
A methodology for large scale finite element models, including multi-physic, multi-domain and multi-timestep aspects

Laurent Menanteau* — Olivier Pantalé** — Serge Caperaa**

* P.E.A.R.L. - Alstom Transport, Rue du Dr Guinier, BP 4, F-65600, Séméac

** L.G.P. - E.N.I. de Tarbes, 47, av. d'Azereix, BP 1629, F-65016 Tarbes cedex

ABSTRACT. This work concerns the development of a virtual prototyping tool dedicated to electro-thermo-mechanical simulation of power converters. The FEM code, written using an object-oriented language, includes a dual Schur Domain Decomposition Method. The solving of problems including floating subdomains can be performed in steady-state cases, whereas one can couple multi-timestep implicit and explicit integration schemes in transient analysis. The last part of this work is about the study of an industrial benchmark concerning the power converters used in railway transport: the electro-thermal simulation of a switch in transient analysis. This example allows to compare different strategies of tearing into subdomains and the use of different timesteps on the same structure.

RÉSUMÉ. Ce travail concerne le développement d'une plateforme de prototypage virtuel dédiée à la simulation numérique électro-thermo-mécanique des convertisseurs de puissance. Le code éléments finis, développé selon un formalisme orienté objets, est basé sur une décomposition duale de type Schur. La résolution des sous-problèmes incluant les domaines flottants peut être réalisée en régime permanent. Il est également possible de coupler les approches implicite multi-temps et explicite pour une analyse transitoire. La dernière section de ce travail présente un cas test industriel concernant les convertisseurs de puissance utilisés dans le transport ferroviaire : la simulation électrothermique transitoire d'un switch. Cet exemple permet de comparer les différentes stratégies et l'utilisation de plusieurs incréments de temps dans une structure.

KEYWORDS: domain decomposition method, finite element, object-oriented programming, parallelization, multi-time stepping.

MOTS-CLÉS : méthode de décomposition de domaine, éléments finis, programmation orientée objets, parallélisation, intégration multiples de temps.

1. Introduction

Industrial products are made up of many parts, and they are submitted to multi-physic and multi-time phenomena. In particular, the prototypes that are built and designed in the field of Power Electronics focus on power integration components, with the goal of weight, volume and cost reductions. Because of their integration, these products are difficult to instrument whereas the demand for reliability increases; therefore there is a real need in more realistic models for their virtual prototyping.

One solution in order to build such models is to link together several semi-analytic models using an integration platform such as VTB, Femlab, Matlab/Simulink... The main inconvenient of this approach relies in the difficulties to build realistic analytic models for non-linear and complex behaviors. Therefore numerical models are usually preferred. A multi-code approach implies the use of several numerical codes and data exchanges between them. So, this platform enhances some disadvantages:

- a relatively high cost due to the number of different softwares involved in the process;
- the need of developing specific data exchange softwares in order to establish the communications between the solvers;
- the problems encountered when new code versions are used.

However multi-code couplings have been developed using for example the CORBA platform (see for example (Pérez *et al.*, 2003)). Therefore, in the approach developed here, we have choosen to build a unique three-dimensional software, with the following goals:

- ability to treat large scale problems, and, for transient analysis, ability to use different timesteps in different parts of the numerical model;
- integration into a single interface, high evolutivity and code maintenance;
- high performance in order to obtain computing times compatible with industrial developments.

Domain Decomposition Method (DDM) has been retained in order to give an answer to the first point; for the second point, the Object-Oriented Programming is used; finally, a procedure of parallelization is introduced to improve the performance of the code.

DDM divides the domain into several sub-domains and restricts the resolution of the finite element problem to the interface between these sub-domains; moreover, these methods allow to perform non-linear analysis. Two main kinds of DDM can be considered, with and without partial overlapping.

The first one derives from the alternative Schwarz method (1869). The solution of the global problem is obtained by alternating partial resolutions on the different sub-domains. The partial solutions obtained on the neighbor sub-domains are reported

as boundary conditions on the studied sub-domain; the global solution is obtained following this iterative process.

In the second kind of DDM (Kron, 1963), the whole structure is divided into a set of adjacent sub-domains linked by an interface (see Figure 1). The first step consists in building the problem related to each sub-domain; after construction and resolution of the interfacial problem, one solves the problem on each sub-domain. This last method, allowing the introduction of contact laws between sub-domains, has been retained in our work.

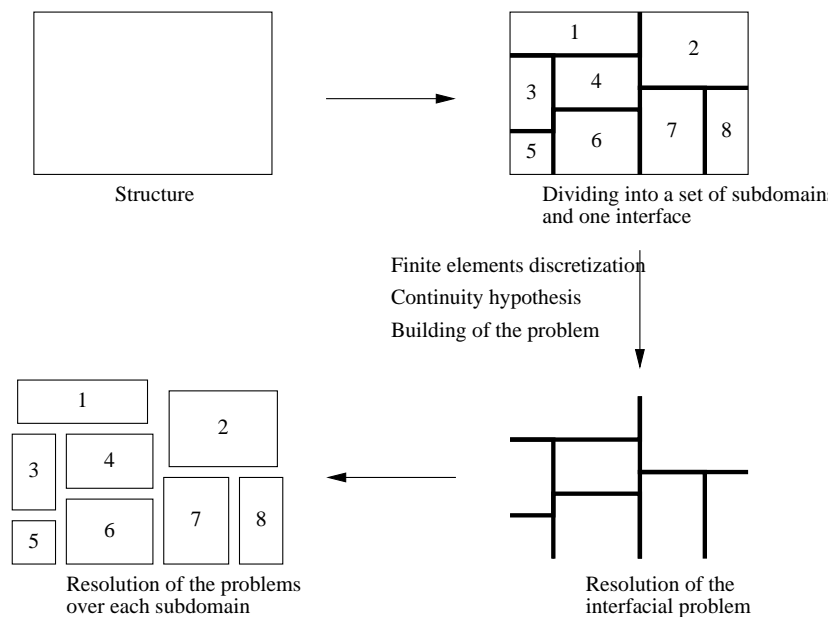


Figure 1. General principle of the DDM without overlapping

First introduced for electrical modelization in the 1960's by Kron (1963), the DDM have been adapted to structural mechanics in the 1980's with the democratization and the increase of computing power. Farhat, Roux and Rixen have developed the resolution of the interfacial problem in steady state analysis leading to the so-called Finite Element Tearing and Interconnecting method (FETI) (Roux, 1990; Farhat, 1991; Farhat *et al.*, 1994; Rixen, 1998). Now, the developments focus on the transient analysis: multi-timestepping in non-linear analysis in mechanics (Combesure *et al.*, 2002; Gravouil, 2000), and in thermics (Smolinski *et al.*, 2000). Moreover, works have been published concerning the generalization of the different formulations of DDM (Papadrakakis, 1997; Fragakis *et al.*, 2002; Fragakis *et al.*, 2003). The FETI has been extended later to a second generation adapted to structural elements and ill-conditioned problems. More recently, these methods have been generalized by the FETI-DP approach, derived as an alternative to the second

generation methods (Farhat *et al.*, 2001). The multi-physic aspect has been treated in this paper using a weak coupling approach; for a full coupling one can use a staggered algorithm such as the the one proposed by B.A. Schreffer for the soil consolidation (Lewis *et al.*, 1999).

Concerning the field of power converters, (Hoppe *et al.*, 2003; Hoppe, 2004; Chow *et al.*, 2001; 2003) have investigated multi-physics (electrics, thermics, mechanics) and multi-domain modelizations.

2. Formulation of domain decomposition methods in steady-state analysis

A structure is subdivided into s sub-domains, that is s sets of elements. The set of nodes belonging to the different sub-domains and situated on the partition lines is the “interface” (see Figure 2). Most of the sub-domains are not concerned with Dirichlet type boundary conditions; they are pointed out as “floating” sub-domains. For example, in Figure 2, all the sub-domains except sub-domain 5 are “floating”.

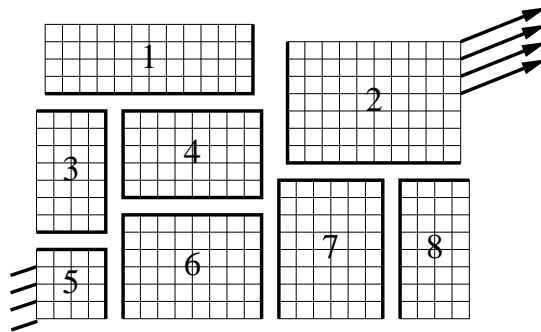


Figure 2. Partition and interface between the sub-domains

For our developments, we have made the choice to use a dual domain decomposition: the equilibrium of dual quantities is imposed at the interface, whereas the continuity of the primal quantities is verified after the calculation.

2.1. Global formulation

Starting from a classical variational formulation, the whole finite element problem can be formulated as a set of all elementary problems on each sub-domain: Equations [1] (a), associated with the Equation [1] (b) enforcing the continuity of primal quantities on the interface.

$$\begin{cases} K^{(j)} q^{(j)} = g^{(j)} + g_{int}^{(j)} & , j \in \{1, \dots, s\} & \text{(a)} \\ \sum_{j=1}^s B^{(j)} q^{(j)} = 0 & & \text{(b)} \end{cases} \quad [1]$$

where $K^{(j)}$ is the matrix relative to the physical behavior of the sub-domain (material and geometry), $q^{(j)}$ is the vector of the unknown primal quantities, $g^{(j)}$ is the vector of the external actions applied on the sub-domain j and $g_{int}^{(j)}$ is the vector of the actions of the adjacent sub-domains on the sub-domain j . System [1] includes all physical boundary conditions. The localization matrices $B^{(j)}$ select the degrees of freedom of the sub-domain j owning to the interface ($B_i^{(j)} = 1$ if node i of sub-domain j belongs to the interface and zero otherwise). The interaction vector $g_{int}^{(j)}$ for each subdomain is expressed by the projection of a unique unknown vector λ with the localization matrix $B^{(j)T}$:

$$g_{int}^{(j)} = B^{(j)T} \lambda \quad [2]$$

where λ is the general vector of the interfacial interactions in the structure. The problem related to the global structure is written as follow:

$$\begin{bmatrix} K^{(1)} & \dots & 0 & -B^{(1)T} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & K^{(s)} & -B^{(s)T} \\ -B^{(1)} & \dots & -B^{(s)} & 0 \end{bmatrix} \begin{bmatrix} q^{(1)} \\ \vdots \\ q^{(s)} \\ \lambda \end{bmatrix} = \begin{bmatrix} g^{(1)} \\ \vdots \\ g^{(s)} \\ 0 \end{bmatrix} \quad [3]$$

or in a block form:

$$\begin{bmatrix} \mathbf{K} & -\mathbf{B}^T \\ -\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix} \quad [4]$$

2.2. Formulation of the interfacial problem

By inverting the relation [1] (a), and taking into account the Equation [2], the unknown vector $q^{(j)}$ can be expressed as:

$$q^{(j)} = K^{(j)-1} \left(g^{(j)} + B^{(j)T} \lambda \right) \quad [5]$$

In case of floating sub-domains, the matrix $K^{(j)}$ is singular and the resolution of the problem is achieved according to the following system:

$$\begin{cases} q^{(j)} = K^{(j)+} \left(g^{(j)} + B^{(j)T} \lambda \right) + R^{(j)} \gamma^{(j)} \\ R^{(j)T} \left(g^{(j)} + B^{(j)T} \lambda \right) = 0 \end{cases} \quad [6]$$

where $K^{(j)+}$ is the generalized inverse matrix of $K^{(j)}$, $R^{(j)}$ (kernel of $K^{(j)}$) is a basis of the null-space of $K^{(j)}$, $\gamma^{(j)}$ contains the unknown amplitudes of $R^{(j)}$. For non floating sub-domains, $K^{(j)+}$ will also be used to represent the inverse of $K^{(j)}$ (in

this case $R^{(j)}\gamma^{(j)} = 0$). The formulation of the interfacial problem is obtained by replacing the system [6] into Equation [1] (b). This leads to the system [7]:

$$\begin{cases} \sum_{j=1}^s B^{(j)} \left(K^{(j)+} \left(g^{(j)} + B^{(j)T} \lambda \right) + R^{(j)} \gamma^{(j)} \right) = 0 \\ (R^{(j)} \gamma^{(j)})^T \left(g^{(j)} + B^{(j)T} \lambda \right) = 0 \end{cases} \quad [7]$$

or in a block form:

$$\begin{bmatrix} \mathbf{F}_I & \mathbf{G}_I \\ \mathbf{G}_I^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} g_\lambda \\ g_\gamma \end{bmatrix} \quad [8]$$

with:

$$\begin{cases} \mathbf{F}_I = \sum_{j=1}^s B^{(j)} K^{(j)+} B^{(j)T} \\ \mathbf{G}_I = \begin{bmatrix} B^{(1)} R^{(1)} & \dots & B^{(f)} R^{(f)} \end{bmatrix} \\ \gamma = \begin{bmatrix} \gamma^{(1)T} & \dots & \gamma^{(f)T} \end{bmatrix}^T \\ g_\lambda = - \sum_{j=1}^s B^{(j)} K^{(j)+} g^{(j)} \\ g_\gamma = - \begin{bmatrix} g^{(1)T} R^{(1)} & \dots & g^{(f)T} R^{(f)} \end{bmatrix}^T \end{cases} \quad [9]$$

where f represents the number of floating sub-domains of the structure. \mathbf{F}_I is commonly called the “dual Schur matrix”.

3. Extension of DDM to multi-timestep transient analysis

We have made the choice of using a dual DDM formulation in transient analysis. It is necessary to consider separately first order and second order problems. The following formulations use classical notations of thermics for first order problems and of structural dynamics for second order problems.

3.1. First order transient problems

As in the steady state case, the finite element problem related to the whole structure is set as the sum of the problems related to the s sub-domains linked together by the use of a continuity equation:

$$\begin{cases} C_n^{(j)} \dot{T}_n^{(j)} + K_n^{(j)} T_n^{(j)} = f_n^{(j)} + B^{(j)T} \lambda_n, \quad j \in \{1, \dots, s\} & \text{(a)} \\ \sum_{j=1}^s B^{(j)} w_n^{(j)} = 0 & \text{(b)} \end{cases} \quad [10]$$

where C_n is the thermal capacity matrix, K_n is the thermal conductivity matrix, f_n is the vector of the external calorific flux, T_n is the temperature and w_n represents the continuous quantity at the interface between the sub-domains; *i.e.* $w_n = T_n$ for a continuity in temperature and $w_n = \dot{T}_n$ for a continuity in flux. In the present work,

the Euler time integration scheme is adopted to integrate Equation [10] through time between increments $n - 1$ and n :

$$T_n^{(j)} = T_{n-1}^{(j)} + (1 - \alpha^{(j)})\Delta t^{(j)}\dot{T}_{n-1}^{(j)} + \alpha^{(j)}\Delta t^{(j)}\dot{T}_n^{(j)}, \alpha^{(j)} \in [0, 1] \quad [11]$$

which can be splitted into the sum of a predictor $^p()$ and a corrector $^c()$:

$$\begin{cases} T_n^{(j)} = ^pT_n^{(j)} + ^cT_n^{(j)} \\ ^pT_n^{(j)} = T_{n-1}^{(j)} + (1 - \alpha^{(j)})\Delta t^{(j)}\dot{T}_{n-1}^{(j)} \\ ^cT_n^{(j)} = \alpha^{(j)}\Delta t^{(j)}\dot{T}_n^{(j)} \end{cases} \quad [12]$$

The scalar value $\alpha^{(j)}$ allows to define an explicit integration scheme if $\alpha^{(j)} = 0$, an implicit integration scheme if $\alpha^{(j)} = 1$ or a Crank-Nicholson one if $\alpha^{(j)} = \frac{1}{2}$. We can choose between two kind of continuity at the interface: $T^{(j)}$ or $\dot{T}^{(j)}$. Regarding these two possible continuities, $w_n^{(j)}$ can be written in a more general form:

$$w_n^{(j)} = ^pw_n^{(j)} + \mu^{(j)}\dot{T}_n^{(j)} \quad [13]$$

where:

$$\begin{cases} \text{if } w_n^{(j)} = \dot{T}_n^{(j)}, & ^pw_n^{(j)} = 0 & \text{and } \mu^{(j)} = 1 \\ \text{if } w_n^{(j)} = T_n^{(j)}, & ^pw_n^{(j)} = ^pT_n^{(j)} & \text{and } \mu^{(j)} = \alpha^{(j)}\Delta t^{(j)} \end{cases} \quad [14]$$

The general first order problem can be rewritten in a block form as:

$$\begin{bmatrix} \mu\tilde{\mathbf{C}} & -\mu\mathbf{B}^T \\ -\mu\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{T}}_n \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_n \\ \sum_{j=1}^s B^{(j)}w_n^{(j)} \end{bmatrix} \quad [15]$$

where:

$$\tilde{C}_n^{(j)} = C_n^{(j)} + \alpha^{(j)}\Delta t^{(j)}K_n^{(j)} \quad \text{and} \quad \tilde{f}_n^{(j)} = \mu^{(j)}(f_n^{(j)} - K_n^{(j)}w_n^{(j)}) \quad [16]$$

Then the general problem is splitted into in a “free” problem and in a “linked” problem (Combescure *et al.*, 2002; Gravouil, 2000): $\dot{T}_n^{(j)}$ is considered as the sum of a “free” vector $\dot{T}_{n_{free}}^{(j)}$ solution of the global problem with no interfacial interactions between the sub-domains, and a “link” vector $\dot{T}_{n_{link}}^{(j)}$ solution of the global problem just considering the interfacial interactions and excluding all the other actions. These two problems are written:

$$\begin{cases} \begin{bmatrix} \mu\tilde{\mathbf{C}}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{T}}_{n_{free}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_n \\ \mathbf{0} \end{bmatrix} \\ \text{and} \\ \begin{bmatrix} \mu\tilde{\mathbf{C}}_n & -\mu\mathbf{B}^T \\ -\mu\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{T}}_{n_{link}} \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \sum_{j=1}^s B^{(j)}w_{n_{free}}^{(j)} \end{bmatrix} \end{cases} \quad [17]$$

with $w_{n_{free}}^{(j)} = p w_n^{(j)} + \mu^{(j)} \dot{T}_{n_{free}}^{(j)}$ only depending upon free quantities. When all the sub-domains share the same timestep $\Delta t^{(j)}$, the interfacial problem is built for each timestep by using the same procedure as the one used for the steady state analysis. As there is no floating sub-domains in a transient computation, the expression of the interfacial problem can be simplified as:

$$H_n \lambda_n = f_{\lambda_n} \tag{18}$$

where:

$$H_n = - \sum_{j=1}^s \mu^{(j)} B^{(j)} \tilde{C}_n^{(j)-1} B^{(j)T} \quad ; \quad f_{\lambda_n} = \sum_{j=1}^s B^{(j)} w_{n_{free}}^{(j)} \tag{19}$$

When the free problem is solved, we are able to solve the interfacial problem; λ_n is then reintroduced in the expressions of the link problem related to each sub-domain and allows to solve them. The expressions of $\dot{T}_n^{(j)}$ can be computed and using the integration scheme [12], we can obtain the temperatures $T_n^{(j)}$.

When different timesteps are used in different sub-domains, the interfacial problem is computed for each minimal timestep of the structure. So it requires the interpolation of the quantities used in continuity and corresponding to the sub-domains which have timesteps different from the minimal timestep. In this work, it is supposed that all the timesteps are multiples of the minimal one through the value of $k^{(j)}$ (see Figure 3 presenting the time discretization for 3 domains with $k = 1, 2$ and 6 respectively).

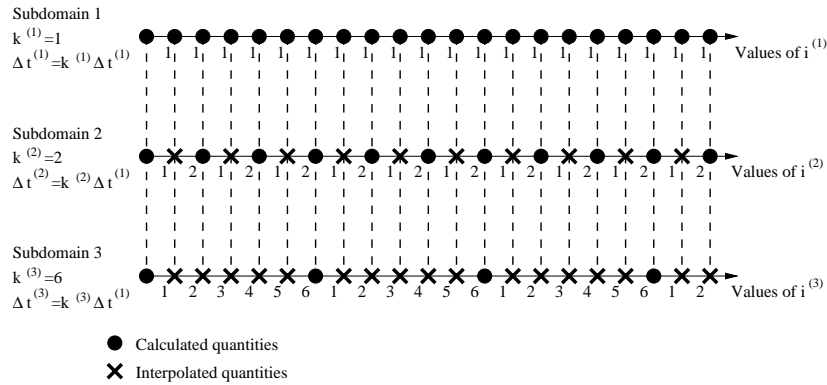


Figure 3. Example of time discretization for three sub-domains

The interfacial problem is modified:

$$H_i \lambda_i = f_{\lambda_i} \tag{20}$$

where:

$$H_i = \sum_{j=1}^s \mu^{(j)} B^{(j)} \tilde{C}_i^{(j)-1} B^{(j)T} \quad \text{and} \quad f_{\lambda_i} = - \sum_{j=1}^s B^{(j)} \hat{w}_{i_{free}}^{(j)} \tag{21}$$

In this equation $\hat{w}_i^{(j)}$ are interpolated quantities defined by:

$$\hat{w}_{i_{free}}^{(j)} = \left(1 - \frac{i}{k^{(j)}}\right)w_{0_{free}}^{(j)} + \frac{i}{k^{(j)}}w_{k_{free}}^{(j)} \quad \text{and} \quad \Delta t^{(j)} = k^{(j)} \min(\Delta t^{(j)}) \quad [22]$$

depending only on $()_{free}$ quantities. The resolution of the problem for each minimal timestep is done in the same way as for one unique timestep on the structure.

3.2. Second order transient problems

As previously done concerning first order transient problems, the second order transient problem is formulated using a dual DDM:

$$\begin{cases} M_n^{(j)}\ddot{u}_n^{(j)} + C_n^{(j)}\dot{u}_n^{(j)} + K_n^{(j)}u_n^{(j)} = f_n^{(j)} + B^{(j)T}\lambda_n, j \in \{1, \dots, s\} & \text{(a)} \\ \sum_{j=1}^s B^{(j)}w_n^{(j)} = 0 & \text{(b)} \end{cases} \quad [23]$$

where M is the mass matrix, C is the capacitance matrix, K is the stiffness matrix, u is the displacement vector, f is the external force vector and w again represents the quantity chosen to be continuous at the interface between the sub-domains. Here the widely used Newmark integration scheme is adopted for time integration:

$$\begin{cases} \dot{u}_n^{(j)} = \underbrace{\dot{u}_{n-1}^{(j)} + (1 - \gamma^{(j)})\Delta t^{(j)}\ddot{u}_{n-1}^{(j)}}_{p\dot{u}_n^{(j)}} + \gamma^{(j)}\Delta t^{(j)}\ddot{u}_n^{(j)} \\ u_n^{(j)} = \underbrace{u_{n-1}^{(j)} + \Delta t^{(j)}\dot{u}_{n-1}^{(j)} + \left(\frac{1}{2} - \beta^{(j)}\right)(\Delta t^{(j)})^2\ddot{u}_{n-1}^{(j)}}_{p u_n^{(j)}} + \beta^{(j)}(\Delta t^{(j)})^2\ddot{u}_n^{(j)} \end{cases} \quad [24]$$

and, depending on the continuity set at the interface, we have:

$$\begin{cases} \text{if } w_n^{(j)} = \ddot{u}_n^{(j)}, \quad p w_n^{(j)} = 0 & \text{and } \mu^{(j)} = 1 \\ \text{if } w_n^{(j)} = \dot{u}_n^{(j)}, \quad p w_n^{(j)} = p\dot{u}_n^{(j)} & \text{and } \mu^{(j)} = \gamma^{(j)}\Delta t^{(j)} \\ \text{if } w_n^{(j)} = u_n^{(j)}, \quad p w_n^{(j)} = p u_n^{(j)} & \text{and } \mu^{(j)} = \beta^{(j)}(\Delta t^{(j)})^2 \end{cases} \quad [25]$$

Implicit, semi-implicit or explicit integration schemes can be selected through the $\beta^{(j)}$ and $\gamma^{(j)}$ parameters. With the use of the integration scheme and the Lagrange multipliers λ_n corresponding to the equilibrium of the interface, the dual transient problem can be written in a block form:

$$\begin{bmatrix} \mu\tilde{M}_n & -\mu\mathbf{B}^T \\ -\mu\mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_n \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{f}}_n \\ \sum_{j=1}^s B^{(j)}p w_n^{(j)} \end{bmatrix} \quad [26]$$

with:

$$\begin{aligned}\tilde{M}_n^{(j)} &= M_n^{(j)} + \gamma^{(j)} \Delta t^{(j)} C_n^{(j)} + \beta^{(j)} (\Delta t^{(j)})^2 K_n^{(j)} \\ \tilde{f}_n^{(j)} &= \mu^{(j)} \left(f_n^{(j)} - C_n^{(j)p} \dot{u}_n^{(j)} - K_n^{(j)p} u_n^{(j)} \right)\end{aligned}\quad [27]$$

The interfacial problem is:

$$H_n \lambda_n = f_{\lambda_n} \quad [28]$$

with the following notations:

$$\begin{aligned}H_n &= \sum_{j=1}^s -\mu^{(j)} B^{(j)} \tilde{M}_n^{(j)-1} B^{(j)T} \\ f_{\lambda_n} &= \sum_{j=1}^s B^{(j)p} w_{n_{free}}^{(j)} + \sum_{j=1}^s B^{(j)T} \tilde{M}_n^{(j)-1} \tilde{f}_n^{(j)}\end{aligned}\quad [29]$$

The problem is then solved by using a direct method. One usually use an iterative method known to enhance quite a fast convergence, but in our case this has not been used for simplicity.

3.3. Discussion about the choice of the continuous quantities at the interface

This section deals with the stability and precision of the integration schemes when they are coupled with the DDM. For this purpose, the so called interfacial energy E_{inter} is introduced, and:

- if $E_{inter} = 0$, stability and precision are identical to those obtained without the DDM;
- if $E_{inter} > 0$, the multi-timestep DDM can introduce numerical instabilities;
- if $E_{inter} < 0$, the multi-timestep DDM can introduce numerical dissipation.

(Combescure *et al.*, 2002) have discussed this point for the second-order problems. Their approach is based on the use of the energy methods developed by (Hughes *et al.*, 1978a; 1978b). We follow the same approach here after and give details within the case of the first order problems.

3.3.1. Single-time transient analysis

For a given quantity A defined respectively by A_n and A_{n+1} at the times n and $n + 1$, one define the average $\langle A \rangle$ and the difference $[A]$:

$$\begin{cases} \langle A_n \rangle = (A_{n+1} + A_n)/2 \\ [A_n] = (A_{n+1} - A_n) \end{cases} \quad [30]$$

With the remark that: $\langle A_n \rangle [A_n] = [\frac{1}{2}(A_n)^2]$, starting from the energy formulation:

$$[\dot{T}_n^e]^T \left(C_n^e [\dot{T}_n^e] + K_n^e [T_n^e] - [f_n^e] \right) = 0 \quad [31]$$

where, relatively to the Euler's integration scheme, we can write:

$$[T_n^e] = \Delta t \langle \dot{T}_n^e \rangle + (\alpha - \frac{1}{2})\Delta t [\dot{T}_n^e] \quad [32]$$

One obtains:

$$[E(\dot{T}_n^e)] = -D([\dot{T}_n^e]) + [E_{ext}(\dot{T}_n^e)] \quad [33]$$

where $[E(\dot{T}_n^e)]$ represents the heat diffused during the interval of time. The term $-D([\dot{T}_n^e]) = -\frac{1}{\Delta t} [\dot{T}_n^e]^T (C + (\alpha - \frac{1}{2})\Delta t K) [\dot{T}_n^e]$ represents the numerical damping. $[E_{ext}(\dot{T}_n^e)] = \frac{1}{\Delta t} [\dot{T}_n^e]^T [f_n^e]$ is the energetic term due to the balance of the external fluxes. (Hughes *et al.*, 1978a; 1978b) have shown that this term have no influence on the stability of the numerical scheme, so this one will be no more taken into account. For a multi-domain study, Equation [33] becomes:

$$\sum_{j=1}^s [E(\dot{T}_n^{(j)})] = -\sum_{j=1}^s D([\dot{T}_n^{(j)}]) + \sum_{j=1}^s [E_{ext}(\dot{T}_n^{(j)})] + E_{inter} \quad [34]$$

with:

$$E_{inter} = \frac{1}{\Delta t} \sum_{j=1}^s [\dot{T}_n^{(j)}]^T B^{(j)T} [\lambda_n] = \frac{1}{\Delta t} [\lambda_n]^T \sum_{j=1}^s B^{(j)} [\dot{T}_n^{(j)}] \quad [35]$$

Thus the stability is depending on the sign of E_{inter} . As presented earlier, for a first order problem, the choice can be achieved between two kind of continuities: continuity of fluxes or continuity of temperatures. Depending on this choice, E_{inter} may vary.

3.3.1.1. Continuity of fluxes

The continuity is given by the equation:

$$\sum_{j=1}^s B^{(j)} \dot{T}_n^{(j)} = 0 \quad [36]$$

Therefore we easily find that in this case, $\sum_{j=1}^s B^{(j)} [\dot{T}_n^{(j)}] = 0$ and so $E_{inter} = 0$. Hence, the stability of the Euler's integration scheme is only affected by the computing errors introduced by the continuity conditions at the interface.

3.3.1.2. Continuity of temperatures

Starting from the expression obtained with the Euler's integration scheme [32]:

$$\sum_{j=1}^s B^{(j)} [T_n^{(j)}] = \Delta t \sum_{j=1}^s B^{(j)} \langle \dot{T}_n^{(j)} \rangle + (\alpha_0 - \frac{1}{2})\Delta t \sum_{j=1}^s B^{(j)} [\dot{T}_n^{(j)}] \quad [37]$$

in the case of a Crank-Nicholson scheme ($\alpha_0 = \frac{1}{2}$), we can write:

$$\sum_{j=1}^s B^{(j)} [T_n^{(j)}] = \Delta t \sum_{j=1}^s B^{(j)} \langle \dot{T}_n^{(j)} \rangle = 0 \quad [38]$$

and, assuming continuity of the fluxes at the initial time, we deduce:

$$\sum_{j=1}^s B^{(j)} [\dot{T}_n^{(j)}] = 0 \quad [39]$$

The conditions of stability and precision are unchanged with the use of DDM in the case of a Crank-Nicholson scheme for all the sub-domains. Therefore, in this work a continuity of fluxes (\dot{T}) is retained in order to ensure the stability of the algorithm for any value of $\alpha^{(j)}$.

3.3.2. Multi-timestep transient thermics

The demonstration uses the notations [30] and new ones related to the transient analysis; for a given increment, A_0 represents the value of the quantities A at the beginning and A_k the same values at the end of the increment:

$$\begin{cases} \langle [A_k] \rangle = (A_k + A_0)/2 \\ [[A_k]] = (A_k - A_0) \end{cases} \quad [40]$$

One remarks that:

$$[[A_k]] = \sum_{n=0}^{k-1} [A_n] \quad [41]$$

In order to simplify the demonstration, we consider only two sub-domains A and B ; for one timestep of sub-domain A

$$\Delta t^A = k^A \Delta t^B = k^A \Delta t \quad [42]$$

On the interval $[t_0, t_k]$, the energy balance of each sub-domain leads to:

$$\begin{cases} \left[\left[E^A(\dot{T}_k^A) \right] \right] = -D \left(\left[\left[\dot{T}_k^A \right] \right] \right) + E_{inter}^A \\ \left[\left[E^B(\dot{T}_k^B) \right] \right] = -\sum_{n=0}^{m-1} D \left(\left[\left[\dot{T}_n^B \right] \right] \right) + E_{inter}^B \end{cases} \quad [43]$$

where

$$\begin{cases} E_{inter}^A = \frac{1}{m\Delta t} \left[\left[\dot{T}_k^A \right] \right]^T B^{AT} [[\lambda_k]] \\ E_{inter}^B = \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\left[\dot{T}_n^B \right] \right]^T B^{BT} [\lambda_n] \end{cases} \quad [44]$$

The study of the stability of these numerical schemes is equivalent to determine the sign of the total interfacial energy E_{inter} defined by:

$$E_{inter} = E_{inter}^A + E_{inter}^B \quad [45]$$

or moreover:

$$E_{inter} = \frac{1}{m\Delta t} \left[\left[\dot{T}_k^A \right] \right]^T B^{AT} [[\lambda_k]] + \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_n^B \right]^T B^{BT} [\lambda_n] \quad [46]$$

In a first time, we separate the “free” and “linked” terms:

$$\begin{aligned} E_{inter} &= \frac{1}{m\Delta t} \left[\left[\dot{T}_{(free)k}^A \right] \right]^T B^{AT} [[\lambda_k]] \\ &\quad + \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_{(free)n}^B \right]^T B^{BT} [\lambda_n] \\ &\quad + \frac{1}{m\Delta t} \left[\left[\dot{T}_{(link)k}^A \right] \right]^T B^{AT} [[\lambda_k]] \\ &\quad + \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_{(link)n}^B \right]^T B^{BT} [\lambda_n] \end{aligned} \quad [47]$$

and we set as an interpolation form

$$\left[\dot{T}_{(free)n}^A \right] = \frac{1}{m} \left[\left[\dot{T}_{(free)0}^A \right] \right] \quad [48]$$

By using the Equations [41] and [48], the expression [47] becomes:

$$\begin{aligned} E_{inter} &= \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left(B^A \left[\dot{T}_{(free)n}^A \right] + B^B \left[\dot{T}_{(free)n}^B \right] \right)^T [\lambda_n] \\ &\quad + \frac{1}{m\Delta t} \left[\left[\dot{T}_{(link)k}^A \right] \right]^T B^{AT} [[\lambda_k]] \\ &\quad + \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_{(link)n}^B \right]^T B^{BT} [\lambda_n] \end{aligned} \quad [49]$$

Assuming that the \dot{T}_n terms are continuous at the interface:

$$B^A \left[\dot{T}_{(free)n}^A \right] + B^B \left[\dot{T}_{(free)n}^B \right] = - \left(B^A \left[\dot{T}_{(link)n}^A \right] + B^B \left[\dot{T}_{(link)n}^B \right] \right) \quad [50]$$

the interfacial energy becomes

$$\begin{aligned} E_{inter} &= - \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left(B^A \left[\dot{T}_{(link)n}^A \right] + B^B \left[\dot{T}_{(link)n}^B \right] \right)^T [\lambda_n] \\ &\quad + \frac{1}{m\Delta t} \left[\left[\dot{T}_{(link)k}^A \right] \right]^T B^{AT} [[\lambda_k]] + \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_{(link)n}^B \right]^T B^{BT} [\lambda_n] \end{aligned} \quad [51]$$

or

$$E_{inter} = \frac{1}{m\Delta t} \left[\left[\dot{T}_{(link)k}^A \right] \right]^T B^{AT} [[\lambda_k]] - \sum_{n=0}^{m-1} \frac{1}{\Delta t} \left[\dot{T}_{(link)n}^A \right]^T B^{AT} [\lambda_n] \quad [52]$$

Taking into account the linked problem on the sub-domain A

$$\tilde{C}^A \dot{T}_{(link)n}^A = B^{AT} \lambda_n \quad [53]$$

we finally obtain

$$E_{inter} = \frac{1}{m\Delta t} \left(\left[\left[\dot{T}_{(link)k}^A \right] \right]^T \tilde{C}^A \left[\left[\dot{T}_{(link)k}^A \right] \right] - m \sum_{n=0}^{m-1} \left[\dot{T}_{(link)n}^A \right]^T \tilde{C}^A \left[\dot{T}_{(link)n}^A \right] \right) \quad [54]$$

It has been shown (Combescure *et al.*, 2002), for a positive definite matrix \tilde{C}^A , that the Equation [54] can be written as the sum of negative square terms:

$$E_{inter} = -\frac{1}{m\Delta t} \sum_{i=0}^{m-1} \sum_{j=i}^{m-1} \left[\dot{T}^A \right]_{(i,j)}^T \tilde{C}^A \left[\dot{T}^A \right]_{(i,j)} \quad [55]$$

with

$$\left[\dot{T}^A \right]_{(i,j)} = \left(\dot{T}_{(link)m-i}^A - \dot{T}_{(link)m-i-1}^A \right) - \left(\dot{T}_{(link)m-j-1}^A - \dot{T}_{(link)m-j-2}^A \right)$$

As a consequence:

$$E_{inter} \leq 0 \quad [56]$$

We have shown that for the multi-timestep transient thermal calculus, if we choose a continuity in $\dot{T}^{(j)}$ at the interface, the numerical integration schemes are stable. However a certain amount of energy is dissipated at the interface (Equation [56]). Moreover, if the variations $\dot{T}^{(j)}$ on the boundary of the sub-domain A are linear, so the dissipation of energy at the interface vanishes (Equation [55]).

3.3.3. Recall of the results in mono and multi-timestep transient mechanics

We recall in this section the results obtained by (Combescure *et al.*, 2002). They consider three types of continuity:

- acceleration continuity: for some coefficients $\gamma^{(j)}$ all equal in each sub-domain, the interfacial energy vanishes and the precision of the resolution depends only on the computing performances;
- speed continuity: the interfacial energy related term vanishes for any parameters $\beta^{(j)}$ and $\gamma^{(j)}$;
- displacement continuity: the interfacial energy vanishes when the initial speed continuity is assumed and if the parameters $\gamma^{(j)}$ and $\beta^{(j)}$ verify $\gamma^{(j)} = \frac{1}{2}\beta^{(j)}$.

4. Numerical implementation

4.1. Structure of the FEM code

We implemented the multi-domain solver by adding new classes libraries into the large deformation FEM code DynELA (Pantalé, 2002) developed in our laboratory.

The whole model is represented by an instance of the class *Structure* composed of several instances of the class *Physic*. Each physic has one or several sub-domains (class *Domain*) which are made of one or several meshes (class *Grid*). Finally, one or more solvers (class *Solver*) are related to each sub-domain: this data structure allows to couple iterative and direct, linear and non-linear (in transient) and explicit and implicit (in transient) computations on a same structure (see Figure 4). Currently, in this new FEM code, named *MulPhyDo* (for *Multi-Physic* and *Multi-Domain*), we implemented three solvers: an electrical, a thermal and a mechanical solver.

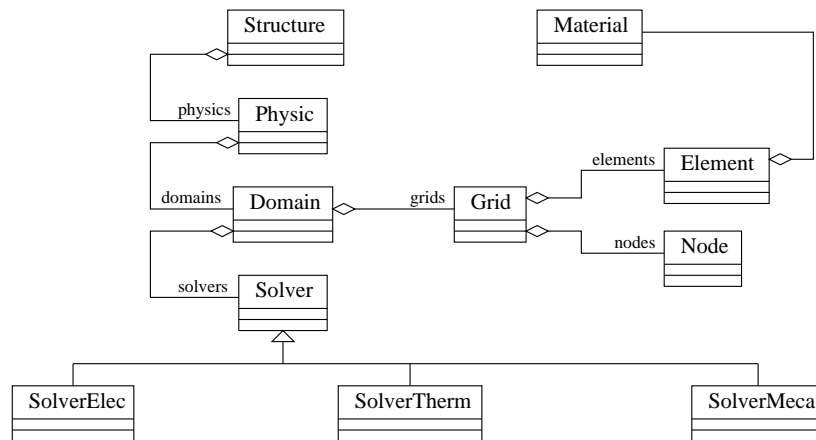


Figure 4. UML diagram of the solver

Concerning the resolution of the problem, the method *Structure::MPSolve()* is used to sequentially solve the problem on each physic; for each physic the method *Physic::ComputeConnections()* computes the interfacial nodes list; the *Physic::Solve()* method contains a loop over the sub-domains to compute the Schur matrix and the right hand side. Then the whole problem is solved using a direct method (*Physic::InterfaceSolve()*). A final loop is then used to compute the results in each sub-domain.

4.2. Parallelization

OpenMP is a recently developed programming standard (1997) offering a standard interface for softwares developed in FORTRAN and C/C++ (Chandra *et al.*, 2001; Hu *et al.*, 2000; Pantalé, 2005), on SMP (Shared Memory Processing) computers. OpenMP is an API (*Application Program Interface*) which allows the development of applications where several threads are executed in parallel: it is composed of compilation pragmas to include directly in the existing code (C/C++ or FORTRAN) and libraries of functions. The existing code must be modified to include the instructions of task sharing. The data handled by the program are common to all the processors

in the shared memory. OpenMP works using the Fork/Join principle: a task is decomposed into several elementary threads running on behalf of some processors (see Figure 5). Moreover, the program is independent of the number of processors: the number of processors necessary to compute a task is determined during the execution. The first elementary thread to be launched is arbitrarily defined as the “master”, the other ones, declared “slaves”, are created from it.

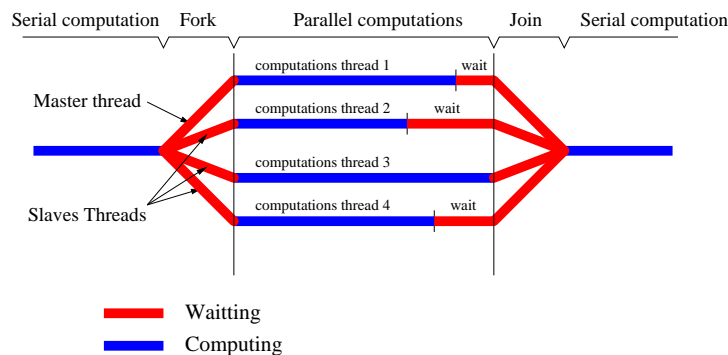


Figure 5. *Parallel computing with OpenMP*

Starting from a calculation domain partitioned into several sub-domains and one interface, the computation steps are the following:

- 1) building the Finite Elements problems related to each sub-domain;
- 2) building the interfacial problem;
- 3) solving the interfacial problem;
- 4) solving the problems on each sub-domain.

These four steps may be parallelized, but the steps 1 and 4 concerning the whole set of sub-domains are more CPU time-consuming than the two other ones (2 and 3); therefore, in a first approach, we only parallelized the two steps 1 and 4.

All the tests and validation of the parallelized version of the code have been done using a Compaq ProLiant 8000 equipped with 8 processors Intel Xeon PIII 550/2Mb and 5 Gb of shared memory. This computer works under Linux Redhat 8.0 and an Intel C++ 7.1 compiler has been used for the compilation of the parallel version of the code.

5. Validation benchmarks

In this section, some benchmark tests are used to validate the approach implemented in the FEM code MulPhyDo. Those tests are all based on the same structure: a rectangular beam (with the following dimensions: $10m \times 1m \times 1m$) subdivided into 8 sub-domains and subjected to thermal or mechanical loads. The Young modulus is

$E = 10^9 Pa$ and the Poisson's ratio is $\nu = 0.25$. Meshing of the beam is done with 13270 elements (4 nodes tetrahedral elements).

5.1. Steady state first order validation

In this first example, the beam is clamped on one end and subjected to plane flexion through the application of a bending force $F = 49000N$ on the opposite end. The analytic solution (Incropera *et al.*, 2002) is compared to the mono-domain 3D computations (MulPhyDo and Abaqus) and to the 8-domain 3D analysis. The same mesh is used for all FE models. As one can see in Table 1, MulPhyDo and Abaqus gives a close response (0,3%).

Table 1. Maximal end beam displacements for different computations

MulPhyDo 1 and 8 sub-domains	Abaqus 6.3	Analytic beam solution
0.1844 m	0.1838 m	0.196 m

5.2. Transient first order problem (mono timestep)

In this second example, the whole beam presents an initial temperature of $20^\circ C$ and one of its ends is suddenly set at $100^\circ C$. The calculations are done for a thermal conductivity $\lambda = 40000 W.m^{-1}.^\circ C^{-1}$, a specific heat $c = 5 J.kg^{-1}.^\circ C^{-1}$ and a volumetric mass $\rho = 1000 kg.m^{-3}$. The analytic model best-matching this benchmark is the semi-infinite solid model (Incropera *et al.*, 2002) submitted to a temperature of $100^\circ C$ suddenly applied on its surface, where the temperature dependence with the distance x to the surface and the time t is given by the equation:

$$T(x, t) = T_s + (T_i - T_s) \operatorname{erf}\left(\frac{x}{2\sqrt{at}}\right) \quad [57]$$

where $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{t=0}^x \exp(-t^2) dt$ is the Gaussian error function, and a is the thermal diffusivity coefficient ($a = \frac{\lambda}{\rho c}$).

Figure 6 illustrates the evolution of the temperature at the node P (situated along the neutral fiber at a distance of 40 cm of the face subjected to the thermal load), for different timesteps and two different values for the parameter α in the numerical Euler's integration scheme ($\alpha = 1$ in Figure 6(a) and $\alpha = 0.5$ in Figure 6(b)).

5.3. Transient second order test (mono timestep)

Here, in this third example, the beam is clamped at one of its ends and submitted to a uniform traction force $F_t = 2.205 \cdot 10^6 N$ on the opposite end. This load is applied via the time-dependent function given in the Figure 7.

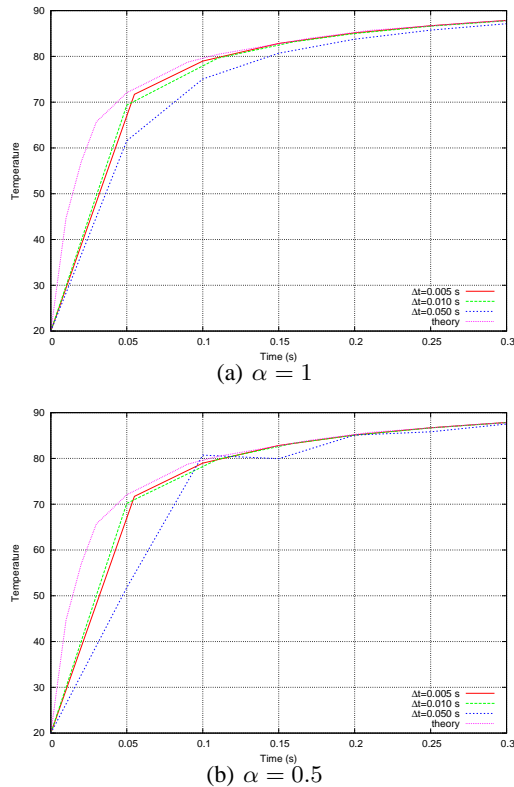


Figure 6. Evolution of the temperature at node P

The integration parameters retained for the simulation with MulPhyDo are $\beta = 0.5$ and $\gamma = 0.25$, which satisfy the stability condition $\gamma = \frac{1}{2}\beta$. We used a timestep $\Delta t = 0.5 \text{ ms}$. The small differences in the oscillations between Abaqus and MulPhyDo results (see Figure 7) are due to the integration parameters β et γ which are different in Abaqus where they are coupled via a third parameter α (named collocation parameter); this parameter is fixed to -0.05 by default (Hibbitt *et al.*, 1997; Hilber *et al.*, 1978) (which corresponds to $\beta = 0.276$ and $\gamma = 0.55$) and ensures an optimal precision of the integration schemes. Whereas, results obtained in both cases, Abaqus with mono-domain simulation and MulPhyDo with 8 sub-domains, are globally in accordance.

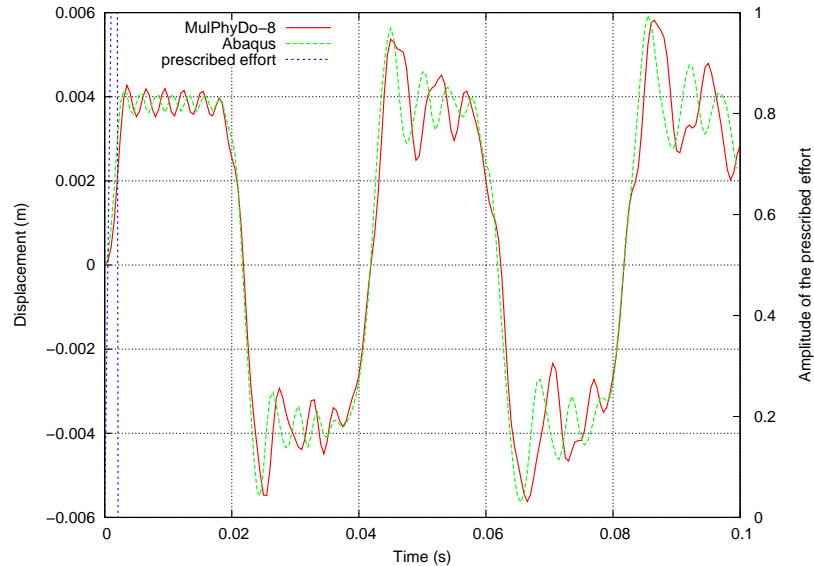


Figure 7. Comparison of the displacements obtained with MulPhyDo and Abaqus 6.4

5.4. Mono-timestepping versus multi-timestepping analysis

The decomposition strategies allow optimization of the transient computations as in steady state analysis. But in the case of the multi-timestep transient analysis, it is necessary to adapt these strategies to the characteristic times of the different zones. Typically, the zones submitted to fast and highly non-linear phenomena must be integrated using smaller timesteps. Tests have been performed to define the best timesteps. We use the same example as in the subsection 5.2. We defined two cases: in the first one, half part of the structure have a prescribed timestep twice to the minimal timestep of the structure, in the other case, 3 sub-domains have a prescribed timestep twice to the minimal timestep and 3 fourfold to the minimal timestep. The ratios between the timesteps of the sub-domains are shown in Figure 8 for both analysis. In Figure 9, the temperature of the node P for both cases is compared to the mono timestep evolution. In the Table 2, we compare the solutions obtained in multi and mono-time analysis. These results show the interest of the development of the multi-time computation: small error (less than 1%) and significative computational time reduction (up to 24%).

2	2	2	2	1	1	1	1	$T = 100\text{ }^{\circ}\text{C}$
---	---	---	---	---	---	---	---	-----------------------------------

4	4	4	2	2	2	1	1	$T = 100\text{ }^{\circ}\text{C}$
---	---	---	---	---	---	---	---	-----------------------------------

Figure 8. Mapping of the different timestep ratios for the 2 test-cases

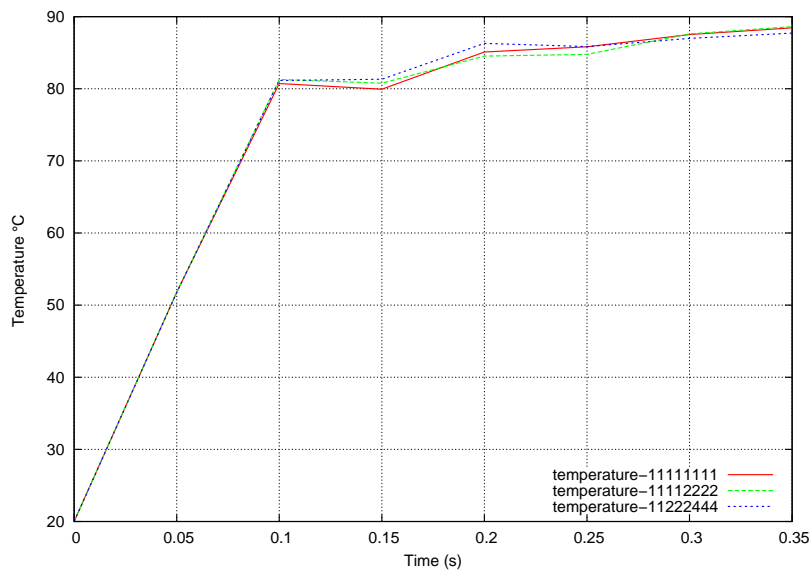


Figure 9. Evolution of the temperature at node *P* for the 2 multi-time configurations and comparison to the mono-time evolution

6. Industrial benchmark

6.1. Definition of the benchmark

The benchmark is an assembly used in power converters: a chip is brazed on a first DBC (Direct Bonded Copper) substrate and connected to a second one by the use of 12 bumps connectors (one lying copper cylinder with top and bottom brazes) as presented in Figure 10. The current circulates in the inner copper layers of the DBC substrates, while the heat generated by the chip commutations goes through the bumps and DBC substrates (see Figure 11).

Table 2. Comparison of mono-time versus multi-time computations

	Mono-time 11111111	Multi-time 11112222	Multi-time 11222444
Error at first increment (0.05 s)	reference	-0.2%	-0.3%
Error at last increment (0.35 s)	reference	+0.2%	-0.8%
Time decrease	reference	-17%	-24%

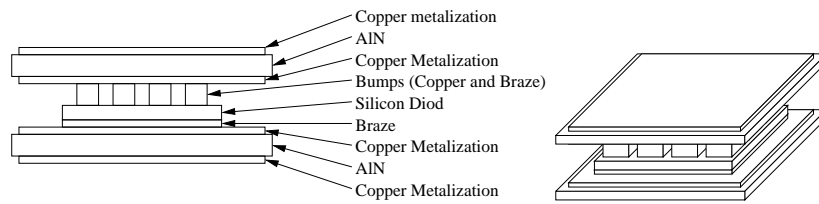


Figure 10. The different parts of the assembly

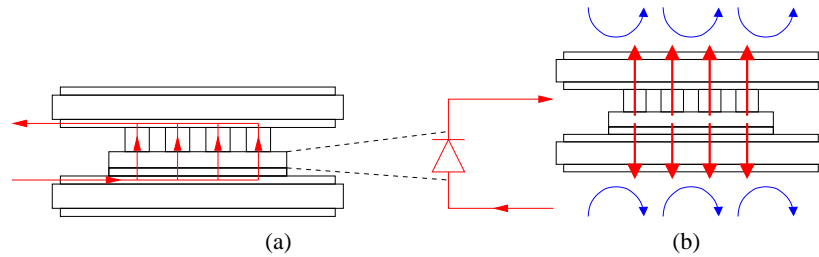


Figure 11. Current circulation (a) and heat circulation with convective cooling (b)

As the repartition of the heat between the top face and the bottom face of the chip is not known, it is assumed that half of heat escapes by the top face and half by the bottom face of the chip. Moreover, the bump behavior is the most significative. So we only modelize the top of the assembly.

6.2. Electro-thermal model

The volumetric calorific power (in $W m^{-3}$) dissipated by the chip at each timestep might be approximated by the following relation:

$$P = -4.41 \cdot 10^8 - 2.04 \cdot 10^8 \alpha - 0.533I + 5.606I\alpha + 9.92 \cdot 10^{-10} I^2 + 2.28 \cdot 10^7 T$$

[58]

where T ($^{\circ}C$) is the temperature of the chip, I ($A\ m^{-3}$) is the volumetric current density and α (adimensional value between 0 and 1) is the cyclic ratio of the power converter. This expression has been extracted from the study defined with the electro-thermal chip model developed by (Mussard *et al.*, 2004). For each timestep, we perform an electric calculation followed by a thermal one. At the initial time, the whole structure is set at a potential of $0\ V$ and a temperature of $50^{\circ}C$.

Two kinds of decomposition strategies are retained, with a specific idea for each one:

- a “slice” tearing: each part or each group of parts is a sub-domain. In this case, the interfacial problem involves large contact surfaces (see Figure 12(a));
- a “block” tearing: the assembly is torn through the parts. This option limits the size of the interface because it uses only the part thicknesses (see Figure 12(b)).

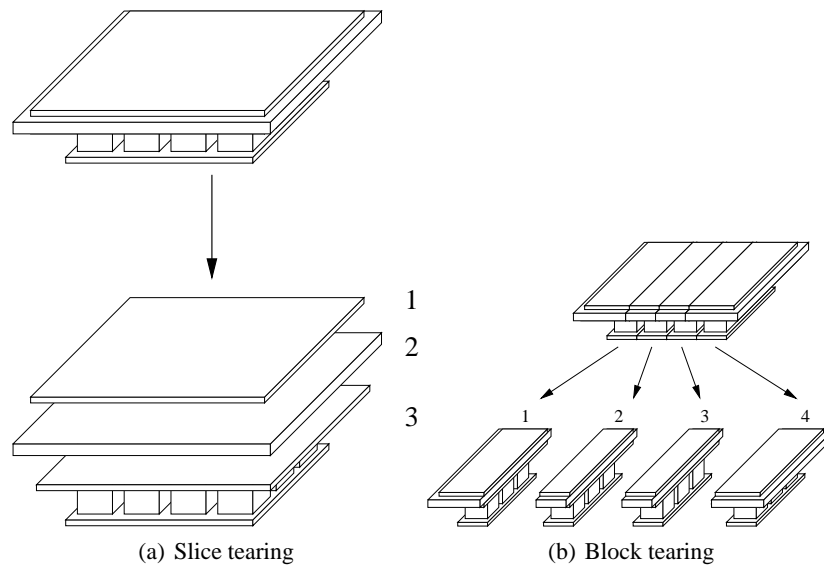


Figure 12. *Slice-tearings and blocks-tearing of the DBC*

For a transient multi-timestep analysis, the slice-tearing is more adapted, since it allows different timesteps on the different parts. In the presented version of the software, a mixed partition of the workpiece using both “slice” and “block” tearing cannot be used since n -uplets ($n > 2$) interfacial nodes are not implemented. In a new version, this has been done.

The properties of the material used in the model are resumed in the Table 3: coefficients of dilatation α , heat capacities C_p , thermal conductivities λ , Young’s modulus E , Poisson’s ratio ν , densities ρ and electrical resistivities ρ_{elec} . The initial temperature for each material is identical $T_{ini} = 20^{\circ}C$.

6.3. Mono and multi-timesteps computation

We use a slice-tearing and set up timesteps in sub-domains 1 and 2 twice the timestep in the sub-domain 3. In fact, this one includes the chip and is submitted to the fastest electro-thermal phenomenon (see Figure 13 for details). The next figure shows the evolution of temperature for a node situated on the bottom surface of the chip for mono (temperature-111) and multi timestep (temperature-112) analysis, leading to very small differences between the two methods: less than 2% for the whole computation (see Figure 14). The time computations are reported in Table 4 for two different computers with the corresponding time reductions. The relatively high values of the ratio reported in Table 4 comes from the fact that the number of elements with a greater timestep value in the model is very low.

Table 3. Table of the material properties

	Copper	Silicon	AlN	Bump
α	$1.64 \cdot 10^{-5}$	$3.24 \cdot 10^{-6}$	$4.5 \cdot 10^{-6}$	$2.19 \cdot 10^{-5}$
$C_p(J/m \cdot ^\circ C)$	385	765	850	385
$\lambda(W/m \cdot ^\circ C)$	389	150	173	30
$E(Pa)$	$1.17 \cdot 10^{11}$	$1.5 \cdot 10^{11}$	$3.3 \cdot 10^{11}$	$5.27 \cdot 10^{10}$
ν	0.343	0.278	0.25	0.3
$\rho(kg/m^3)$	8700	2330	3260	7360
$\rho_{elec}(\Omega/m)$	$1 \cdot 10^{-8}$	$3.8 \cdot 10^{-8}$	$1 \cdot 10^3$	$1 \cdot 10^{-6}$

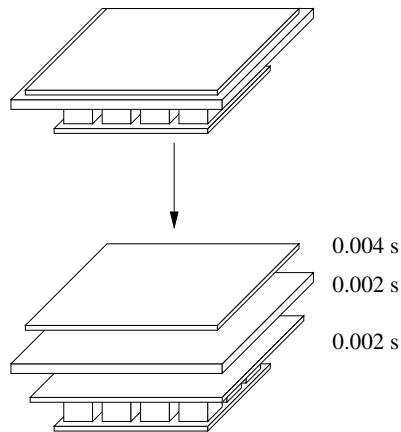


Figure 13. Timesteps definition for the multi-timestep analysis

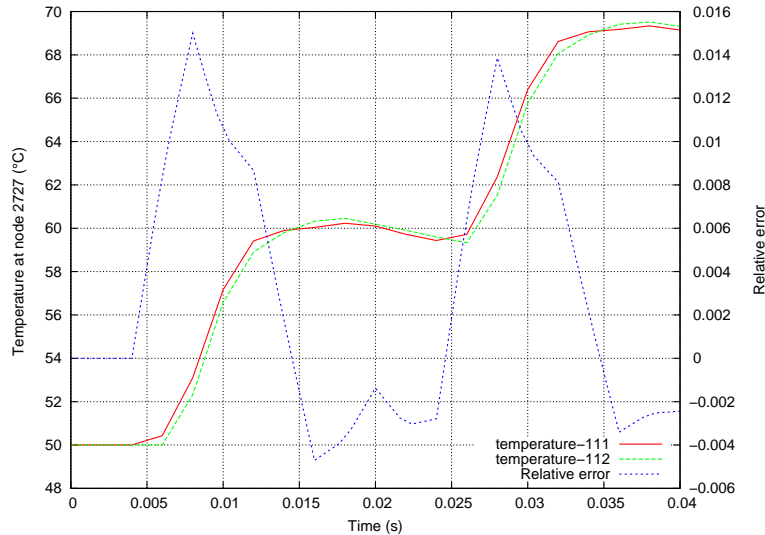


Figure 14. Temperature in mono and multi-time evolutions

Table 4. Computation times on two different computers and slice tearing

	mono timestep	multi-timestep	Ratio
PIII 1GHz	5778	5609	0.97
PIV 2.6 GHz	1903	1830	0.96

7. Conclusions

This work reports some theoretical results and a validation approach concerning multi-timestepping and the application to an industrial problem. Moreover, MulPhyDo includes now three physics (resistive electrics, thermics and structural dynamics), which permit multi-timestepping analysis for both of them. This tool can be currently used in industry for power converters modeling and explores an other way, compared to other ones (Hoppe *et al.*, 2003; Chow *et al.*, 2001). This first application leads to a plan for further developments:

- introduction of non-linearities by using the collaboration of explicit (for zones with fine timesteps) and implicit (for zones with larger timesteps) schemes;
- finest taking into account of the silicon chip heat generation in the power converters. In fact, the use of multi-timestepping allows the use of timesteps close to the commutation frequency of the chips.

The work of the authors has been supported by the funds of the Power Electronics Associated Research Laboratory of Alstom Transport in S emeac (France).

8. References

- Chandra R., Dagum L., Kohr D., Maydan D., McDonald J., Menon R., *Parallel Programming in OpenMP*, Academic Press, 2001.
- Chow P., Bailey C., Addison C., “ Solving non-linear electronic packaging problems on parallel computers using domain decomposition”, *12th International Conference on Domain Decomposition Methods*, 2001.
- Chow P., Lai C. H., “ Electronic Packaging and Reduction in Modelling Time Using Domain Decomposition”, *15th International Conference on Domain Decomposition Methods*, 2003.
- Combesure A., Gravouil A., “ A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis”, *Computer Methods in Applied Mechanics and Engineering*, vol. 191, p. 1129-1157, 2002.
- Farhat C., “ A method of finite element tearing and interconnecting and its parallel solution algorithm”, *International Journal of Numerical Methods in Engineering*, vol. 32, p. 1205-1227, 1991.
- Farhat C., Lesoinne M., LeTallec P., Pierson K., Rixen D., “ FETI-DP: a dual-primal unified FETI method - part I: A faster alternative to the two-level FETI method”, *International Journal of Numerical Methods in Engineering*, vol. 50(7), p. 1523-1544, 2001.
- Farhat C., Roux F. X., *Implicit parallel processing in structural mechanics*, Computational Mechanics Advances 2, 1994.
- Fragakis Y., Papadrakakis M., A unified framework for formulating Domain Decomposition Methods in Structural Mechanics, Technical report, Institute of Structural Analysis and Seismic Research - National Technical University Athens, 2002.
- Fragakis Y., Papadrakakis M., “ The mosaic of high performance domain decomposition methods for structural mechanics: formulation, interrelation and numerical efficiency of primal and dual methods”, *Computer Methods in Applied Mechanics and Engineering*, vol. 195, n° 35-36, p. 3799-3830, 2003.
- Gravouil A., Méthodes multi-échelles en temps et en espace avec décomposition de domaines pour la dynamique non-linéaire des structures, PhD thesis, LMT - ENS Cachan, 2000.
- Hibbitt K., Sorensen I., *Anaquis Theory Manual Version 5.7*, HKS, 1997.
- Hilber H. M., Hughes T. J. R., “ Collocation, dissipation and overshoot for time integration schemes in structural dynamics”, *Earthquake Engineering and Structural Dynamics*, vol. 6, p. 99-117, 1978.
- Hoppe R. H. W., “ Adaptative multigrid and domain decomposition methods in the computation of electromagnetic fields”, *Journal of Computational and Applied Mathematics*, vol. 168, p. 245-254, 2004.
- Hoppe R. H. W., Iliash Y., Ramminger S., Wachutka G., “ Domain Decomposition Methods in Electrothermomechanical Coupling Problems”, *15th International Conference on Domain Decomposition Methods*, 2003.
- Hu Y. C., Lu A. L. H., Cox A. L., Zwaenepoel W., “ OpenMp for networks of SMPs”, *Journal of parallel and distributed computing*, vol. 60, n° 12, p. 1512-1530, 2000.
- Hughes T. J. R., Liu W. K., “ Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples”, *Journal of Applied Mechanics*, vol. 45, p. 375-378, June, 1978a.

- Hughes T. J. R., Liu W. K., “ Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory”, *Journal of Applied Mechanics*, vol. 45, p. 371-374, June, 1978b.
- Incropera F. P., Dewitt D. P., *Introduction to Heat Transfert*, 4th edn, John Wiley & Sons, 2002.
- Kron G., *Diakoptics: The piecewise solution of large-scale systems*, MacDonald and Co, 1963.
- Lewis R. W., Schrefler B. A., “ The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media”, *Meccanica*, vol. 34(3), p. 231-232, 1999.
- Mussard L., Tounsi P., Austin P., Dorkel J. M., Antonini E., “ New Electro-Thermal Modeling Method for IGBT Power Module”, *IEEE Bipolar/BiCMOS Circuit and Technology Meeting 2004*, 2004.
- Pantalé O., “ An object-oriented programming of an explicit dynamics code: application to impact simulation”, *Advances in Engineering Software*, vol. 33, p. 297-306, 2002.
- Pantalé O., “ Parallelization of an Object-Oriented FEM Dynamics Code: Influence of the strategies on the SpeedUp”, *Advances in Engineering Software*, vol. 36, n° 6, p. 361-373, 2005.
- Papadrakakis M., *Parallel Solution Methods in Computational Mechanics, Chapter 3: Domain Decomposition Techniques for Computational Structural Mechanics*, edited by M. Papadrakakis, John Wiley & Sons Ltd, 1997.
- Pérez C., Priol T., Ribes A., “ A Parallel CORBA Component Model for Numerical Code Coupling”, *The International Journal of High Performance Computing Applications*, vol. 17(4), p. 417-429, 2003.
- Rixen D. J., “ Dual Schur Complement Method for Semi-Definite Problems”, *Contemporary Mathematics*, 1998.
- Roux F. X., “ Méthodes de résolution par sous-domaines en statique”, *La Recherche Aéronautique*, vol. 1, p. 37-48, 1990.
- Schwarz H. A., “ Uber einige Abbildungsaufgaben”, *Gesammelte Mathematische Abhandlungen*, vol. 11, p. 65-83, 1869.
- Smolinski P., Palmer T., “ Procedures for multi-time step integration of element-free Galerkin methods for diffusion problems”, *Computers and Structures*, vol. 77, p. 171-183, 2000.