HAL
open science

A nonlinear dual domain decomposition method : application to structural problems with damage

Julien Pebrel, Christian Rey, Pierre Gosselet

To cite this version:

Julien Pebrel, Christian Rey, Pierre Gosselet. A nonlinear dual domain decomposition method : application to structural problems with damage. International Journal for Multiscale Computational Engineering, 2008, 6 (3), pp.251-262. hal-00344445

HAL Id: hal-00344445

https://hal.science/hal-00344445v1

Submitted on 4 Dec 2008

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 nonlinear dual domain decomposition method: application to structural problems with damage

Julien Pebrel, Christian Rey, Pierre Gosselet

LMT-Cachan (ENS Cachan/CNRS UMR8535/UPMC/PRES UniverSud Paris)
61, avenue du Président Wilson
F-94235 CACHAN CEDEX, France
{pebrel,rey,gosselet}@lmt.ens-cachan.fr

Key Words : Nonlinear Domain Decomposition, Dual Approach.

Abstract

A dual domain decomposition method dedicated to nonlinear problems is presented. The decomposition is introduced in the nonlinear formulation and the nonlinear problem is first condensed on the interface then solved by a Newton-type method. Considering the specificities of the introduced operators, the algorithm can be interpreted as a local/global strategy with global Newton-type iterations and nonlinear relocalizations per subdomain. Such a strategy is particularly interesting in cases where the nonlinearity is localized. First results are presented on structural problems with damage.

1 Introduction

Nonlinear analysis is a keypoint in the simulation of structures. Nowadays, lots of computational strategies exist to solve nonlinear problems such as contact, plasticity, damage or hyperelasticity problems. One of the most popular and general family of methods is the so-called "Newton-Krylov-Schur" (NKS) family. These methods rely on three points. First, the nonlinearity is handled at the global scale via the use of a Newton-type algorithm which leads to a sequence of linear systems. Second, domain decomposition methods (see for instance [1, 2] for a review) are used to condense these linear systems. Third, a Krylov iterative solver is chosen to solve condensed interface problems, allowing a high rate of parallelism (due to the additive structure of the interface problem). Among these methods, some are particularly popular as the Finite Element Tearing and Interconnecting method (FETI method) [3] which is classified as a dual method, the Balanced Domain Decomposition method (BDD method) [4, 5] considered as a primal method. These strategies allow to solve efficiently a lot of physical problems (due to the robustness of Newton-type methods) with high number of degrees of freedom (due to the efficiency of the domain decomposition methods) [6, 7, 8, 9]. Nonetheless, these methods may loose their efficiency when they encounter pathological phenomena such as local nonlinearities. In the field of continuum mechanics, such problems can be local buckling, localized damage, fracture, contact, plasticity, etc. Because the convergence of Newton-type algorithms is linked to the strongest nonlinearity

in the domain, local nonlinear phenomena may penalize the convergence of the global algorithm [10, 11]. In [11], the convergence of the NKS strategy is dramatically slowed down by local buckling, and a strategy which introduces nonlinear relocalization steps, inspired from the principles of the LaTIn method [12, 13], is designed and assessed. In this paper we adopt a different point of view.

Stating that standard NKS solvers do not exploit the domain decomposition in the nonlinear context, we propose to introduce the domain decomposition directly in the nonlinear formulation of the problem instead of only using it to solve linearized problems. First, the global nonlinear problem is decomposed into local (per subdomain) problems under a global constraint imposed via a Lagrange multiplier. Second, a condensation step allows to formulate the nonlinear problem in terms of the unknown Lagrange multiplier. Third a Newton-type algorithm is chosen to solve this nonlinear interface problem. It comes out that an iteration of our algorithm requires to solve a global linear interface problem and a set of independent nonlinear local problems (which are solved by another Newton-type solver) with prescribed Neumann interface conditions.

As exposed in this paper, the will to handle nonlinearity at its own local scale leads to a modification of the classical domain decomposition methods by the addition of a set of local nonlinear iterations (or relocalizations) which are parallel and not expensive. These nonlinear iterations allow to decrease the number of global resolutions and to reach significant CPU speed up. Note that even if the strategy is prompted by the problematic of localized non linearity, it is also efficient in the case of diffuse one.

After a short presentation of the reference problem (section 2), we introduce the nonlinear interface formulation of the initial problem (section 3). The solution to this nonlinear interface problem is presented in section 4. Section 5 provides first assessments of the method. Section 6 concludes the paper and gives few prospects.

2 Model Problem

Let us consider the mechanical equilibrium under small perturbations hypothesis of the structure Ω submitted to the body force f, while on the part $\partial_u\Omega$ (with $\text{mes}(\partial_u\Omega) \neq 0$) of $\partial\Omega$ the displacement u^d is prescribed, on the complementary of $\partial_u\Omega$, named $\partial_f\Omega$, the domain is submitted to the traction force g. Though the method we propose can be used on saddle point problems which embrace most anelastic problems [14], for easier explanation's sake we assume that the equilibrium problem can be written as a minimization problem.

$$E(u) = \min_{v \in \mathcal{U}} E(v) \quad \text{with} \quad E(v) = \int_\Omega e(\nabla^s v)\, d\Omega - \int_\Omega f.v\, d\Omega - \int_{\partial_f\Omega} g.v\, d\Gamma$$

where e is an energy density, function of the symmetric gradient $\nabla^s v = \frac{1}{2}\left(\nabla u + \nabla^T u\right)$. The space of research of the minimum $\mathcal{U}$ is the affine space of the kinematically admissible displacement fields, let $\mathcal{U}^0$ be the associated linear space:

$$\mathcal{U} = \left\{ u \in \left(\mathrm{H}^1(\Omega)\right)^3, \text{ such as } u = u^d \text{ on } \partial_u\Omega \right\}$$

$$\mathcal{U}^0 = \left(\mathrm{H}_0^1\left(\Omega, \partial_u\Omega\right)\right)^3 = \left\{u \in \left(\mathrm{H}^1\left(\Omega\right)\right)^3, \text{ such as } u = 0 \text{ on } \partial_u\Omega\right\}$$

The minimization problem leads to the variational formulation:

$$\begin{cases} u \in \mathcal{U} \\ \\ a(u, \delta u) = l(\delta u) \quad \forall\, \delta u \in \mathcal{U}_0 \end{cases} \tag{1}$$

with

$$a(u, \delta u) = \int_\Omega \nabla_u e(u).\delta u \, d\Omega \quad \text{and} \quad l(\delta u) = \int_\Omega f.\delta u \, d\Omega + \int_{\partial_f\Omega} g.\delta u \, d\Gamma$$

The nonlinearity of the form a may come from the non-quadratic character of the internal energy e. In mechanics, the form a typically writes:

$$a(u, \delta u) = \int_\Omega \sigma(\nabla^s u) : \nabla^s \delta u \, d\Omega$$

where σ is the Cauchy stress tensor.

3 Nonlinear formulation

A nonlinear interface formulation of the problem (1) is introduced. After a substructuration stage, a Lagrangian formulation of the decomposed problem is proposed. The saddle point problem is formulated in terms of interface variables via a condensation step. The condensation is based on the introduction of a nonlinear Neumann-to-Dirichlet operator. A particular attention will be paid to the handling of the local rigid body motions.

3.1 Decomposition of the problem

Let us consider the non-overlapping domain decomposition $\{\Omega_i\}_{i=1..m}$ of Ω. We note Γ_{ij} the interface between Ω_i and Ω_j, and Γ_i the interface of the subdomain Ω_i. $x_{|_i}$ and $x_{|_{ij}}$ denote the restriction on Γ_i and Γ_{ij} of a field x defined on Γ. Let us introduce $\mathcal{U}_i$, the functional space associated to the subdomain displacement field, and $\mathcal{U}_i^0$ the associated linear space.

$$\mathcal{U}_i = \left\{u \in \left(\mathrm{H}^1\left(\Omega_i\right)\right)^3, \text{ such as } u = u^d \text{ on } \partial_u\Omega_i\right\}$$

$$\mathcal{U}_i^0 = \left(\mathrm{H}_0^1\left(\Omega_i, \partial_u\Omega_i\right)\right)^3 = \left\{u \in \left(\mathrm{H}^1\left(\Omega_i\right)\right)^3, \text{ such as } u = 0 \text{ on } \partial_u\Omega_i\right\}$$

Let us define $\mathcal{I}_i = \left\{j \ , \ mes(\Gamma_{ij}) \neq 0\right\}$ the set of the subscripts of the substructures neighboring subdomain Ω_i, and introduce operators to handle the different fields:

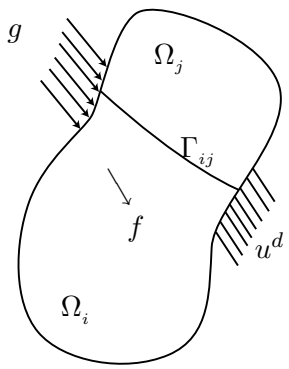
- Trace operator t_i that associates to a field defined on the subdomain Ω_i its trace on the subdomain interface Γ_i ($\mathcal{W}_i$ is the convenient trace space [18]):

$$t_i \; : \; \left\{ \begin{array}{ccc} \mathcal{U}_i & \longrightarrow & \mathcal{W}_i \\ u_i & \longmapsto & u_{i|_{\Gamma_i}} \end{array} \right.$$

- Signed injection operator A_i that associates to a field defined on a subdomain interface Γ_i a field defined on the whole interface Γ (trace space $\mathcal{W}$):

$$A_i \; : \; \left\{ \begin{array}{ccc} \mathcal{W}_i & \longrightarrow & \mathcal{W} \\ v_i & \longmapsto & v \end{array} \right. \qquad v \text{ being defined as: } v_{|_{ij}} = \left\{ \begin{array}{rl} 0 & \text{if} \quad j \notin \mathcal{I}_i \\ \epsilon_{ij} v_{i_{|_j}} & \text{if} \quad j \in \mathcal{I}_i \end{array} \right.$$

with $\epsilon_{ij} = \pm 1$ and $\epsilon_{ij} + \epsilon_{ji} = 0$.
Figure 1 summarize the notations.



$$\left\{ \begin{array}{rcl} \Gamma_{ij} & = & \bar{\Omega}_i \cap \bar{\Omega}_j \\ \Gamma_i & = & \bigcup_j \Gamma_{ij} \end{array} \right.$$

Figure 1: Domain decomposition, notations

3.2 Lagrangian formulation

The initial minimization problem is defined with an energy depending on the global field u. The substructuration and the introduction of associated independent fields u_i leads to local minimization problems under regularity constraint:

$$\min_{\hat{u} \in \hat{\mathcal{U}}} \sum_{i=1}^{m} E_i(u_i) \quad \text{under the constraint} \quad \sum_{i=1}^{m} A_i \, t_i \, u_i = 0,$$

where $\hat{u} = (u_i)_{i=1..n}$, $\hat{u} \in \hat{\mathcal{U}} = \Pi_i \, \mathcal{U}_i$, and:

$$E_i(v_i) = \int_{\Omega_i} e(\nabla^s v_i) \, d\Omega - \int_{\Omega_i} f.v_i \, d\Omega - \int_{\partial_f \Omega_i} g.v_i \, d\Gamma.$$

This constraint is taken into account by a standard dualization technique (we introduce the Lagrange multiplier λ and $\langle a, b\rangle_{\Gamma}$ the duality pairing between $\mathcal{W}$ and its dual $\mathcal{W}^*$).

$$\mathcal{L}\left(\hat{u}, \lambda\right) = \sum_{i=1}^{m} E_i(u_i) + \left\langle \sum_{i=1}^{m} A_i\, t_i\, u_i, \lambda \right\rangle_{\Gamma}$$

We then consider the saddle-point problem:

$$\text{Find } (\hat{u}, \lambda) \in \hat{\mathcal{U}} \times \mathcal{W}^* \quad \text{so that} \quad \mathcal{L}(\hat{u}, \lambda) = \inf_{\hat{w}} \sup_{\mu} \mathcal{L}(\hat{w}, \mu)$$

which leads to the following problem:

$$\left\{ \begin{array}{lrcll} \text{Find } (\hat{u}, \lambda) \in \hat{\mathcal{U}} \times \mathcal{W}^* & & & & \\ \forall\, i\,, & a_i\left(u_i, \delta u_i\right) & = & l_i\left(\delta u_i\right) - \langle A_i\, t_i\, \delta u_i, \lambda\rangle_{\Gamma} & \forall\, \delta u_i \in \mathcal{U}_i^0 \\ & \left\langle \sum_{i=1}^{n} A_i\, t_i\, u_i, \delta\lambda \right\rangle_{\Gamma} & = & 0 & \forall\, \delta\lambda \in \mathcal{W}^* \end{array} \right.$$

We assume classical locality property of the duality pairing:

$$\langle a, b\rangle_{\Gamma} = \sum_{1 \leqslant i < j \leqslant m} \langle a, b\rangle_{\Gamma_{ij}}$$

Introducing A_i^* the adjoint of A_i, with respect to $\langle a, b\rangle_{\Gamma}$ and denoting $\lambda_i = A_i^* \lambda$, the system writes

$$\left\{ \begin{array}{lrcll} \text{Find } (\hat{u}, \lambda) \in \hat{\mathcal{U}} \times \mathcal{W}^* & & & & \\ \forall\, i\,, & a_i\left(u_i, \delta u_i\right) & = & l_i\left(\delta u_i\right) - \langle t_i\, \delta u_i, \lambda_i\rangle_{\Gamma_i} & \forall\, \delta u_i \in \mathcal{U}_i^0 \\ & \left\langle \sum_{i=1}^{n} A_i\, t_i\, u_i, \delta\lambda \right\rangle_{\Gamma} & = & 0 & \forall\, \delta\lambda \in \mathcal{W}^* \end{array} \right.$$

The first equations are a set of local nonlinear problems with traction force λ_i (Neumann condition) on the interface and boundary conditions of the initial problem. A necessary condition of existence of a solution to such problems writes:

$$l_i(\rho_i) - \langle t_i\, \rho_i, \lambda_i\rangle_{\Gamma_i} = 0 \quad \forall \rho_i \in \mathcal{R}_i = Ker(\nabla^s) \cap \mathcal{U}_i^0 \tag{2}$$

Under this condition, solutions are defined up to an element of $\mathcal{R}_i$ (i.e. infinitesimal admissible rigid body motion). Finally, the decomposed problem writes:

$$\left\{ \begin{array}{lrcll} \text{Find } (\hat{u}, \lambda) \in \hat{\mathcal{U}} \times \mathcal{W}^* & & & & \\ \forall\, i\,, & \langle t_i\, \delta\rho_i, \lambda_i\rangle_{\Gamma_i} & = & l_i\left(\delta\rho_i\right) & \forall\, \delta\rho_i \in \mathcal{R}_i^0 \\ \forall\, i\,, & a_i\left(u_i, \delta u_i\right) & = & l_i\left(\delta u_i\right) - \langle t_i\, \delta u_i, \lambda_i\rangle_{\Gamma_i} & \forall\, \delta u_i \in \mathcal{U}_i^0 \\ & \left\langle \sum_{i=1}^{n} A_i\, t_i\, u_i, \delta\lambda \right\rangle_{\Gamma} & = & 0 & \forall\, \delta\lambda \in \mathcal{W}^* \end{array} \right.$$

3.3 Nonlinear condensation

3.3.1 General case

The principle of the condensation is to form an equivalent problem only posed in term of interface unknowns. The key point of this operation is to separate the equations arising from the saddle-point problem in two groups: the first group gathers the per-subdomain equations, the other one gathers the global equations that link subdomains together. In the case we consider, it writes:
• Local nonlinear problems:

$$\forall\, i\,, \quad a_i\left(u_i, \delta u_i\right) = l_i\left(\delta u_i\right) - \left\langle t_i \delta u_i, \lambda_i \right\rangle_{\Gamma_i} \qquad \forall\, \delta u_i \in \mathcal{U}_i^0 \tag{3}$$

• Interface equations:

$$\left\{ \begin{array}{lcll} \left\langle \displaystyle\sum_{i=1}^{m} A_i\, t_i\, u_i, \delta\lambda \right\rangle_{\Gamma} & = & 0 & \forall\, \delta\lambda \in \mathcal{W}^* \\ \\ \forall\, i\,, \quad \left\langle t_i\, \delta\rho_i, \lambda_i \right\rangle_{\Gamma_i} & = & l_i\left(\delta\rho_i\right) & \forall\, \delta\rho_i \in \mathcal{R}_i^0 \end{array} \right. \tag{4}$$

We assume that if the interface reaction λ_i verifies the admissibility condition (2), then the local nonlinear problems (3) have a unique solution u_i up to an element of $\mathcal{R}_i$ which we write R_i. It is thus possible to associate to λ_i one local field u_i solution to the nonlinear problem. We introduce a nonlinear operator $\mathbb{F}_i$ that links the interface reaction and the trace of the displacement field:

$$t_i\, u_i = \mathbb{F}_i\left(\lambda_i; f_i, g_i, u_i^d\right) + t_i\, \mathrm{R}_i \;\text{ with } u_i \text{ solution of (3)} \tag{5}$$

The computation of $\mathbb{F}_i\left(\lambda_i; f_i, g_i, u_i^d\right)$ involves the solution to nonlinear problems on the subdomains with prescribed Neumann conditions on the interface and given external loading. The nonlinear condensed problem then writes:

$$\left\{ \begin{array}{lcll} \left\langle \displaystyle\sum_{i=1}^{m} A_i \mathbb{F}_i\left(\lambda_i; f_i, g_i, u_i^d\right) + A_i t_i\, \mathrm{R}_i, \delta\lambda \right\rangle_{\Gamma} & = & 0 & \forall\, \delta\lambda \in \mathcal{W}^* \\ \\ \forall\, i\,, \qquad \left\langle t_i\, \delta\rho_i, \lambda_i \right\rangle_{\Gamma_i} & = & l_i\left(\delta\rho_i\right) & \forall\, \delta\rho_i \in \mathcal{R}_i^0 \end{array} \right. \tag{6}$$

3.3.2 Linear case

In the case where $a_i(.,.)$ is a bilinear form, the local problems are linear with respect to the interface conditions and to each boundary conditions, the influence of the interface can be decoupled from the influence of the external loadings, equation (5) writes:

$$t_i\, u_i = \mathbb{F}_i\, \lambda_i + b_i + t_i\, \mathrm{R}_i$$

where b_i is defined by $b_i = t_i\, \tilde{u}_i$ with $\tilde{u}_i$ one of the solution to:

$$\left\{ \begin{array}{l} \tilde{u}_i \in \mathcal{U}_i \\ a_i\left(\tilde{u}_i, \delta u_i\right) = l_i\left(\delta u_i\right) \quad \forall\, \delta u_i \in \mathcal{U}_i^0 \end{array} \right. \tag{7}$$

and operator $\mathbb{F}_i \lambda_i = t_i\, \tilde{\tilde{u}}_i$ with $\tilde{\tilde{u}}_i$ one of the solution to:

$$\left\{ \begin{array}{l} \tilde{\tilde{u}}_i \in \mathcal{U}_i^0 \\ a_i\left(\tilde{\tilde{u}}_i, \delta u_i\right) = -\left\langle t_i\, \delta u_i, \lambda_i \right\rangle_{\Gamma_i} \quad \forall\, \delta u_i \in \mathcal{U}_i^0 \end{array} \right. \tag{8}$$

which finally leads to the standard FETI formulation [3].

3.4 Discretization

The unknowns are approximated on each subdomain by a classical Finite Element Method. For simplicity reasons discretized fields are written the same way as their continuous version. We assume that meshes are conforming and that the duality pairing can be represented by classical Euclidian dot product (see [18, 19] for other choices), hence the discrete form of A_i^* is A_i^T. We also denote $\mathbb{G}_i = A_i\, t_i\, \mathbb{R}_i$ ($\mathbb{R}_i$ is a basis of $\mathcal{R}_i$ and R_i is represented by $\mathbb{R}_i \alpha_i$), F_i the generalized force vector, $e_i = \mathbb{R}_i^T F_i$, and $\mathbb{K}_i(u_i)$ the internal reaction vector. $\mathbb{G}$, α and e results from the concatenation of $\mathbb{G}_i$, α_i and e_i. Then the discretized form of the condensed problem (6) writes:

$$\left\{ \begin{array}{l} \displaystyle\sum_{i=1}^{m} \left(A_i\, \mathbb{F}_i\left(\lambda_i; F_i, u_i^d\right) + \mathbb{G}_i\, \alpha_i \right) = 0 \\ \\ \forall\, i\,, \;\; \mathbb{G}_i^T \lambda = e_i \end{array} \right. \tag{9}$$

or in a shorter way,

$$\left\{ \begin{array}{lcl} \mathbb{F}\left(\lambda; F, u^d\right) + \mathbb{G}\, \alpha & = & 0 \\ \\ \mathbb{G}^T \lambda & = & e \end{array} \right. \tag{10}$$

with $\mathbb{G} = (\mathbb{G}_1, ..., \mathbb{G}_m)$, $\alpha^T = \left(\alpha_1^T, ..., \alpha_m^T\right)$ and $e^T = \left(e_1^T, ..., e_m^T\right)$.

4 Solution to the nonlinear condensed problem

4.1 A local/global Newton-Raphson algorithm

The nonlinear interface problem is solved by a Newton-type algorithm. Considering the solution to the nonlinear system (10), an iteration of the Newton-Raphson algorithm is obtained by a Taylor expansion with respect to (λ, α). It leads to the sequence of tangent linear problems:

$$\lambda^{n+1} = \lambda^n + \delta\lambda^n \quad , \quad \alpha^{n+1} = \alpha^n + \delta\alpha^n \quad ,$$

$$\begin{pmatrix} D_\lambda \mathbb{F}\left(\lambda^n; F, u^d\right) & \mathbb{G} \\ \mathbb{G}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta\lambda^n \\ \delta\alpha^n \end{pmatrix} = \begin{pmatrix} 0 \\ e \end{pmatrix} - \begin{pmatrix} \mathbb{F}\left(\lambda^n; F, u^d\right) + \mathbb{G}\alpha^n \\ \mathbb{G}^T\lambda^n \end{pmatrix},$$

Note that if we choose λ^0 satisfying $\mathbb{G}^T\lambda^0 = e$, then previous system writes :

$$\begin{pmatrix} D_\lambda \mathbb{F}\left(\lambda^n; F, u^d\right) & \mathbb{G} \\ \mathbb{G}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \delta\lambda^n \\ \delta\alpha^n \end{pmatrix} = - \begin{pmatrix} \mathbb{F}\left(\lambda^n; F, u^d\right) + \mathbb{G}\alpha^n \\ 0 \end{pmatrix}, \tag{11}$$

and automatically

$$\mathbb{G}^T\lambda^n = e. \tag{12}$$

The solution to the tangential problem requires the computation of the tangential operator $D_\lambda \mathbb{F}\left(\lambda^n; F, u^d\right)$ and of the residual $r^n = -\left(\mathbb{F}\left(\lambda^n; F, u^d\right) + \mathbb{G}\alpha^n\right)$.

4.2 Computation of the residual: a nonlinear relocalization step

For a given λ^n satisfying $\mathbb{G}^T\lambda^n = e$, the computation of the residual

$$r^n = -(\mathbb{F}(\lambda^n; F, u^d) + \mathbb{G}\alpha^n)$$

requires the solution to local nonlinear problems and assembly operation. The interface residual writes:

$$r^n = -\sum_{i=1}^{m} A_i \, t_i \, u_i \tag{13}$$

where u_i is the solution to the local nonlinear problem:

$$\mathbb{K}_i\left(u_i\right) = F_i - t_i^T \lambda_i^n \quad \text{for a given } \lambda_i^n = A_i^T\lambda^n \tag{14}$$

These local nonlinear problems can be solved by any nonlinear solver. For Newton-type algorithms, it leads to the following sequence of linear systems:

$$D_{u_i}\mathbb{K}_i(u_i^n) \,.\, \delta u_i^n = \left(F_i - t_i^T A_i^T \lambda^n - \mathbb{K}_i(u_i^n)\right) \quad ; \quad u_i^{n+1} = u_i^n + \delta u_i^n \quad , \tag{15}$$

where $D_{u_i}\mathbb{K}_i(u_i^n)$ denotes the tangential operator.
This system is solved using the pseudo-inverse [15] and associated orthogonality condition:

$$\begin{aligned} &\tilde{\delta u_i}^n = \left(D_{u_i}\mathbb{K}_i(u_i^n)\right)^+ . \left(F_i - t_i^T A_i^T \lambda^n - \mathbb{K}_i(u_i^n)\right) + \mathbb{R}_i \alpha_i \\ &\mathbb{R}_i^T . \left(F_i - t_i^T A_i^T \lambda^n - \mathbb{K}_i(u_i^n)\right) = 0 \end{aligned} \tag{16}$$

Due to the definition of $\mathbb{R}_i$ we have $\mathbb{R}_i^T.\mathbb{K}_i(u_i^n) = 0$. The previous condition sums up to:

$$\mathbb{R}_i^T t_i^T A_i^T \lambda^n = \mathbb{R}_i^T F_i \quad \Leftrightarrow \quad \mathbb{G}_i^T \lambda^n = e_i \tag{17}$$

Because $\mathbb{G}$ is not varying from one iteration to another, this is exactly the condition (12) automatically satisfied by the solution to the tangential problem (11).

4.3 Computation of the tangential operator

In order to simplify the notation, in this subsection the dependence of $\mathbb{F}$ with respect to the external loadings is not mentioned and the iteration number n is omitted. Thus the condensed operator simply writes $\mathbb{F}(\lambda)$.
For given λ satisfying $\mathbb{G}^T\lambda = e$, the solution to the nonlinear interface problem implies the inversion of a linear system involving the tangential operator associated to $\mathbb{F}$. The tangential operator of $\mathbb{F}$, $D_\lambda \mathbb{F}(\lambda)$ is defined by:

$$D_\lambda \mathbb{F}(\lambda) \,.\, \delta\lambda = \lim_{k \to 0} \frac{\mathbb{F}(\lambda + k\,\delta\lambda) - \mathbb{F}(\lambda)}{k} = \sum_i A_i \,.\, D_{\lambda_i} \mathbb{F}_i(\lambda_i) \,.\, \delta\lambda_i$$

with the following local contributions:

$$D_{\lambda_i} \mathbb{F}_i(\lambda_i) \,.\, \delta\lambda_i = \lim_{k \to 0} \frac{1}{k} \left(\mathbb{F}_i(\lambda_i + k\,\delta\lambda_i) - \mathbb{F}_i(\lambda_i) \right)$$

In order to compute these local contributions, let us introduce $\lambda_i^* = \lambda_i + k\,\delta\lambda_i$ and consider the following associated problems:

$$\mathbb{K}_i(u_i^*) = F_i - t_i^T \lambda_i^* \quad \text{and} \quad \mathbb{K}_i(u_i) = F_i - t_i^T \lambda_i$$

By the definition of $\mathbb{F}_i$ we have :

$$\mathbb{F}_i(\lambda_i^*) - \mathbb{F}_i(\lambda_i) = t_i \left(u_i^* - u_i \right) - t_i \mathbb{R}_i \left(\alpha_i^* - \alpha_i \right)$$

Introducing δu_i defined by $k\,\delta u_i = u_i^* - u_i$ and subtracting the two problems we obtain:

$$\frac{\mathbb{K}_i(u_i + k\,\delta u_i) - \mathbb{K}_i(u_i)}{k} = -t_i^T \,\delta\lambda_i$$

Passing to the limit ($k \to 0$) and introducing $\delta\alpha_i$ defined by $k\,\delta\alpha_i = \alpha_i^* - \alpha_i$, we obtain:

$$t_i \,.\, \delta u_i = D_\lambda \mathbb{F}_i(\lambda_i) \,.\, \delta\lambda_i - t_i \mathbb{R}_i \,.\, \delta\alpha_i$$

with δu_i one of the solution to the problem $D_{u_i} \mathbb{K}_i(u_i) \,.\, \delta u_i = -t_i^T \,.\, \lambda_i$.
Hence, on each subdomain, the tangential operator of the nonlinear interface operator is the Schur complement associated to the tangential operator of the nonlinear problem. The resulting tangential linear problem is then a typical condensed linear problem with dual formulation (FETI-type operator but modified right-hand-side). Hence all standard FETI ingredients may be used to solve it (see appendix).

4.4 Nonlinear dual domain decomposition algorithm

Finally, the algorithm is summarized in algorithm 1.

Algorithm 1: Nonlinear dual domain decomposition algorithm

– Initialization : λ^0 , such that $\mathbb{G}^T\lambda^0 = e$

for *n=0* ***to*** m **do**

– Solution to local nonlinear problems: $\mathbb{K}_i\left(u_i^n\right) = F_i - t_i^T . \lambda_i^n$
(Newton-type iterations)

– Assembly of local residual $r^n = -\sum_i A_i . t_i . u_i^n$

– Solution to the global interface problem $\begin{pmatrix} D_\lambda \mathbb{F}\left(\lambda^n\right) & \mathbb{G} \\ \mathbb{G}^T & 0 \end{pmatrix} . \begin{pmatrix} \delta\lambda^n \\ \delta\alpha^n \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \end{pmatrix}$
(FETI-type iterations - see Appendix)

– Update of the interface unknown $\lambda^{n+1} = \lambda^n + \delta\lambda^n$

end

Among the different possibility to initialize λ^0, one can use a classical FETI coarse problem

$$\lambda^0 = \mathbb{G}\left(\mathbb{G}^T\mathbb{G}\right)^{-1} e \tag{18}$$

or any low-precision FETI computation.

5 Numerical assessments

To assess the algorithm, we consider damaging elasticity mechanical problems. The Cauchy stress tensor writes:

$$\sigma(u(t)) = (1 - d(u(t)))\, \mathcal{K} : \nabla^s(u(t))$$

where the scalar damage variable $d(u(t))$ is given by:

$$d(u(t)) = \frac{\sqrt{Y_m(u(t))} - \sqrt{Y_0}}{\sqrt{Y_c} - \sqrt{Y_0}}.$$

Y_m denotes the maximum of the thermodynamical force Y over the history of the resolution.

$$Y_m(u(t)) = \sup_{\tau \leq t} Y(u(\tau)) \quad \text{with} \quad Y(u(t)) = \frac{1}{2}\epsilon(u(t)) : \mathcal{K} : \epsilon(u(t))$$

For a detailed study of this kind of behavior, see [16].
Figure 2 presents the wheel of a turbine (courtesy of Frederic Feyel, ONERA). The structure is meshed using P2-Lagrange triangle elements (6 nodes). The problem contains 33000 degrees of freedom and is decomposed in m ($m = 2, 4,$ or 6) subdomains. The structure is clamped on its left side and displacement is prescribed on its right side.

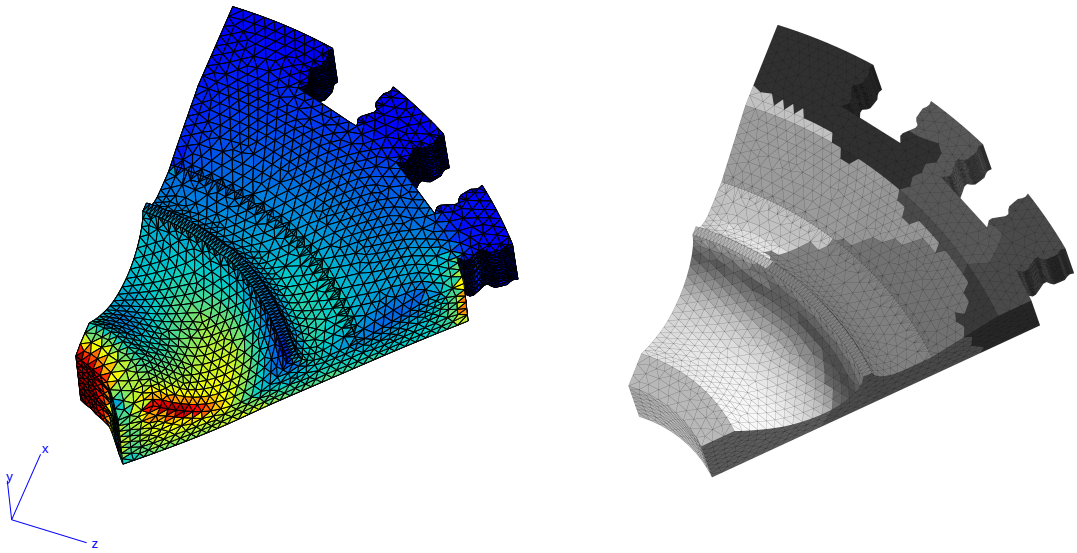


Figure 2: (a) Damage map, (b) 6-sd decomposition

A Newton-Raphson with update of operators at each iteration is used for both global and local nonlinear solvers. A FETI linear solver with Dirichlet preconditioner and associated coarse grid projector is used as linear solver for the solution to global linear problems. Table 1 summarizes numerical results for the two methods and for equivalent final precision (relative nonlinear residual 10^{-7}). Column 3 gives the number of global

Decomp.	Solver	# global NL it.	# local NL it. (max./sd)	# local update (av./sd)	CPU time	Speed up
2	NKS	5	–	–	238 s	
	Dual NL DD	1	7	7	126 s	47 %
4	NKS	5	–	–	145 s	
	Dual NL DD	1	9	7.5	80 s	45 %
6	NKS	5	–	–	116 s	
	Dual NL DD	1	8	6.8	66 s	43 %

Table 1: Performance results on the wheel problem

Newton-Raphson iterations. This number is obviously independent on the number of subdomains for standard Newton-Krylov-Schur (NKS) approach, but might be varying for the Dual nonlinear domain decomposition (Dual NL DD) approach. In this case, we get the optimal reduction of global Newton-Raphson iterations since independently on

the decomposition our algorithm converges in only 1 global iteration, plus the initialization step for the computation of λ^0, instead of 5 for the standard approach. Of course, our algorithm requires local nonlinear iterations, which also means local reactualization of operators. Moreover depending on the decompositions, such nonlinear relocalization step might be poorly balanced between subdomains. Indeed, in the case of the two-subdomain substructuration, the damage is equilibrated between the substructures, hence the convergence rate is the same on each subdomain. On the other hand, when the number of subdomain is increasing, some subdomains are less damaging than others. Local convergence rates are different and consequently the nonlinear relocalization steps are unbalanced between subdomains (see fig. 3). The lack of balance can be measured in Table 1 when the average of the number of local iterations per subdomain (column 5) is significantly different from the maximum number of local iterations (column 4).
In term of CPU time, the speedup is almost independent on the substructuration, around 45%. A finer control on the different stopping criteria of the algorithms (local nonlinear solver and global linear interface solver) and the utilization of a not fully updated local Newton-Raphson algorithm should improve this result.

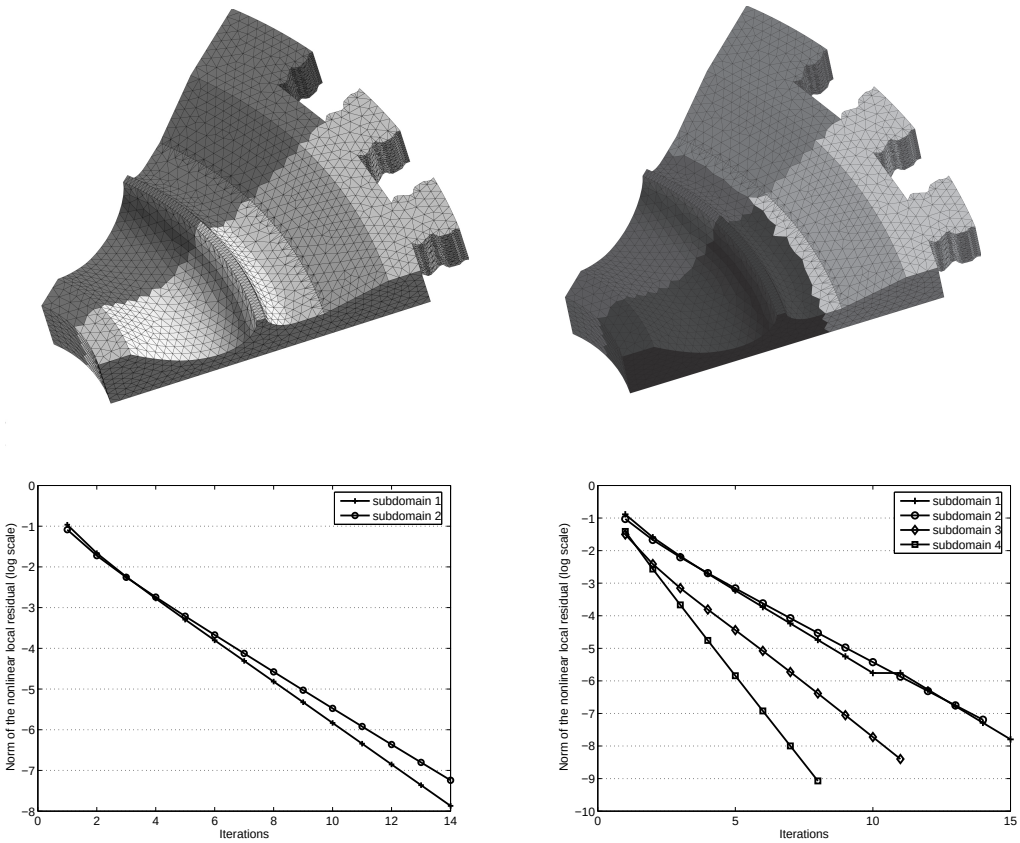


Figure 3: (left) 2 subdomains case, (right) 4 subdomains case, (top) decomposition, (bottom) convergence of the relocalization problems

6 Conclusion

In this paper a dual nonlinear domain decomposition method is presented. We show that using it as a nonlinear solver leads to competitive strategies for the solution to nonlinear problems (particularly in case where the nonlinearity is localized) as often encountered in mechanics. The main idea is to use domain decomposition methods to split a global nonlinear problem into a set of local nonlinear problems. One iteration of our algorithm consists of the solution to a condensed linear problem (global step) and to independent local nonlinear resolutions (nonlinear relocalization step). The first interest of this approach is to bring back the handling of the nonlinearity from a global scale to a local one. Consequently nonlinear iterations are effected mainly in the subdomains where nonlinearities are concentrated and much less on the whole domain. As local nonlinear iterations are fully independent, it is possible to solve them in parallel and if necessary with adapted nonlinear algorithm.

First results are very encouraging since on a realistic test problem the speedup of 45% is reached. However many questions remain to be studied. Among others, let us cite the problem related to the potential excitation of non-physical nonlinearities at the subdomain scale, the redefinition of the concept of load balancing and the extension to finite transformations (with an appropriate coarse problem to handle the local rigid body motions).

In order to control the development of the nonlinearities, the use of mixed boundary conditions to glue subdomains together is under development. This approach is based on the use of an augmented Lagrangian [17, 18] that leads to a linear interface problem similar to the 2-Lagrangian-Multiplier approach [20] and to nonlinear relocalization with mixed (Robin) interface conditions.

References

[1] P. Gosselet, C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, vol. 13(4), 515-572, 2007.

[2] F. Magoulès, F-X. Roux. Lagrangian formulation of domain decomposition methods: A unified theory. *Applied Mathematical Modelling*, vol. 30(7), 593-615, 2006.

[3] C. Farhat, F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.*, vol. 32, 1205-1227, 1991.

[4] J. Mandel. Balancing domain decomposition *Comput. Methods Appl. Mech. Engrg.*, vol. 9, 233-241, 1993.

[5] P. Le Tallec, Y.-H. De Roeck, M. Vidrascu Domain decomposition methods for large linearly elliptic three dimensional problems. *J. Com. Appli. Math* , vol. 34, 93-117, 1991.

[6] P. Gosselet, C. Rey, DJ. Rixen. On the initial estimate of interface forces in FETI methods. *J. Comp Meth. Appl. Mech. Eng.*, vol. 192 (25), 2749-2764, 2003.

[7] F. Risler, C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems, *Numerical Algorithms*, vol. 23, 1-30, 2000.

[8] C. Farhat, K. Pierson, M. Lesoinne. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems, *J. Comp Meth. Appl. Mech. Eng.,* vol. 184(2-4), 333-374, 2000.

[9] Lene F, Rey C. Some strategies to compute elastomeric lamified composite structures. *Composite Structures*, vol. 54 (2-3), 231-241, 2001.

[10] X-C. Cai, D. E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Compu.*, vol. 24, 183-200, 2002.

[11] P. Cresta, O. Allix, C. Rey, S. Guinard. Nonlinear localization strategies for domain decomposition methods: application to post-buckling analysis. *Comput. Methods Appl. Mech. Engrg.*, 196, 1436-1446, 2007.

[12] P. Ladevèze. Nonlinear Computationnal Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation. *Springer Verlag*, 1999.

[13] P. Ladevèze, D. Néron, P. Gosselet. On a mixed and multiscale domain decomposition method, *Computer Methods in Applied Mechanics and Engineering*, vol. 196(8), 1526-1540, 2006.

[14] J. C. Simo, T. J. R. Hughes Computational inelasticity,. *Interdisciplinary Applied Mathematics*, vol 7. Springer 1997.

[15] C. Farhat, M. Géradin On the computation of the null spaces and generalized inverse of large matrix and the zero energy modes of a structure. *Technical Report, Center for aerospace structures*, CU-CAS-96-15, may 1996.

[16] J. Besson, G. Cailletaud, J. L. Chaboche, S. Forest. Mécanique non linéaire des matériaux. *Hermès - Lavoisier.*

[17] M. Fortin, R. Glowinski. Méthodes de Lagrangien augmenté. Applications à la résolution numérique de problèmes aux limites. *Dunod-Méthodes mathématiques de l'informatique*, 1982.

[18] P. Le Tallec, T. Sassi. Domain Decomposition with nonmathching grids: Augmented Lagrangian Approach *Mathematics of Computation*, vol. 64(212), 1367-1396, 1995.

[19] C.Bernadi, T. Chacòn Rebollo, E. Chacòn Vera. A FETI method with a mesh independent condition number for the iteration matrix, *Computer Methods in Applied Mechanics and Engineering*, vol. 197, 1410-1429, 2008.

[20] F-X. Roux, F. Magoulès, L. Series, Y. Boubendir Approximation of optimal interface boundary conditions for Two-Lagrange-Multilpier FETI Method. *Lectures Notes in Computationsl Science and engineering*, vol. 40, 283-290, 2005.

APPENDIX - Solution to the tangential problem

This part concerns the solution to the tangential interface problem (11), which is a classical issue in DDM. To simplify the notations, let us consider the following linear problem where $\mathbb{F}$ is a linear operator:

$$\begin{pmatrix} \mathbb{F} & \mathbb{G} \\ \mathbb{G}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} b \\ e \end{pmatrix} \tag{19}$$

This problem is solved using Krylov iterative solvers which are commonly used in classical domain decomposition methods [1] mainly because they only require operator/vector products that can be easily effected in parallel, thanks to the additive structure of the Schur complement. Coupling the dual approach to conjugate gradient leads to the well-known FETI method [3] which provides two important features:

- the existence of efficient preconditioners based on the scaled assembly of approximations of local primal Schur complement;
- the use of a projector orthogonal to the admissible infinitesimal rigid body motions which allows to use pseudo-inversion to solve the local problems and which provides a coarse problem that ensures the scalability of the method.

Projection is used to satisfy the constraint $\mathbb{G}^T\lambda = e$ by decomposing λ in two contributions:

$$\begin{cases} \lambda = \lambda^0 + \mathbb{P}.\lambda^* \\ \mathbb{G}^T\lambda^0 = e \\ \mathbb{G}^T\mathbb{P} = 0 \end{cases} \tag{20}$$

A possible definition for initialization and projector is:

$$\begin{cases} \lambda^0 = \mathbb{Q}\mathbb{G}\left(\mathbb{G}^T\mathbb{Q}\mathbb{G}\right)^{-1} . \, e \\ \mathbb{P} = \mathbb{I} - \mathbb{Q}\mathbb{G}\left(\mathbb{G}^T\mathbb{Q}\mathbb{G}\right)^{-1}\mathbb{G}^T \end{cases} \tag{21}$$

where $\mathbb{Q}$ is a parameter of the method. The remaining system to solve writes:

$$\left(\mathbb{P}^T\mathbb{F}\,\mathbb{P}\right) . \, \lambda^* = \mathbb{P}^T\left(b - \mathbb{F} \, . \, \lambda^0\right) \tag{22}$$

The method is summarized in algorithm 2.

Algorithm 2: Linear FETI Solver

Initialization:

Initialization:	λ^{00} arbitrary.
Projection:	$\lambda^{0} = \mathbb{Q}\mathbb{G}\left(\mathbb{G}^T\mathbb{Q}\mathbb{G}\right)^{-1} . e + \mathbb{P} . \lambda^{00}$
Residual:	$r^{0} = \mathbb{P}^T(b - \mathbb{F} . \lambda^{0})$
Preconditionner:	$z^{0} = \tilde{\mathbb{F}}^{-1} r^{0}$

Iterations:

for *i=1 to m* **do**

Update :	$\lambda^{i+1} = \lambda^{i} + \alpha^{i} w^{i}$	, $\alpha^{i} = \dfrac{(r^{i}, z^{i})}{\left(\mathbb{P}^T\mathbb{F}\mathbb{P}.w^{i}, w^{i}\right)}$
Update :	$r^{i+1} = r^{i} - \alpha^{i} \mathbb{P}^T\mathbb{F}\mathbb{P} . w^{i}$	
Preconditioner:	$z^{i+1} = \tilde{\mathbb{F}}^{-1} . r^{i+1}$	
Orthogonalization:	$w^{i+1} = z^{i+1} + \sum_{j=1}^{i} \beta_i^j w^{j}$	, $\beta_i^j = -\dfrac{(r^{i+1}, z^{i+1})}{(r_i, z_i)}$

end