mathcal{U}^{[0, T]}$

– Initial conditions: $\forall M \in \Omega_E$

$$u_E(t=0) = U_E^0 \quad \frac{du_E}{dt}(t=0) = V_E^0 \quad (1)$$

– Boundary conditions: $\forall t \in [0, T], \forall M \in \Gamma_{EE'}$

$$\left. \frac{du_E}{dt} \right|_{\Gamma_{EE'}} = W_E \quad u_E|_{\partial\Omega_1} = U_d \quad F_E|_{\partial\Omega_2} = F_d \quad (2)$$

- Equilibrium: $\forall t \in [0, T], \forall \dot{u}^* \in \mathcal{U}_0^{[0, T]}, \sigma_E \in \mathcal{S}^{[0, T]}$

$$\int_{\Omega_E} \left(\rho \frac{d^2 u_E}{dt^2} + f_d \right) \dot{u}^* d\Omega + \int_{\Omega_E} \text{Tr}(\sigma_E \varepsilon(\dot{u}^*)) d\Omega = \sum_{E'} \int_{\Gamma_{EE'}} F_{EE'} \dot{u}^* d\Gamma \quad (3)$$

- Elastic behavior: $\forall t \in [0, T], \forall M \in \Omega_E$

$$\sigma_E = \mathbf{K}_E \varepsilon(u_E) \quad (4)$$

where \mathbf{K}_E is the Hooke's operator.

2.2.2. The problem at the interfaces: The mechanical problem to be solved at each interface $\Gamma_{EE'}$ is:

Find the evolutions of the force fields $F_E(M, t)$, $F_{E'}(M, t)$ and velocity fields $W_E(M, t)$, $W_{E'}(M, t)$ such that:

$$\bullet \forall t \in [O, T], \forall M \in \Gamma_{EE'} \quad (F_E, F_{E'}) = A_{\Gamma_{EE'}}(W_E, W_{E'}) \quad (5)$$

where the behavior is expressed as an evolution law $A_{\Gamma_{EE'}}$. This evolution law describes the behavior (perfect, contact, friction...) of the interface $\Gamma_{EE'}$. More details can be found in [30].

2.3. Resolution strategy: the LATIN method

The LATIN (LArge Time INcrement) method [27] is a general, mechanics-based computational strategy for the resolution of time-dependent nonlinear problems which operates over the entire time-space domain. It has been applied successfully to a variety of problems [29, 39, 40, 41].

In the case of linear elastic substructures considered here, the solution $u_E(M, t)$, $\sigma_E(M, t)$ can be calculated from the boundary values $W_E(M, t)$, $F_E(M, t)$. Thus, a solution s is characterized by the force and velocity fields on both sides of an interface. The solution of problem s_{ref} is expressed as a set of time-dependent fields within each substructure and at the corresponding interfaces:

$$s_{ref} = \sum_E s_E \quad s_E = \{F_E(M, t), W_E(M, t)\}$$

2.3.1. Separation of the difficulties: The LATIN approach is based on the idea of dealing with each difficulty separately in order not to have to solve a global problem and a nonlinear problem at the same time. The equations are divided into global linear equations and local nonlinear equations, so that $s_{ref} = A_d \cap \Gamma$ is the intersection of two subspaces:

- A_d , the space of the solutions of the linear equations associated with the substructures Ω_E : kinematic admissibility, equilibrium, elastic behavior and admissibility of the macroquantities;
- Γ , the space of the solutions of the local equations related to the interfaces $\Gamma_{EE'}$ and expressing their behavior.

2.3.2. A two-step iterative strategy: The LATIN method consists in seeking fields of Γ and A_d alternatively along two search directions E^+ and E^- , as shown in Figure 2. Each iteration involves two stages, called the local stage and the linear stage:

Local stage: Given $s_n = \{F_E, W_E\} \in A_d$, find $\widehat{s}_{n+1/2} = \{\widehat{F}_E, \widehat{W}_E\}$ such that:

$$\begin{aligned} \widehat{s}_{n+1/2} &\in \Gamma && \text{(interfaces)} \\ \widehat{s}_{n+1/2} - s_n &\in E^+ && \text{(search directions)} \end{aligned} \quad (6)$$

Linear stage: Given $\widehat{s}_{n+1/2} = \{\widehat{F}_E, \widehat{W}_E\} \in \Gamma$, find $s_{n+1} = \{F_E, W_E\}$ such that:

$$\begin{aligned} s_{n+1} &\in A_d && \text{(substructures)} \\ s_{n+1} - \widehat{s}_{n+1/2} &\in E^- && \text{(search directions)} \end{aligned} \quad (7)$$

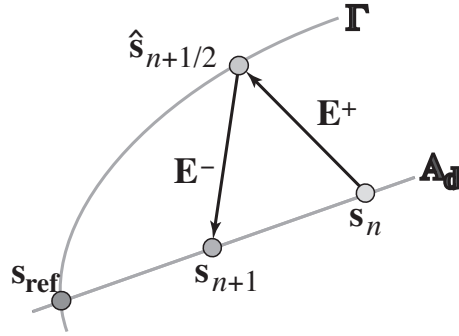


Figure 2. An iteration of the LATIN method

In the case of linear elastic substructures, which we are considering here, the search directions are defined as follows:

$$\widehat{s}_{n+1/2} - s_n \in E^+ \iff \widehat{F}_E - F_E = \mathbf{k}_0(\widehat{W}_E - W_E) \quad (8)$$

$$s_{n+1} - \widehat{s}_{n+1/2} \in E^- \iff F_E - \widehat{F}_E = -\mathbf{k}_0(W_E - \widehat{W}_E) \quad (9)$$

where \mathbf{k}_0 is a scalar parameter of the method. For the dynamic cases being addressed here, the optimum value of \mathbf{k}_0 for a one-dimensional problem was given in [41] and is $\mathbf{k}_0 = \sqrt{\rho E}$, where E is the Young's modulus and ρ the density. \mathbf{k}_0 can be viewed as a local impedance of the material.

The convergence of the algorithm toward s_{ref} is controlled using an error indicator η , which is a measure of the distance between the two solutions s_{n+1} and $\widehat{s}_{n+1/2}$ (Figure 2).

2.4. The local stage: $\widehat{s}_{n+1/2}$

The local stage consists in building $\widehat{s}_{n+1/2} \in \Gamma$ knowing $s_n \in Ad$. Then, $(\widehat{s}_{n+1/2} - s_n)$ must follow the search direction E^+ defined in Equation 8. We obtain a differential equations problem-type in each point of the interface that is function of its behavior introduced in Equation 5 [30].

2.5. The linear stage: s_{n+1}

The linear stage consists in building $s_{n+1} \in Ad$ knowing $\widehat{s}_{n+1/2} \in \Gamma$:

- Macroadmissibility: in order to ensure the admissibility conditions of the macro variables F_E^M and W_E^M .
- Search direction: the unknowns (W_E, F_E) must follow the search direction (Equation 9).

- Equations associated with the substructures: the unknowns (W_E, F_E) must verify the dynamic equilibrium and the elastic behavior (Equations 3 and 4).

These equations lead to the resolution of three independent problems. First one have to solved a “micro” problem, in each substructure, then a “macro” problem and at the end a second “micro” problem (see Table I and [30]).

2.6. The algorithm and its parallelization

The LATIN method associated with the mixed domain decomposition method is inherently parallelizable [38]. In our case, this strategy was programmed in C++ in the framework of the finite element platform developed by H. Leclerc [43]. Libraries such as MPI (Message Passing Interface) for the transfer of information among machines were used in order to enable the use of PC-cluster types of architectures. In order to parallelize the strategy, the first step consists in allocating the substructures and interfaces among the different processors. This is done through the METIS libraries [44], which enable the amount of data which must circulate among the processors to be minimized in order to avoid an excessive decline in speedup.

Initialize														
<table border="1"> <tr> <td>Loop over the substructures (on each processor)</td> </tr> <tr> <td> Calculation of rigidity and mass matrices of substructure</td> </tr> <tr> <td> Factorization</td> </tr> <tr> <td>Calculation and factorization of the macroproblem (on a single processor)</td> </tr> <tr> <td>Loop over the interfaces (on each processor)</td> </tr> <tr> <td> Loop over time ($\forall t \in [0, T]$)</td> </tr> <tr> <td> $\widehat{W}_E = \widehat{W}_{E'} = 0$ $\widehat{F}_E = \widehat{F}_{E'} = 0$</td> </tr> </table>	Loop over the substructures (on each processor)	Calculation of rigidity and mass matrices of substructure	Factorization	Calculation and factorization of the macroproblem (on a single processor)	Loop over the interfaces (on each processor)	Loop over time ($\forall t \in [0, T]$)	$\widehat{W}_E = \widehat{W}_{E'} = 0$ $\widehat{F}_E = \widehat{F}_{E'} = 0$							
Loop over the substructures (on each processor)														
Calculation of rigidity and mass matrices of substructure														
Factorization														
Calculation and factorization of the macroproblem (on a single processor)														
Loop over the interfaces (on each processor)														
Loop over time ($\forall t \in [0, T]$)														
$\widehat{W}_E = \widehat{W}_{E'} = 0$ $\widehat{F}_E = \widehat{F}_{E'} = 0$														
Iteration $n = 1, 2, \dots$ until convergence														
<table border="1"> <tr> <td>Linear stage - Loop over time ($\forall t \in [0, T]$)</td> </tr> <tr> <td> Loop over the substructures (on each processor)</td> </tr> <tr> <td> First microproblem:</td> </tr> <tr> <td> Solve the first micro problem at time step t</td> </tr> <tr> <td> Macroproblem (for a single processor):</td> </tr> <tr> <td> Determination of the macro part of the solution a time step t</td> </tr> <tr> <td> Loop over the substructures (on each processor)</td> </tr> <tr> <td> Second microproblem:</td> </tr> <tr> <td> Solve the second micro problem at time step t</td> </tr> <tr> <td> (operators are the same as those used in the first microproblem)</td> </tr> <tr> <td>Local stage - Loop over time ($\forall t \in [0, T]$)</td> </tr> <tr> <td> Loop over the interfaces (on each processor)</td> </tr> <tr> <td> Determination of $(\widehat{W}_E, \widehat{F}_E)$ given (W_E, F_E)</td> </tr> <tr> <td>Convergence test</td> </tr> </table>	Linear stage - Loop over time ($\forall t \in [0, T]$)	Loop over the substructures (on each processor)	First microproblem:	Solve the first micro problem at time step t	Macroproblem (for a single processor):	Determination of the macro part of the solution a time step t	Loop over the substructures (on each processor)	Second microproblem:	Solve the second micro problem at time step t	(operators are the same as those used in the first microproblem)	Local stage - Loop over time ($\forall t \in [0, T]$)	Loop over the interfaces (on each processor)	Determination of $(\widehat{W}_E, \widehat{F}_E)$ given (W_E, F_E)	Convergence test
Linear stage - Loop over time ($\forall t \in [0, T]$)														
Loop over the substructures (on each processor)														
First microproblem:														
Solve the first micro problem at time step t														
Macroproblem (for a single processor):														
Determination of the macro part of the solution a time step t														
Loop over the substructures (on each processor)														
Second microproblem:														
Solve the second micro problem at time step t														
(operators are the same as those used in the first microproblem)														
Local stage - Loop over time ($\forall t \in [0, T]$)														
Loop over the interfaces (on each processor)														
Determination of $(\widehat{W}_E, \widehat{F}_E)$ given (W_E, F_E)														
Convergence test														

Table I. Algorithm - The micro/macro LATIN method

Then, the different operators which are specific to the substructures are constructed on each processor. During the iterative resolution phase, the first and second microproblems are solved. The macroproblem, which has not been parallelized yet, is solved on a single processor. Finally, the local stage is completely parallelized because the interfaces are distributed among the different processors.

The LATIN method consists in performing linear stages and local stages alternatively. The iterations concern the whole time interval, *i.e.* a solution over the whole time interval is calculated at each iteration of the method. In the linear stage, an incremental formulation is used to solve problems within the substructures over the whole time interval. Thus, only two time steps of a substructure's fields (displacement, velocity and acceleration) need to be stored: the current time step being calculated and the previous time step. Conversely, the interface quantities $(\widehat{F}, \widehat{W}, F, W)$ must be stored over the whole time interval. The fields are saved after the last iteration of the method for post-processing purposes. Algorithm (Table I) shows the key steps of an iteration of the multiscale method.

3. THE STRATEGY FOR THE PARAMETRIC STUDY

The strategy proposed consists in calculating design configuration where each design configuration corresponds to a set of values of all the variable parameters (friction coefficients, prestresses) which are introduced into the mechanical analysis.

At each iteration, the LATIN method leads to an approximate solution to the problem over the whole time interval. Therefore, the trick is to reuse one approximation (associated to one set of values of all the design parameters) to find the solution to another design configuration (another set of the design parameters) similar to the one for which it was calculated in the first place. Our strategy for the parametric study uses the fact that the LATIN algorithm can be initialized with any solution (usually an elastic solution) provided that it verifies the admissibility conditions. Therefore, the key to our technique is to initialize the process associated with a new design configuration using the results of the calculation carried out on the preceding set of values of the design parameter (Figure 3). In this scheme, Γ_i correspond to one set of parameters and Γ_{i+1} to another one. S_i (equal to S_{ref} for Γ_i) is the solution of the first set of parameters and correspond to the initialization of the LATIN algorithm for the new ones. In this manner, a first approximation of the solution to the new design with a strong mechanical content is immediately available from the start. Note that as the previous computation is expected to have converged, the solution of the local stage should be identical to the one provided by the linear stage. One can use any of them to initialize the algorithm.

In this particular case of elastic structures in contact, the interfaces play a vital role: they enable one to initiate the calculation on the new design configuration without having to save all data on the substructures as well as to search for the solution of the new design configuration with an initial solution well-suited to the target problem. In the best-case scenario, only a few iterations are necessary: the solution to the problem is obtained at low cost. If the solutions to the design configurations are close enough, the latter can still be derived at a significantly lower cost than by using a full calculation. For the parametric study presented herein, we just change the parameters between iterations. Thus, the new computation is initialized by the solution to the previous one. If the parameters change slowly, the two solutions are close and only a few iterations are needed to reach convergence in the new calculation.

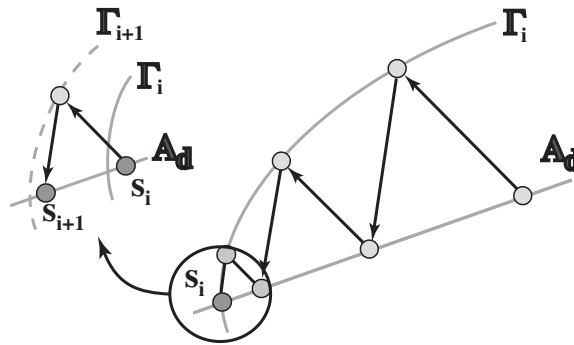


Figure 3. Schematic view of the reuse of a previous LATIN solution

Initialize

Loop over the substructures (on each processor) <ul style="list-style-type: none"> Calculation of matrices K_E, M_E, \dots Factorization Calculation and factorization of the macroproblem (on a single processor) <ul style="list-style-type: none"> Loop over the interfaces (on each processor) <ul style="list-style-type: none"> Loop over time ($\forall t \in [0, T]$) <ul style="list-style-type: none"> $\widehat{W}_E = \widehat{W}_{E'} = 0 \quad \widehat{F}_E = \widehat{F}_{E'} = 0$ Definition of the sets of values of all the design parameters
--

Loop over the sets of values of all the design parameters ($k = 1, 2, \dots$)

Restoration of the quantities at the interfaces <ul style="list-style-type: none"> Iteration until convergence <ul style="list-style-type: none"> Linear stage (Table I) Local stage (Table I) Convergence test Save the interface solution for the k^{th} parameter set

Table II. Algorithm of the parametric study using the LATIN method

4. FIRST EXAMPLE: AN ACADEMIC PROBLEM

In order to test the method described previously, let us consider a simple 3D example consisting in the propagation of a compression wave in a bar composed of three parallelepipeds (Figure 4). Each parallelepiped is 30 mm-long and 15 mm-wide. The material properties are: Young's modulus 150 GPa, mass density 7,800 kg/m³ and Poisson's ratio 0.45 for the central parallelepiped, and Young's modulus 200 GPa, mass density 7,800 kg/m³ and Poisson's ratio 0.3 for the other two. Contact with friction is assumed to take place between these parallelepipeds with friction coefficients f_1 for Interface 1 and f_2 for Interface 2 (Figure 4).

First, the bar was subjected to a static loading in the form of a prescribed displacement applied to the upper parallelepiped ($U = -0.1\text{mm}$ along the \bar{z} -axis). This loading consisted of a normal pressure at each contact interface, which corresponds to a static prestress in the

structure. Then, a pressure was applied laterally to the central parallelepiped going from zero

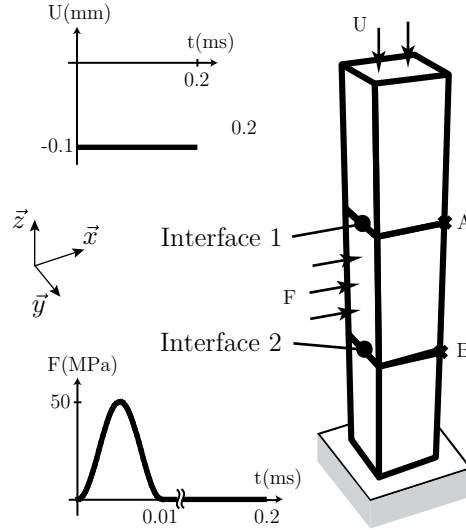


Figure 4. The numerical example

to a maximum value of 50 MPa over 10 μ s. This pressure resulted in slipping of the middle parallelepiped along the \vec{x} axis by an amount which depends on the value of the two friction coefficients. For the purpose of this test, each parallelepiped was decomposed into several substructures. The duration of the test was 0.2 ms and we used 50 time steps of 4 μ s each.

The objective of this example was to study the influence of the friction coefficients on the slippage of the central parallelepiped. Each friction coefficient was given 11 different values from 0.1 to 0.2 in increments of 0.01. At the end of each simulation, the amounts of slippage at points A and B were saved (Figure 4). The parameters of the simulation (material properties, loading intensities, friction coefficients,...) were chosen such that the pressure on the central parallelepiped would induce slipping during the study and the parallelepipeds would stick at the end of the study. Figure 5 shows the displacement and velocity of point A (along \vec{x}) obtained for $f_1 = 0.1$ and $f_2 = 0.1$, with the fields of the upper parallelepiped in solid black lines and the fields of the middle parallelepiped in dotted black lines. The slippage of point A is equal to the difference between the two displacement curves.

Each parallelepiped was divided into 16 substructures consisting of $10 \times 10 \times 10$ 8-node linear elements. The simulation was carried out using 16 processors. This example of a parametric study required 121 resolutions of a nonlinear problem meshed with 144,000 DOFS over 50 time steps. We studied the slippage of points A and B of the two contact interfaces (Figure 4). Figure 6 shows the response surfaces of the parametric study, *i.e.* the evolutions of the slippage at points A and B as functions of the two friction coefficients. These two surfaces illustrate the influence of the friction coefficients. For example, the slippage of point A depends on the friction coefficient f_1 alone, and this dependence is highly nonlinear. Moreover, above $f = 0.2$,

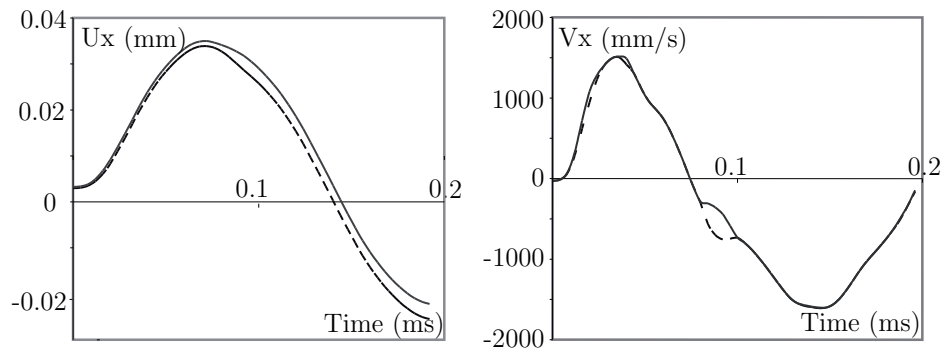


Figure 5. Evolution of the displacement and velocity of point A (along \vec{x}), upper parallelepiped in solid black lines and middle parallelepiped in dotted black lines

the slippage of the interfaces is nearly zero.

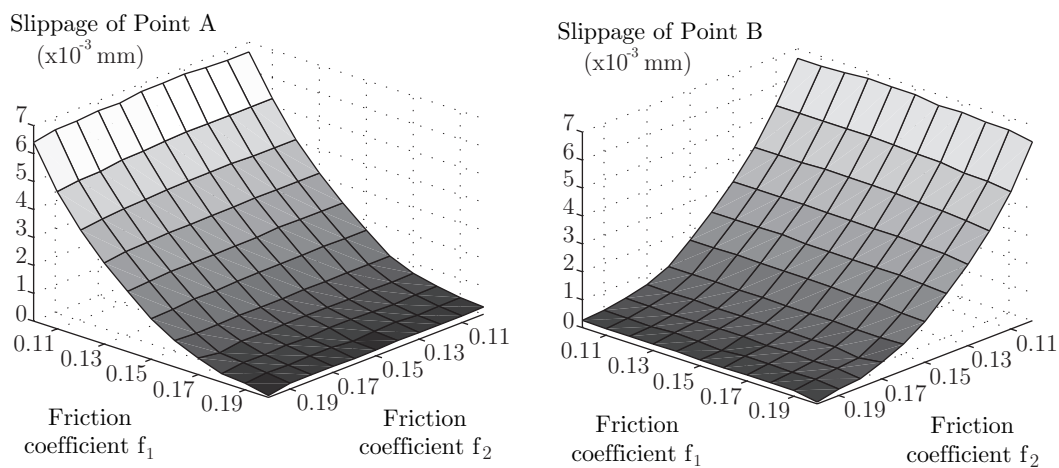


Figure 6. Response surface - slippage of points A and B

For each value of f_1 starting with $f_1 = 0.1$, we applied the parametric strategy to perform the 11 simulations associated with each value of f_2 starting with $f_2 = 0.1$, *i.e.* we applied the multiparametric strategy 11 times for each value of f_1 . In order to optimize also the first calculation of each multiparametric strategy ($f_2 = 0.1$), we saved the solution of the previous simulation to initialize the next multiparametric strategy (for the next value of f_1). For this example, the convergence criterion of the iterative algorithm was set to 2.10^{-4} . Figure 7 shows the evolution of the LATIN error indicator during the first multiparametric strategy ($f_1 = 0.1$). The number of iterations and the CPU time associated with each simulation (for each value of f_2) are shown in Table III. The error indicator increased every time the

friction coefficient changed, but to a lesser extent than in the first initialization thanks to the multiparametric strategy. Thus, one can observe that the number of iterations required to achieve convergence decreased throughout the parametric study. With the multiparametric strategy, both the number of iterations and the computation cost decreased because the results of the successive computations were very similar (as shown by Figure 6).

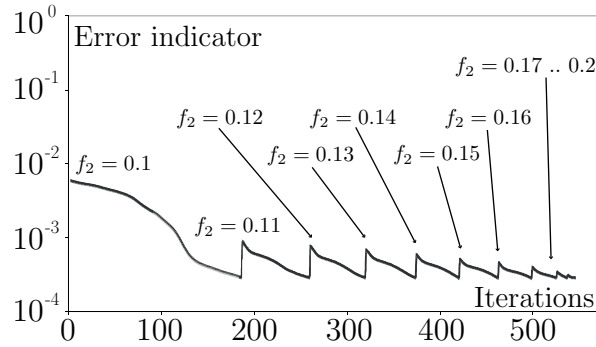


Figure 7. Evolution of the error indicator for $f_1 = 0.1$

f_2	0.1	0.11	0.12	0.13	0.14	0.15	0.16	0.17	0.18 .. 0.20
Iterations	194	76	62	56	48	43	37	27	< 10
Time (min)	26.4	10.3	8.4	7.6	6.5	5.8	5	3.7	< 1.5

Table III. Numerical costs of the first multiparametric strategy ($f_1 = 0.1$)

In Table III, the number of iterations decreases continuously. The gain is very important for the last values of f_2 (0.18...0.2). That means that the improvement does not only depend on the method, but so on the nature of the solutions. Indeed the solutions for f_2 equal to 0.18.. 0.2, only involve stick statuses over the interface, which is easier to solve than mixed solutions with stick and slip statuses. This particular point of the strategy is highlighted by testing a parametric study starting for f_2 from 0.2 to 0.1 by a decreasing way. Figure 8 shows the evolution of the LATIN error indicator during this new multiparametric strategy ($f_1 = 0.1$). The total number of iterations in this case is 753 (to compare with the 570 iteration of the first parametric study). In terms of CPU time, for this second study total time is 102.4 min (to compare with 71 min). We can also compare the gain: 2.9 for the second parametric study and 4.1 for the first one. This result shows that it is possible to improve the parametric strategy by seeking the best point for reinitializing the LATIN algorithm. Indeed, at a given time in the parametric study, one can have access to several already calculated solutions. Thus, in performing a new calculation, one can reasonably assume that among the previously calculated solutions one is likely to maximize the cost reduction for this new calculation. In this example the i^{th} calculation is reinitialized from the results of calculation $i - 1$, but one can easily imagine that it could be better if the i^{th} calculation is reinitialized from the results of the closest calculation, in the

sense of a distance in the parametric space [46].

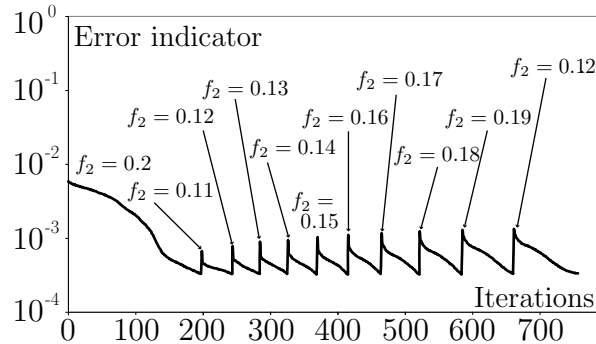


Figure 8. Evolution of the error indicator for $f_1 = 0.1$

f_2	0.2	0.19	0.18	0.17	0.16	0.15	0.14	0.13	0.12	0.11	0.10
Iterations	199	45	40	41	43	45	49	56	63	76	96
Time (min)	27	6.1	5.4	5.5	5.8	6.1	6.6	7.6	8.5	10.3	13

Table IV. Numerical costs for f_2 from 0.2 to 0.1 ($f_1 = 0.1$)

Table IV shows a comparison of the computation costs of the different strategies. The first CPU time (Direct LATIN, sequential) corresponds to 121 times the CPU time of the first calculation (194 iterations) carried out using a single processor, the second CPU time (Direct LATIN, parallel) corresponds to 121 times the CPU time of the first calculation carried out using 16 processors, and the third CPU time corresponds to the multiparametric strategy carried out using 16 processors. For this example, the gain in CPU time obtained thanks to parallelism using 16 processors was 11 (588/53.4). The gain in CPU time obtained thanks to the multiparametric strategy was 4.4 (comparison between the direct LATIN parallel and the Multiparametric LATIN parallel: 53.4/12.1). Parallelism and the multiparametric strategy together enabled us to carry out the multiparametric study in a reasonable time compared to the direct approach, with a global gain of about 48.

Calculation	Cost (hrs)	Cost (days)	Gain vs. Direct LATIN sequential
Direct LATIN (sequential, estimated)	588	24.5	
Direct LATIN (parallel, estimated)	53.4	2.2	11
Multiparametric LATIN (parallel)	12.1	0.5	48

Table V. Cost comparison of the different calculations

We compare the results with those obtained with the industrial explicit dynamic code LS-DYNA [47]. To obtain the same quality in terms of contact and sliding at the interface, we

had to use a finer mesh in LS-DYNA. For each parallelepiped we use 40x40x80 elements (the number of elements in each direction is doubled in comparison with the mesh used with our own software: 20x20x40 elements). Indeed, if we use the same mesh with LS-DYNA, the behavior of the interface is not verified exactly: the parallelepipeds are not in contact as they should be, they interpenetrate at the end of the calculation. We compare the results in terms of velocity of point A (figure 9). It is obvious that the solutions are similar and that a finer mesh is necessary for LS-DYNA to obtain the solution.

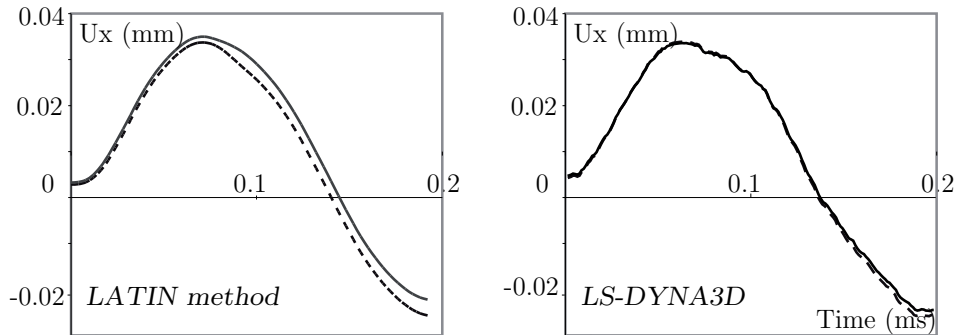


Figure 9. Evolution of the displacement of point A (along \bar{x}), upper parallelepiped in solid black lines and middle parallelepiped in dotted black lines

The total computation time with LS-DYNA3D on one computation and one processor ($f_1 = 0.1$ and $f_2 = 0.1$) is about 7h and the memory used about 850 MBytes. In comparison, the computation time with our own software is about 4h and 50 minutes and the memory used is about 1800 MBytes. Therefore, the computational times associated with a solution quality are comparable. Our research software is not optimized in terms of memory occupation. Nevertheless, one limitation of the proposed approach is a limited number of time steps. We compare the memory used for several time discretization : 50, 100, and 200 time steps. For this 3 values, the memory used is respectively : 1800, 2400 and 3600 MBytes. Contrary to a classical incremental scheme for which the volume of computations roughly doubles when doubling the time steps, here, not only this number of operations doubles, but also the storage of the iterates on which the computations are performed. Moreover, when we double the number of time step, the total time of the analysis nearly double also.

5. APPLICATION TO A BOLTED JOINT USED IN THE AEROSPACE INDUSTRY

Let us now consider the example of a bolted joint. Damping in such joints is sensitive to friction, pretension in the bolts, etc... These parameters are naturally scattered, and a full test campaign to evaluate their actual influence would involve a very large number of specimens. Such studies in quasi-statics have been reported in [48].

This example concerns the propagation of a compression wave in the bolted joint. This 3D assembly is a joint between two sandwich composite structures consisting of two metal parts

and three bolts (Figure 10). For the purpose of this example, the core and the composite material of the sandwich were assumed to be homogeneous, elastic and isotropic media. The interfaces between the sandwich composites and the metal parts were assumed to be perfect. Frictional contact was taken into account both at the interfaces between the bolts and the metal parts and at the interface between the two metal parts. First, we applied a static loading and prestresses in the bolts. A static calculation was performed in order to define the initial conditions of the dynamic analysis. For the dynamic calculation, the loading consisted of the prescribed velocity shown in Figure 10.

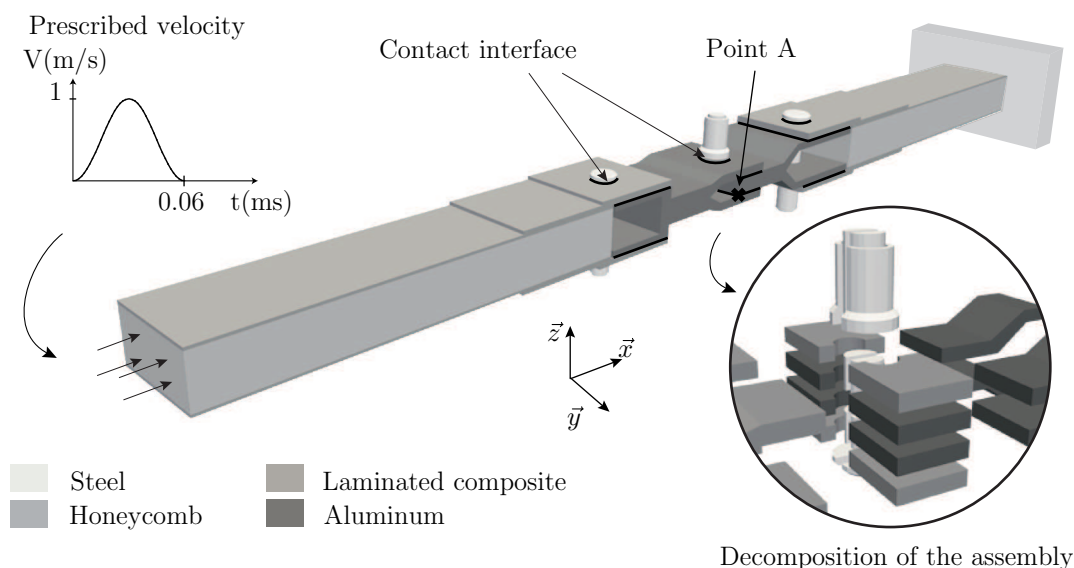


Figure 10. The model of the joint

The calculations were performed using a 450,000-DOF mesh for the whole assembly. In order to use our approach, we divided the different parts into several substructures. Figure 10 shows a portion of this decomposition. We chose to use substructures with similar numbers of DOFs in order to achieve a well-balanced load among the different processors and a good speedup of the parallel algorithm. We used 194 substructures with about 2,500 DOFs each. An implicit scheme with 39 time steps of $6 \mu\text{s}$ each was used for the time integration for a total duration of $240 \mu\text{s}$.

The objective of this example was to study the influence of the friction coefficients and of the prestress in the bolt on the slippage between the two metal parts. The friction coefficient was given 11 different values from 0.2 to 0.4 in increments of 0.02, and the prestress in the middle bolt was also given 11 different values from 0.02 mm and 0.04 mm in increments of 0.002 mm. The prestress in the bolt was represented as a relative displacement between the body and the head of the bolt. At the end of each simulation, we saved the amount of slippage of point A along \vec{x} (Figure 10). Figure 11 shows the displacement Ux and the velocity Vx of point A along \vec{x} obtained for a friction coefficient equal to 0.2 and a prestress equal to 0.02 mm, with

the fields of the right-hand side in solid black lines and the fields of the left-hand side in dotted black lines. The slippage of point A is equal to the difference between the two displacement curves.

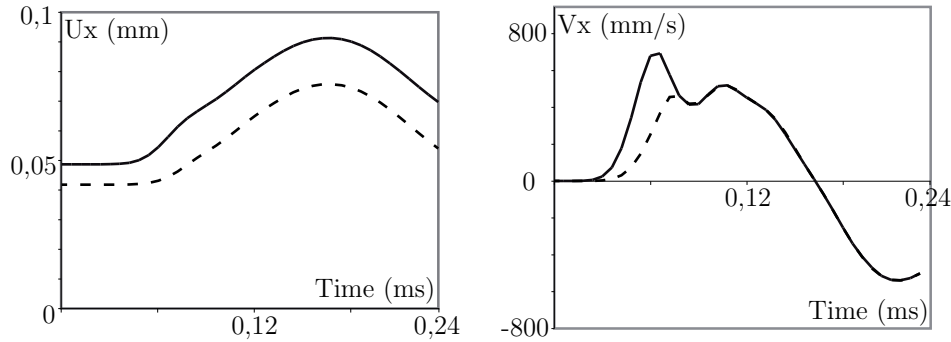


Figure 11. Evolution of the displacement and velocity of point A along \vec{x} , right-hand side in solid black lines and left-hand side in dotted black lines

This example of a parametric study involved 121 resolutions of a nonlinear problem with a 450,000-DOF mesh and 39 time steps. The simulation was carried out using 16 processors. Figure 12 shows the response surfaces of the parametric study: the evolution of the slippage of point A as a function of the friction coefficient and of the prestress at the beginning and at the end of the study. These two surfaces illustrate the influence of the parameters on the response of the structure. For example, the dependence of the slippage on these parameters at the beginning of the study is highly nonlinear, with the slippage being equal to zero for part of the variation range of the parameters.

For each value of the prestress starting with 0.02 mm, we applied the multiparametric strategy to calculate the 11 simulations associated with each value of the friction coefficient starting with 0.2, *i.e.* we applied the multiparametric strategy 11 times for each value of the prestress. In order to optimize also the first calculation of each multiparametric strategy ($f = 0.1$), we saved the solution of that first simulation to initialize the next multiparametric strategy (for the next value of the prestress). Figure 13 shows the number of iterations carried out for each parameter set. While the first calculation required more than 700 iterations, each of the subsequent calculations took less than 100 iterations thanks to the multiparametric strategy.

Table V shows a comparison of the computation costs of the different strategies. The first CPU time (Direct LATIN, sequential) corresponds to 121 times the CPU time of the first calculation (700 iterations) carried out using a single processor, the second CPU time (Direct LATIN, parallel) corresponds to 121 times the CPU time of the first calculation carried out using 16 processors and the third CPU time corresponds to the multiparametric strategy carried out using 16 processors. For this example, the gain in CPU time obtained thanks to parallelism using 16 processors was 9.5 (1354/141.1). The gain in CPU time obtained thanks to the multiparametric strategy was 5.8 (comparison between direct LATIN parallel and

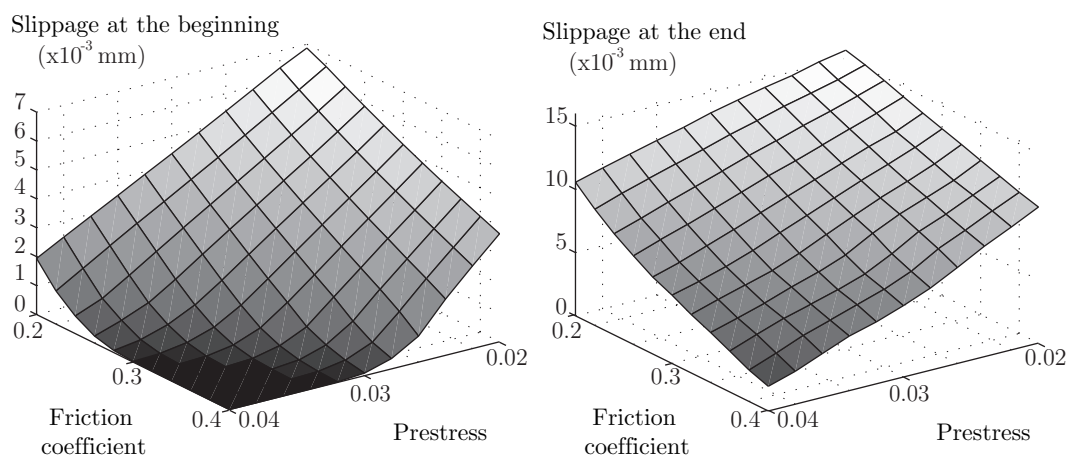


Figure 12. Response surface - slippage of point A at the beginning and at the end of the study

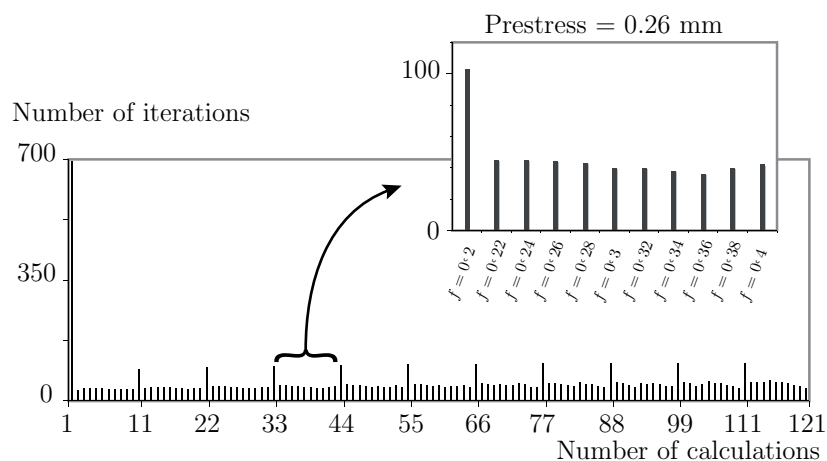


Figure 13. Number of iterations required for convergence in each calculation

multiparametric LATIN parallel: 141.1/24.4). Parallelism and the multiparametric strategy together enabled us to carry out the multiparametric study in a reasonable time compared to the direct approach, with a global gain of about 55.

6. CONCLUSION

We presented the extension of the multiparametric strategy based on the LATIN method to dynamic problems involving complex 3D structural assemblies. The LATIN method is based

Calculation	Cost (hrs)	Cost (days)	Gain vs. Direct LATIN sequential
Direct LATIN (sequential, estimated)	1354	56.4	
Direct LATIN (parallel, estimated)	141.1	5.9	9.5
Multiparametric LATIN (parallel)	24.4	1	55

Table VI. Cost comparison of the different calculations

on three components: a mixed decomposition of the structure (which provides significant modularity to the description of the problem), a multiscale description and an iterative resolution scheme. Using scale separation, a macroproblem is defined at the interface by condensing the microproblem, which accelerates the convergence of the iterative algorithm. This method is particularly well-suited to problems involving frictional contact. The strategy is also fully parallelized. Its implementation into a cluster architecture leads to good behavior of the parallel algorithm and enables one to treat complex real-life structural assemblies with very large numbers of degrees of freedom.

The multiparametric strategy proposed here is based on the LATIN method and, more specifically, on its capability to reuse the solution of a given problem in order to solve similar problems. Our first numerical example showed the very good behavior of the algorithm in the case of multiple resolutions in the analysis of dynamic problems with contact and friction. The second example concerned a real 3D joint. For this assembly, parametric studies were carried out on the values of the connection parameters (friction coefficient, prestress). The two examples presented show that the algorithm can be very efficient numerically. The parametric studies were carried out about 50 times faster than with the direct approach. Thanks to the multiparametric strategy and parallelism of the LATIN method, it is possible to carry out such parametric studies in reasonable time.

REFERENCES

1. Hallquist JO, Goudreau GL, Benson DJ. Sliding interfaces with contact-impact in large-scale lagrangien computations. *Computer Methods in Applied Mechanics and Engineering* 1985; 51:107–137.
2. Benaroya H, Rehak M. Finite element methods in probabilistic structural analysis: A selective review. *Mechanical Engineering* 1998; **41**(5):201–213.
3. Macias OF, Lemaire M. Stochastic Finite Elements and fiability. Application to fracture mechanics (in french). *Revue Franaise de gnie civil* 1997, 1(2).
4. Liu WK, Bestereld G, Belytschko T. Transient probabilist systems. *Computer Methods in Applied Mechanics and Engineering* 1988; **67**(1):27–54.
5. Qiu Z, Xiaojun Wang. Parameter perturbation method for dynamic responses of structures with uncertain-but-bounded parameters based on interval analysis. *International Journal of Solids and Structures* 2005;**42**:4958–4970.
6. Kleiber M, Hein TD. *Basic perturbation technique and computer implementation*. John Willey and Sons, 1992.
7. Zhao L, Qiu C. Neumann dynamic stochastic finite element method of vibration for structures with stochastic parameters to random excitation. *Computers and Structures* 2000;**77**:651–657.
8. Ghanem R, Spanos P. Stochastic finite element : a spectral approach. Ed. Springer-Verlag New York, 1991.
9. Chatterjee A, An introduction to the proper orthogonal decomposition. *Current Science* 2000; **78**(7):808817.
10. Ryckelynck D, Chinesta F, Cueto E, Ammar A. On the a priori model reduction : Overview and recent developments. *Archive of Computational Methods in Engineering* 2006;**13**(1):91–128.

11. Kim YY, Yoon GH. Multi-resolution multi-scale topology optimization: a new paradigm. *Int. J. Solids Structures* 2000; **37**(39):5529–5559.
12. Boucard PA, Buytet S, Guidault PA. A multiscale strategy for structural optimization. *Int Jnl for Numerical Methods in Engineering* 2009; **78**(1):101–126.
13. Kikuchi N, Oden JT (Eds) *Contact problems in elasticity: a study of variational inequalities and finite element methods*. SIAM, Philadelphia, 1988.
14. Zhong Z, Mackerle J. Static contact problems : a review. *Engineering Computations* 1992; **9**:3–37.
15. Wriggers P. Finite element algorithms for contact problems. *Archives of Computational Methods in Engineering* 1995; **2**:1–49.
16. Wriggers P. and al. Penalty and augmented Lagrangian formulations for contact problems. *Proc. NUMETA Conf.*, Swansea, 1985.
17. Kikuchi N. Penalty/finite element approximations of a class of unilateral contact problems. In *Penalty method and finite element method*. ASME, New York, 1982.
18. Armero F, Petocz E. Formulation and analysis of conserving algorithms for dynamic contact / impact problems. *Computer Methods in Applied Mechanics and Engineering* 1998; **158**:269–300.
19. Alart P, Curnier A. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Computer Methods in Applied Mechanics and Engineering* 1991; **92**:253–375.
20. Arora JS, Chahande AI, Paeng JK. Multiplier methods for engineering optimization. *International Journal for Numerical Methods in Engineering* 1991; **32**:1485–1525.
21. Klarbring A. Mathematical programming and augmented lagrangian methods for frictional contact problems. *Proceedings of the Contact Mechanics International Symposium, Curnier A. (ed.)*, Presses Polytechniques et Universitaires Romandes 1992; 409–422.
22. Radi B, Baba OA, Gelin JC. Treatment of the frictional contact via a Lagrangian formulation. *Mathematical and Computer Model ling, Rodin E.Y., Shillor M. (Eds.)*, Pergamon Press: Oxford 1998; **28**:407–412.
23. Chabrand P, Dubois F, Raous M. Various numerical methods for solving unilateral contact problems with friction. *Mathematical and Computer Model ling, Rodin E.Y., Shillor M. (Eds.)*, Pergamon Press: Oxford 1998; **28**:97–108.
24. Arrow KJ, Hurwicz L, Uzawa H. Studies in nonlinear programming. University Press, Stanford, CA, 1958.
25. Simo JC, Laursen TA. An augmented lagrangian treatment of contact problems involving friction, *Computers and Structures* 1992; **42**:97–116.
26. Champany L, Cognard JY, Ladevèze P. Modular analysis of assemblages of three-dimensional structures with unilateral contact conditions. *Computers and Structures* 1999; **73**:249–266.
27. Ladevèze P. *Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation*. Springer Verlag, 1999.
28. Ladevèze P, Dureissex D. A micro/macro approach for parallel computing of heterogeneous structures. *Int Jnl for Computational Civil and Structural Engineering* 2000; **1**:18–28.
29. Ladevèze P, Nouy A, Loiseau O. A multiscale computational approach for contact problems. *Computer Methods in Applied Mechanics and Engineering Software* 2002; **191**:4869–4891.
30. Odievre D, Boucard PA, Gautuingt F. A parallel, multiscale domain decomposition method for the transient dynamic analysis of assemblies with friction. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(21-22):1297–1306.
31. Lions P. On the schwarz alternating method iii. a variant for non-overlapping subdomains. In *Proceedings of Domain Decomposition Methods for Partial Differential Equations* 1990.
32. Glowinski R, Le Tallec P. Augmented lagrangian interpretation of the nonoverlapping schwarz alternating method. In *Internationnal Symposium on Domain Decomposition Methods*, Philadelphia, SIAM 1990; 224–231.
33. Boucard PA, Ladeveze P. A multiple solution method for non-linear structural mechanics. *Mechanical Engineering* 1999; **50**(5):317–328.
34. Boucard PA, Champany L. A suitable computational strategy for the parametric analysis of problems with multiple contact. *International Journal for Numerical Methods in Engineering* 2003; **00**:1–6.
35. Champany L, Boucard PA, Guinard S. Adaptive multi-analysis strategy for contact problems with friction. Application to aerospace bolted joints. *Computational Mechanics* 2008; **42**(2):305–316.
36. Allix O, Vidal P. A new multi-solution approach suitable for structural identification problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:2727–2758.
37. Ladevèze P, Loiseau O, Dureissex D. A micro-macro and parallel computational strategy for highly heterogeneous structures. *Int Jnl for Numerical Methods in Engineering* 2001;**52**:121–138.
38. Champany L, Cognard JY, Dureisseix D, Ladevèze P. Large scale applications on parallel computers of a mixed domain decomposition method. *Computational Mechanics* 1997;**19**:253–263.
39. Ladevèze P, Nouy A. On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:3061–3087.
40. Boucard PA, Ladevèze P, Lemoussu H. A modular approach to 3D impact computation with frictional

- contact, *Computer & Structures* 2000; **78**:45–52.
41. Lemoussu H, Boucard PA, Ladevèze P. A 3D shock computational strategy for real assembly and shock attenuator. *Adv Engrg Software* 2002; **33**:517–526.
 42. Sen Gupta J, Allix O, Boucard PA, Ladevèze P. Mesodynamics of 3D C/C : a dedicated numerical strategy. *Comuters & Structures* 2006; **84**(19/2):1177–1189.
 43. Lecerc H, Guidualt PA, Developments and experimentations on some iterative numerical strategies for solid mechanics on multicore and graphic processors. *USNCCM 10-10th US National Congress of Computational Mechanics*, 2009.
 44. Karypsis G, Kumar V. “METIS”, A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0, Department of Computer Science, University of Minnesota, 1998.
 45. Farhat C, Chen PS, Mandel J. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Int Jal for Numerical Methods in Engineering* 1995;**38**:3831–53.
 46. Boucard PA, Buytet S, Soulier B, Chandrashekarappa P, Duvigneau R. *Multidisciplinary Design Optimization in Computational Mechanics . Section Multilevel Modeling*, Publisher Hermes, 199-260, 2010.
 47. Hallquist JO, *LS-DYNA3D Theoretical Manual*, Livermore Software Technology Corp., Livermore, 1991.
 48. Caignot A, Ladevèze P, Néron D, Durand JF. Virtual testing for the prediction of damping in joints. *Engineering Computations* 2010; **27**(5):621–644.