



HAL
open science

A 2-level and mixed domain decomposition approach for structural analysis

David Dureisseix, Pierre Ladevèze

► **To cite this version:**

David Dureisseix, Pierre Ladevèze. A 2-level and mixed domain decomposition approach for structural analysis. Contemporary mathematics, 1998, 218, pp.238-245. 10.1090/conm/218/03014 . hal-00321763

HAL Id: hal-00321763

<https://hal.science/hal-00321763>

Submitted on 22 Jun 2018

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

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



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A 2-level and mixed domain decomposition approach for structural analysis

David Dureisseix¹ and Pierre Ladevèze¹

February 6, 1998

¹ LMT Cachan (E.N.S. de Cachan / Université Paris 6 / C.N.R.S.), 61 Avenue du Président Wilson, F-94235 CACHAN CEDEX, FRANCE

1991 MSC: Primary 65N55, 73C35; Secondary 73V05, 65F10

This is a preprint of an article originally published as: David Dureisseix, Pierre Ladevèze, A 2-level and mixed domain decomposition approach for structural analysis, *Contemporary Mathematics* 218:238-245, AMS, 1998. DOI: 10.1090/conm/218/03014

1 Introduction

When using domain decomposition methods without overlapping, one can focus on displacements, such as primal approaches, [11]... , or on efforts, such as dual approaches, [6]. Since the LATIN approach used herein allows interfaces to play a major role, both displacements and efforts are the unknowns; it is a “mixed” approach. A general drawback with domain decomposition methods is the decrease in convergence as increases the number of substructures. Using a global mechanism to propagate information among all substructures can eliminate this drawback.

We are proposing herein to take into account the introduction of two scales when decomposing the structure into substructures and interfaces. As a first step, the implemented version is concerned with linear elasticity. The large scale problem is then used to build a global exchange of information and therefore to improve performance. Moreover, comparisons with other decomposition methods, and in particular with several variants of the FETI method, are proposed.

2 Formulation of the problem

The studied structure is seen as the assembly of two mechanical entities: sub-structures Ω^E , $E \in \mathbf{E}$, and interfaces $L^{EE'}$. Each possess its own variables and equations. The principles of this one-level approach have been described in [9], its feasibility has been shown in [10], and [2] proposes some significant examples.

Since we are dealing herein with linear elasticity, only the final configuration is of interest.

2.1 Substructure behaviour

Each substructure Ω^E is submitted to the action of its environment (neighbouring interfaces): an effort \underline{F}^E and a displacement field \underline{W}^E on its boundary $\partial\Omega^E$. Eventually, \underline{f}_d is a prescribed body force (Figure 1).

For each $E \in \mathbf{E}$, $(\underline{W}^E; \underline{F}^E)$ has to satisfy:

- kinematic equations:

$$\exists \underline{U}^E \in \mathcal{U}^E, \quad \varepsilon^E = \varepsilon(\underline{U}^E) \quad \text{and} \quad \underline{U}^E|_{\partial\Omega^E} = \underline{W}^E \quad (1)$$

where \mathcal{U}^E is the set of displacement fields defined on Ω^E which possess a finite energy, and ε^E is the associated strain.

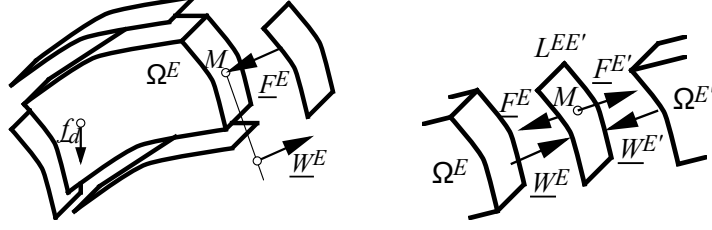


Figure 1: Substructure and interface

- equilibrium equations: a stress field σ^E balances \underline{f}_d and \underline{F}^E , i.e.:

$$\forall \underline{U}^* \in \mathcal{U}^E, \quad \int_{\Omega^E} \text{Tr}[\sigma^E \varepsilon(\underline{U}^*)] d\Omega = \int_{\Omega^E} \underline{f}_d \cdot \underline{U}^* d\Omega + \int_{\partial\Omega^E} \underline{F}^E \cdot \underline{U}^* dS \quad (2)$$

- constitutive relation: herein, the behaviour is linear and elastic (\mathbf{K} denotes Hooke's tensor) and

$$\sigma^E = \mathbf{K} \varepsilon^E \quad (3)$$

\mathbf{s} denotes the set of unknowns $(\underline{W}^E, \underline{F}^E, \underline{U}^E, \sigma^E)$ for $E \in \mathbf{E}$, that characterises the state of all substructures.

2.2 Interface behaviour

The state of the liaison between two substructures Ω^E and $\Omega^{E'}$ is defined by values on its surface of both the displacements and efforts $(\underline{W}^E; \underline{F}^E)$ and $(\underline{W}^{E'}; \underline{F}^{E'})$ (see Figure 1). For a perfect liaison, they must satisfy:

$$\underline{F}^E + \underline{F}^{E'} = 0 \quad \text{and} \quad \underline{W}^E = \underline{W}^{E'} \quad (4)$$

Of course, other kinds of liaison can be expressed, such as the prescribed effort liaison, the prescribed displacement liaison, and the unilateral contact liaison with or without friction, as described in [9], [2]. Here, we are only dealing with perfect interfaces that continuously transfer both efforts and displacements.

2.3 Description of the one-level algorithm

According to the framework of LARge Time INcrement (LATIN) methods, equations are split into two groups in order to separate difficulties, [10]:

- $\mathbf{\Gamma}$ is the set of unknowns \mathbf{s} satisfying each interface behaviour (4), and
- \mathbf{A}_d is the set satisfying each substructure behaviour (1), (2), (3).

The solution \mathbf{s}_{ex} searched is then the intersection of \mathbf{A}_d and $\mathbf{\Gamma}$. A two-stage algorithm successively builds an element of \mathbf{A}_d and an element of $\mathbf{\Gamma}$. Each stage involves a search direction; these are the parameters of the method:

- the local stage uses the search direction \mathbf{E}^+ : $(\hat{\underline{F}} - \underline{F}) - k(\hat{\underline{W}} - \underline{W}) = 0$

Finding $\hat{\mathbf{s}} \in \mathbf{\Gamma}$ in such a way that $\hat{\mathbf{s}} - \mathbf{s}_n$ belongs to the search direction \mathbf{E}^+ is a local problem on the interfaces. For instance, with perfect interfaces, the solution is explicitly written: $\hat{\underline{W}} = \hat{\underline{W}}' = \frac{1}{2}[(\underline{W} + \underline{W}') - k^{-1}(\underline{F} + \underline{F}')] and $\hat{\underline{F}} = -\hat{\underline{F}}' = \frac{1}{2}[(\underline{F} - \underline{F}') - k(\underline{W} - \underline{W}')]]. It can easily be parallelised.$$

- the linear stage uses the search direction \mathbf{E}^- : $(\underline{F} - \hat{\underline{F}}) + k(\underline{W} - \hat{\underline{W}}) = 0$

Finding $\mathbf{s}_{n+1} \in \mathbf{A}_d$ in such a way that $\mathbf{s}_{n+1} - \hat{\mathbf{s}}$ belongs to \mathbf{E}^- is a global problem on each substructure. When using the search direction, (2) is an elasticity-like problem on each substructure, with Robin boundary conditions. It can be solved concurrently once the substructures have been distributed among the available processors, along with their neighbouring interfaces.

Finally, a convergence check can be built with $\|\hat{\mathbf{s}} - \mathbf{s}_n\|$. More details for this one-level approach can be found in [10]. In the case of linear elasticity, this algorithm is similar to the one proposed in [8], [12], [7], i.e. it is one version of Uzawa algorithm.

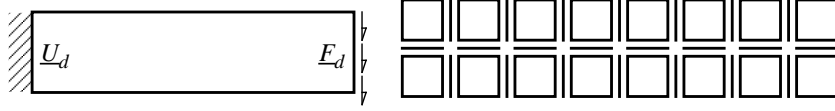


Figure 2: Model problem and example of decomposition into 16 substructures and interfaces

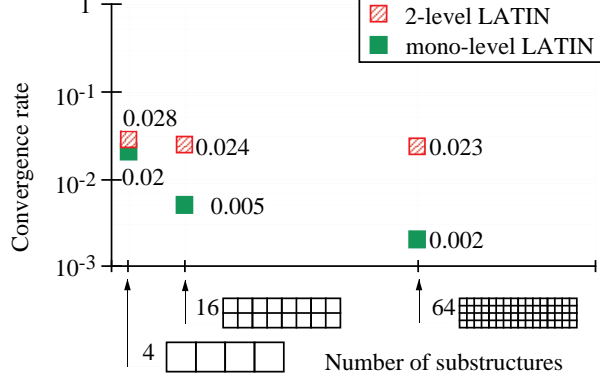


Figure 3: Convergence rate versus number of substructures

3 A 2-level extension

Let us first consider the model problem of a slendered bidimensional structure submitted to a parabolic bending loading (see Figure 2). The reference $(\underline{U}; \sigma)_{\text{ref}}$ here is the direct finite element solution without decomposition. It allows us to define the convergence rate in energy norm:

$$\tau = -\log \frac{e_{n+1}}{e_n} \quad \text{where} \quad e_n^2 = \frac{\frac{1}{2} \int_{\Omega} \text{Tr}[(\sigma_n - \sigma_{\text{ref}}) \mathbf{K}^{-1} (\sigma_n - \sigma_{\text{ref}})] d\Omega}{\frac{1}{2} \int_{\Omega} \text{Tr}[\sigma_{\text{ref}} \mathbf{K}^{-1} \sigma_{\text{ref}}] d\Omega}$$

Figure 3 presents the averaged convergence rate (up to convergence: $e_n \leq 0.1\%$) versus the number of substructures. It illustrates a well-known behaviour of domain decomposition methods: slowing the convergence rate when increasing the number of subdomains, [1]. To remedy such a drawback, we select herein to express the solution on two different scales:

$$(\underline{U}^E; \sigma^E) = (\underline{U}_1^E; \sigma_1^E) + (\underline{U}_2^E; \sigma_2^E) \quad (5)$$

1 and 2 denote unknowns related to large scale (effective quantities) and related to corrections on the fine scale respectively. The large scale problem is kept global in order to build the global information exchange mechanism, while the fine scale is managed with the previous substructuring technique.

Each level can arise from a different model for the structure; here, they are related to 2 different meshes with embedded elements. Let Ω_1 and Ω_2 denote these meshes. The principles of such a technique are described in [4]. As in the multigrid terminology, information transfer between levels is performed with a prolongation operator, \mathbf{P} , and a restriction operator, $\mathbf{R} = \mathbf{P}^T$. $(\underline{U}, \bar{\sigma})$ is the effective part of the solution, i.e. the part defined on the mesh Ω_1 (then, $\underline{U}_1^E = \mathbf{P}^E \underline{U}$ and $\bar{\sigma} = \sum_{E \in \mathbf{E}} \mathbf{R}^E \sigma_1^E$). With embedded grids, the

prolongation is straightforward and performed with a classical hierarchical finite element projection, as hierarchical bases are used for splitting \mathcal{U}^E into \mathcal{U}_1 and \mathcal{U}_2^E . With such a splitting, the global equilibrium equations become:

$$\begin{aligned} \forall \underline{U}_1^* \in \mathcal{U}_1, \quad \forall \underline{U}_2^* \in \mathcal{U}_2^E, \quad \sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\sigma(\varepsilon(\underline{U}_1^*) + \varepsilon(\underline{U}_2^*))] d\Omega = \\ = \sum_{E \in \mathbf{E}} \int_{\Omega^E} (\underline{f}_d \cdot \underline{U}_1^* + \underline{f}_d \cdot \underline{U}_2^*) d\Omega + \sum_{E \in \mathbf{E}} \int_{\partial\Omega^E} \underline{F}_2^E \cdot \underline{U}_2^* dS \quad (6) \end{aligned}$$

with $\sigma = \sigma_1 + \sigma_2 = \mathbf{K}\varepsilon(\underline{U}_1) + \mathbf{K}\varepsilon(\underline{U}_2)$, and the search direction on fine scale $\underline{F}_2^E = \hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E - k\underline{W}_2^E$, with $\underline{W}_2^E = \underline{U}_2^E|_{\partial\Omega^E}$, it leads to:

- on the fine scale 2, for each substructure Ω_2^E , the stress field σ_2^E also has to balance $-\sigma_1^E = -\mathbf{K}\varepsilon(\underline{U}_1|_{\Omega^E}) = -\mathbf{K}\varepsilon(\mathbf{P}^E\underline{U})$:

$$\begin{aligned} \forall \underline{U}_2^* \in \mathcal{U}_2^E, \quad & \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_2)\mathbf{K}\varepsilon(\underline{U}_2^*)]d\Omega + \int_{\partial\Omega^E} \underline{U}_2 \cdot k\underline{U}_2^*dS = \\ & = \int_{\Omega^E} \underline{f}_d \cdot \underline{U}_2^*d\Omega + \int_{\partial\Omega^E} (\hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E) \cdot \underline{U}_2^*dS - \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_1)\mathbf{K}\varepsilon(\underline{U}_2^*)]d\Omega \quad (7) \end{aligned}$$

The discretised displacement-oriented formulation of the problem (7) is:

$$([K^E] + [k^E])[U_2^E] = [f_d^E] + [\hat{f}^E] - [B_2\sigma_1^E] \quad (8)$$

$[K^E]$ and $[k^E]$ denote rigidity matrices (constant along iterations), arising from material and search direction respectively, $[f^E]$ is a load due to $\hat{\underline{F}}_2^E + k\hat{\underline{W}}_2^E$, and B_2 is the operator giving the generalised forces that balance a given stress field on mesh Ω_2^E . We can notice that the problem to solve is global on the substructure and is elasticity-like in nature.

- on the large scale 1,

$$\begin{aligned} \forall \underline{U}_1^* \in \mathcal{U}_1, \\ \int_{\Omega} \text{Tr}[\varepsilon(\underline{U}_1)\mathbf{K}\varepsilon(\underline{U}_1^*)]d\Omega = \int_{\Omega} \underline{f}_d \cdot \underline{U}_1^*d\Omega - \sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\varepsilon(\underline{U}_2)\mathbf{K}\varepsilon(\underline{U}_1^*)]d\Omega \quad (9) \end{aligned}$$

with $\underline{U}_1^* = \mathbf{P}\underline{U}$, the last term is: $-\sum_{E \in \mathbf{E}} \int_{\Omega^E} \text{Tr}[\mathbf{R}^E \varepsilon(\underline{U}_2)\mathbf{K}\varepsilon(\underline{U}_1^*)]d\Omega$. As the stress field $\bar{\sigma}$ must balance $-\bar{\sigma}_2 = -\sum_{E \in \mathbf{E}} \mathbf{R}^E \mathbf{K}\varepsilon(\underline{U}_2^E)$, the scales are not separated. The discretised displacement-oriented formulation of the problem (9), with $\bar{\sigma}_d$ arising from external loads, is:

$$[K_1][\bar{U}] = [B_1\bar{\sigma}_d] - [B_1\bar{\sigma}_2] \quad (10)$$

The solution is searched successively from the two levels within each LATIN iteration on the substructured fine scale, in a fixed point method, as described in [4]. The linear stage is then performed on both scales, while local stage is still the same as for the one-level approach but only deals with fine scale quantities: $(\hat{\underline{W}}_2^E; \hat{\underline{F}}_2^E)$. Table 1 describes the algorithm. It has been implemented in the industrial-type code CASTEM 2000 developed at the CEA in Saclay, [13].

For the previous example, the convergence rate has been illustrated in Figure 3. The quasi-independence of the convergence rate with respect to the number of substructures shows the numerical scalability of the 2-level LATIN method. One can notice that for this example, the new optimum value for the search direction is now related to the interface length (L_0 has then be chosen as equal to 0.25 times the length of one substructure). It is no longer characterised by the behaviour of the whole structure [2], but becomes a substructuring characteristic, see [4].

4 Comparisons

Several domain decomposition algorithms currently use a global mechanism, like the FETI method, [6]. It produces at each iteration a solution that satisfies equilibrium through the interfaces, and that has to accommodate global equilibrium on each subdomain. This leads to the resolution of a global problem on all subdomains to find their rigid body movement, related to the large-scale problem.

The proposed example at this time is a tridimensional beam with a parabolic bending loading at one extremity. 32 substructures and a mesh with 20-node cubic elements are considered for this problem (one substructure has 3 675 d.o.f. and requires 12.8 Mb of storage for the factorised rigidity, while the direct problem has 95 043 d.o.f. and requires 1 252 Mb). For the large scale, the influence of the discretisation with 8-node cubic elements is studied, as also shown in Figure 4.

Table 1: 2-level algorithm

Large scale — 1 processor	Fine scale — n processors
Initialisation initialisation of $[B_1\bar{\sigma}_2] = 0$ receiving $[B_1\bar{\sigma}_d]_{ \Omega^E}$ ← assembling the contributions factorisation of $[K_1]$ forward-backward on (10) sending \bar{U} →	Initialisation computing contributions $[B_1\bar{\sigma}_d]_{ \Omega^E}$ ← sending $[B_1\bar{\sigma}_d]_{ \Omega^E}$ ← initialisation of $\hat{\mathbf{s}} = 0$ factorisation of $[K^E] + [k^E]$ → receiving \bar{U} → computing coupling term $[B_2\sigma_1^E]$ forward-backward on (8)
Loop over iterations receiving $[B_1\bar{\sigma}_2]_{ \Omega^E}$ ← assembling the contributions forward-backward on (10) sending \bar{U} →	Loop over iterations computing coupling term $[B_1\bar{\sigma}_2]_{ \Omega^E}$ ← sending $[B_1\bar{\sigma}_2]_{ \Omega^E}$ ← local stage , convergence check ↔ → receiving \bar{U} → computing coupling term $[B_2\sigma_1^E]$ forward-backward on (8)

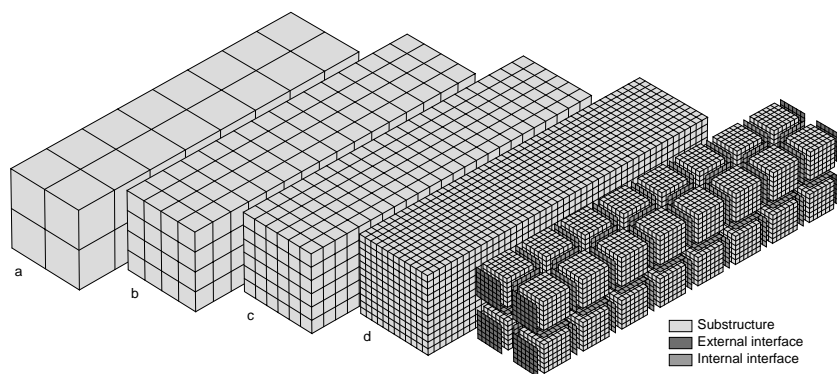


Figure 4: Meshes of the large-scale problem (a,b,c,d) and of the substructured problem (assembly of substructures and interfaces)

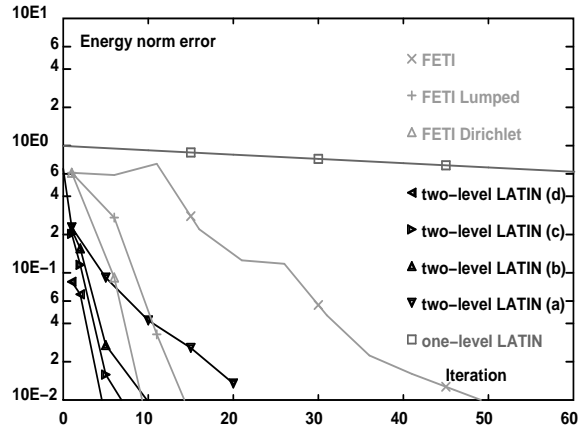


Figure 5: Comparison of methods – error versus iterations

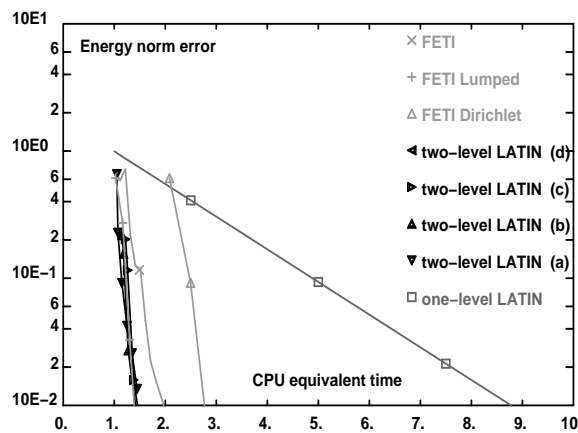


Figure 6: Comparison of costs

Figure 5 shows error e_n versus iterations, for the FETI method without preconditioning, then with lumped preconditioning, and finally with optimal Dirichlet preconditioning. These three computations have been performed by F.-X. Roux with the PARAGON machine at ONERA-Châtillon, France. The previous single-level LATIN algorithm as well as the 2-level extension for the different large-scale discretisations are also reported. These computations have been performed on the CRAY-T3D computer at IDRIS in Orsay. Both of these parallel computers have been used with 32 processors. Since time comparisons between two approaches depends on the processor, the intercommunication network, the compilers, disk usage, etc., we retain only the major tendencies by weighting the previous results; after analysing the costly parts of simulations, we identified CPU costs of initialisations for the FETI approach and the LATIN single-level to 1, in terms of CPU equivalent time (accumulated on the 32 processors). Afterwards, the FETI iteration and the 2-level LATIN iteration for the case (a) are identified in terms of cost. Figure 6 then shows the evolution of error versus this CPU equivalent time.

The cost for a direct finite element approach is 18 in terms of CPU equivalent time. When using the multi-frontal scheme, [3], [5], the condensed Schur complement problem has 19875 d.o.f. and requires 329 Mb of storage. The costs are 3 for local condensations and forward-backward substitutions (which can be performed concurrently) and 2.6 for the resolution of the condensed problem (sequentially). Total cost of the analysis is then 5.6 in CPU equivalent time. The cost of a local condensation is higher than a simple factorisation due to the higher fill-in of the local rigidity matrix (in order to treat the boundary d.o.f. at the end).

One can initially note that when increasing the large-scale problem size of the 2-level LATIN algorithm, the error indicator starts out lower at the first iteration because the large-scale first solution is used to initiate the algorithm. Another effect is the increase in the convergence rate (Figure 5), but since iteration costs are also increasing, the two effects cancel each other for the proposed example, (Figure 6).

5 Conclusions

The originality in both the use of the large time increment method and a substructuring approach is the major role played by interfaces, which are considered as structures in their own right. This leads to a “pure parallel” algorithm that can be improved when using a 2-level scheme. The consequence is the generation of a global problem to solve on the whole structure at each iteration. The resulting algorithm is then numerically scalable.

The ultimate goal is the extension to non-linear structural analysis with a large number of d.o.f. One approach which is currently under development deals with a 2-level version more suited to homogenisation techniques, completely merged with non-incremental LATIN methods.

Acknowledgements The authors wish to thank F.-X. Roux from ONERA-Châtillon, for having performed the computations with FETI approaches on the PARAGON, as well as IDRIS at Orsay, for accessing the CRAY-T3D.

References

- [1] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103–134, 1986.
- [2] L. Champaney, J.-Y. Cognard, D. Dureisseix, and P. Ladevèze. Large scale applications on parallel computers of a mixed domain decomposition method. *Computational Mechanics*, (19):253–263, 1997.
- [3] I. S. Duff. Parallel implementation of multifrontal schemes. *Parallel Computing*, 3:192–204, 1986.
- [4] D. Dureisseix and P. Ladevèze. Parallel and multi-level strategies for structural analysis. In J.-A. Désidéri, editor, *Proceedings of the Second European Conference on Numerical Methods in Engineering*, pages 599–604. Wiley, September 1996.
- [5] Y. Escaig, G. Touzot, and M. Vayssade. Parallelization of a multilevel domain decomposition method. *Computing Systems in Engineering*, 5(3):253–263, 1994.
- [6] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2. North-Holland, June 1994.
- [7] R. Glowinski and P. Le Tallec. Augmented lagrangian interpretation of the nonoverlapping Schwarz alternating method. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–231, Philadelphia, 1990. SIAM.
- [8] J. Ladevèze. Algorithmes adaptés aux calculs vectoriel et parallèle pour des méthodes de décomposition de domaines. In J. P. Grellier and G. M. Campel, editors, *Actes du 3ème colloque Tendances Actuelles en Calcul de Structures*, pages 893–907, Bastia, November 1985. Pluralis.
- [9] P. Ladevèze. *Mécanique non-linéaire des structures — Nouvelle approche et méthodes de calcul non incrémentales*. Hermès, Paris, 1996.
- [10] P. Ladevèze and P. Lorong. A large time increment approach with domain decomposition technique for mechanical non linear problems. In R. Glowinski, editor, *Comput. Meths. Appl. Sc. Engng.*, pages 569–578, New York, 1992. INRIA, Nova Science.
- [11] P. Le Tallec. Domain decomposition methods in computational mechanics. In *Computational Mechanics Advances*, volume 1. North-Holland, 1994.
- [12] P.-L. Lions. On the Schwarz alternating method III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 202–223, Philadelphia, 1990. SIAM.
- [13] P. Verpeaux, T. Charras, and A. Millard. CASTEM 2000 : une approche moderne du calcul des structures. In J.-M. Fouet, P. Ladevèze, and R. Ohayon, editors, *Calcul des Structures et Intelligence Artificielle*, volume 2, pages 261–271. Pluralis, 1988.