



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 15703

To cite this version:

Menanteau, Laurent and Pantalé, Olivier and Caperaa, Serge *A coupled electro-thermo-mechanical FEM code for large scale problems including multi-domain and multiple time-step aspects*. (2005) In: Coupled problems 2005, 25 May 2005 - 27 May 2005 (Santorin, Greece).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

A COUPLED ELECTRO-THERMO-MECHANICAL FEM CODE FOR LARGE SCALE PROBLEMS INCLUDING MULTI-DOMAIN AND MULTIPLE TIME-STEP ASPECTS

Laurent Menanteau *, Olivier Pantale * and Serge Caperaa*

*Laboratoire Genie de Production - Ecole Nationale d'Ingenieurs de Tarbes
47 av d'Azereix, 65016 Tarbes Cedex, France
e-mail: Laurent.Menanteau@enit.fr, web page: <http://www.enit.fr/>

Key words: Finite Element, Dual Domain Decomposition Method, Parallelization, OpenMP, SMP computer

Abstract. *This work concerns the development of a virtual prototyping tool for large scale electro-thermo-mechanical simulation of power converters used in railway transport including multi-domain and multiple time-steps aspects. For this purpose, Domain Decomposition Method (DDM) is used to give on one hand the ability to treat large scale problems and on the other hand, for transient analysis, the ability to use different time-steps in different parts of the numerical model. An Object-Oriented programming in C++ is used for the development of the application, allowing integration in a single interface, high evolutivity and code maintenance. Finally, parallelization is introduced to improve the performances of the code and obtain computing times compatible with industrial developments.*

In a first part, we present the formulation of DDM based on a dual approach in steady-state analysis including the building of the interfacial problem. Then, as presented by Combescure et al. we extend this formulation to multiple time-step transient analysis. The Euler semi-implicit scheme used to integrate first-order systems and the Newmark scheme used to integrate second order systems are detailed for both single time-step and multiple time-step computations. Then, we present a discussion concerning the choice of the continuity at the interface for first-order systems in single time-step and multiple time-step transient analysis.

In the second part, the numerical implementation of the proposed approach in the FEM code DynELA is presented with a focus on the treatment of multi-physic aspects. Parallelization of the code on a Shared Memory Processors (SMPs) computer is achieved using the OpenMP programming standard.

The last part of this work concerns the study of an industrial benchmark concerning the power converters: the electro-thermal simulation of a semi-conductor chip in transient analysis. This example allows to compare different strategies of tearing into subdomains and to couple different time-steps on the same structure.

1 INTRODUCTION

Industrial products are made of many parts, and 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. Due to their high integration level, these products are difficult to instrument whereas the demand for reliability increases. Therefore there is a real need in more realistic models for 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 inconvenients of this approach relies on the difficulties in building realistic analytic models for non-linear and complex behaviours; therefore, numerical models are usually preferred.

A multi-physics approach needs the use of several numerical codes and data exchanges between them. Such a platform enhances some inconvenients:

- a relatively high cost due to the number of different softwares,
- the need of consulting many specialists,
- the need of developping specific data exchange softwares in order to establish the communications,
- the problems encountered when new code versions occur.

Therefore, in this work, we choose to build an unique, three-dimensional code with the following goals:

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

Domain Decomposition Methods (DDM) have 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 performances.

2 THEORETICAL APPROACH

2.1 Steady state multidomain computation

A dual DDM formulation is used, considering a structure subdivided into s subdomains. The whole finite element problem related to the structure can be set as the sum of all the elementary

finite element problems on the s subdomains (equation (1-a)) , associated with an equation enforcing the primal quantities continuity at the interface (equation(1-b)).

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

where $K^{(j)}$ is the matrix relative to the physical behaviour of the subdomain (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 subdomain j and $g_{int}^{(j)}$ is the vector of the actions of the adjacent subdomains on the subdomain j . The $B^{(j)}$ are localization matrices that select the degrees of freedom of the subdomain j which are related to the interface. These matrices are also used to express the interactions λ between the subdomains at the interface:

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

Then the steady state general problem can be written as:

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

or in a condensed form:

$$\begin{bmatrix} K^e & -B^{eT} \\ -B^e & 0 \end{bmatrix} \begin{bmatrix} q^e \\ \lambda \end{bmatrix} = \begin{bmatrix} g^e \\ 0 \end{bmatrix} \quad (4)$$

In the case of floating subdomains (i.e without any boundary conditions of Dirichlet type) and non-floating subdomains, the expressions of the primal unknowns are respectively of the form:

$$\begin{cases} q^{(j)} = K^{(j)-1} (g^{(j)} + B^{(j)T} \lambda) & \text{for non-floating subdomains} & \text{(a)} \\ \begin{cases} q^{(j)} = K^{(j)+} (g^{(j)} + B^{(j)T} \lambda) + R^{(j)}\gamma^{(j)} \\ (R^{(j)}\gamma^{(j)})^T (K^{(j)}q^{(j)}) = 0 \end{cases} & \text{for floating subdomains} & \text{(b)} \end{cases} \quad (5)$$

where $K^{(j)+}$ is the generalised inverse matrix of $K^{(j)}$, $R^{(j)}$ (kernel of $K^{(j)}$) includes the vectors of the base of the null-space of $K^{(j)}$, $\gamma^{(j)}$ shows the unknown norms of $R^{(j)}$. For non floating subdomains, $K^{(j)+}$ will be used to represent the inverse of $K^{(j)}$ (with $R^{(j)}\gamma^{(j)} = 0$). C. Farhat and F.-X. Roux⁴ have shown that it is not necessary to compute explicitly the generalised inverse of $K^{(j)}$; they proposed an iterative algorithm based on the partition of $K^{(j)}$. By reintroducing these expressions in the equation (1-b), we can define an interfacial problem of the form:

$$\begin{bmatrix} F_I & G_I \\ G_I^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} g_\lambda \\ g_\gamma \end{bmatrix} \quad (6)$$

where:

$$\begin{cases} F_I = \sum_{j=1}^s B^{(j)} K^{(j)+} B^{(j)T} \\ G_I = \begin{bmatrix} B^{(1)}R^{(1)} & \dots & B^{(f)}R^{(f)} \end{bmatrix} \\ \gamma = \begin{bmatrix} \gamma^{(1)T} & \dots & \gamma^{(f)T} \end{bmatrix}^T \\ g_\lambda = -\sum_{j=1}^s B^{(j)} K^{(j)+} g^{(j)} \\ g_\gamma = \begin{bmatrix} -R^{(1)T} g^{(1)} & \dots & -R^{(f)T} g^{(f)} \end{bmatrix}^T \end{cases} \quad (7)$$

F_I is commonly named “dual Schur matrix”.

The resolution of the interfacial problem gives λ and γ . These results are then introduced in the equations (1-a), leading to the resolution on the s subdomains.

2.2 Transient multidomain computation

As in section (2.1), we chose to use a dual DDM formulation; in transient analysis, we need 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.

2.2.1 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 subdomains linked by a continuity equation:

$$\begin{cases} C^{(j)}\dot{T}_n^{(j)} + K^{(j)}T_n^{(j)} = f_n^{(j)} + B^{(j)T} \lambda_n & \text{for } j \in \{1, \dots, s\} \quad (\text{a}) \\ \sum_{j=1}^s B^{(j)}w_n^{(j)} = 0 & (\text{b}) \end{cases} \quad (8)$$

The transient resolution requires the introduction of an integration scheme, as the Euler’s scheme:

$$T_n^{(j)} = T_{n-1}^{(j)} + (1 - \alpha^{(j)})\Delta t^{(j)}\dot{T}_{n-1}^{(j)} + \alpha^{(j)}\Delta t^{(j)}\dot{T}_n^{(j)} \quad , \quad \alpha^{(j)} \in [0, 1] \quad (9)$$

which can be written as the sum of a predictor and a corrector:

$$\begin{cases} T_n^{(j)} = pT_n^{(j)} + cT_n^{(j)} \\ pT_n^{(j)} = T_{n-1}^{(j)} + (1 - \alpha^{(j)})\Delta t^{(j)}\dot{T}_{n-1}^{(j)} \\ cT_n^{(j)} = \alpha^{(j)}\Delta t^{(j)}\dot{T}_n^{(j)} \end{cases} \quad (10)$$

We can choose two types of continuity at the interface: $T^{(j)}$ or $\dot{T}^{(j)}$. Regarding these two possible types of continuity, $w_n^{(j)}$ can be written in a general form:

$$w_n^{(j)} = p w_n^{(j)} + \mu^{(j)} \dot{T}_n^{(j)} \quad (11)$$

where:

$$\begin{cases} \text{if } w_n^{(j)} = \dot{T}_n^{(j)}, & p w_n^{(j)} = 0 & \text{and } \mu^{(j)} = 1 \\ \text{if } w_n^{(j)} = T_n^{(j)}, & p w_n^{(j)} = p T_n^{(j)} & \text{and } \mu^{(j)} = \alpha^{(j)} \Delta t^{(j)} \end{cases} \quad (12)$$

The general first order problem can be rewritten as:

$$\begin{bmatrix} \tilde{C}_n^{(1)} & \dots & 0 & -\mu^{(1)} B^{(1)T} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \tilde{C}_n^{(s)} & -\mu^{(s)} B^{(s)T} \\ -\mu^{(1)} B^{(1)} & \dots & -\mu^{(s)} B^{(s)} & 0 \end{bmatrix} \begin{bmatrix} \dot{T}_n^{(1)} \\ \vdots \\ \dot{T}_n^{(s)} \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \tilde{f}_n^{(1)} \\ \vdots \\ \tilde{f}_n^{(s)} \\ \sum_{j=1}^s B^{(j)} w_n^{(j)} \end{bmatrix} \quad (13)$$

where:

$$\tilde{C}_n^{(j)} = C_n^{(j)} + \alpha^{(j)} \Delta t^{(j)} K_n^{(j)} \quad \text{and} \quad \tilde{f}_n^{(j)} = \mu^{(j)} (f_n^{(j)} - K_n^{(j)p} w_n^{(j)}) \quad (14)$$

Then the general problem is decomposed in a “free” problem and in a “linked” problem^{3,5}: $\dot{T}_n^{(j)}$ is considered as the sum of a “free” vector $\dot{T}_n^{(j)}_{free}$ solution of the global problem with no interfacial interactions between the subdomains and a “link” vector $\dot{T}_n^{(j)}_{link}$ solution of the global problem just considering the interfacial interactions and excluding all the other actions. These two problems are:

$$\left\{ \begin{array}{l} \begin{bmatrix} \tilde{C}_n^{(1)} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \tilde{C}_n^{(s)} & 0 \\ 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{T}_n^{(1)}_{free} \\ \vdots \\ \dot{T}_n^{(s)}_{free} \\ 0 \end{bmatrix} = \begin{bmatrix} f_n^{(1)} - K_n^{(1)p} T_n^{(1)} \\ \vdots \\ f_n^{(s)} - K_n^{(s)p} T_n^{(s)} \\ 0 \end{bmatrix} \\ \text{and} \\ \begin{bmatrix} \mu^{(1)} \tilde{C}_n^{(1)} & \dots & 0 & -\mu^{(1)} B^{(1)T} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \mu^{(s)} \tilde{C}_n^{(s)} & -\mu^{(s)} B^{(s)T} \\ -\mu^{(1)} B^{(1)} & \dots & -\mu^{(s)} B^{(s)} & 0 \end{bmatrix} \begin{bmatrix} \dot{T}_n^{(1)}_{link} \\ \vdots \\ \dot{T}_n^{(s)}_{link} \\ \lambda_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sum_{j=1}^s B^{(j)} w_n^{(j)}_{free} \end{bmatrix} \end{array} \right. \quad (15)$$

When all the subdomains share the same time-step $\Delta t^{(j)}$, the interfacial problem is built for each time-step by using the same procedure as the one used for the steady state analysis. As there is no floating subdomains in a transient computation, its expression can be simplified as:

$$H_n \lambda_n = f_{\lambda_n} \quad (16)$$

where:

$$H_n = \sum_{j=1}^s \mu^{(j)} B^{(j)} \tilde{C}_n^{(j)} B^{(j)T} \quad \text{and} \quad f_{\lambda_n} = - \sum_{j=1}^s B^{(j)} w_n^{(j)}_{free} \quad (17)$$

The resolutions of the free problems allow to solve the interfacial problem; λ_n is then reintroduced in the expressions of the link problems related to each subdomain and permit to solve them. The expressions of $\dot{T}_n^{(j)}$ can be computed and using the integration scheme (10), we can obtain the temperatures $T_n^{(j)}$.

In the case of different time-steps used in different subdomains, the interfacial problem is computed for each minimal time-step of the structure. So it requires the interpolation of the

quantities used in continuity and corresponding to the subdomains which have time-steps different from the minimal time-step (in this work, it is supposed that all the time-steps are multiples of the minimal one). The interfacial problem is modified:

$$H_i \lambda_i = f_{\lambda_i} \quad (18)$$

where:

$$H_i = \sum_{j=1}^s \mu^{(j)} B^{(j)} \tilde{C}_i^{(j)} B^{(j)T} \quad \text{and} \quad f_{\lambda_n} = - \sum_{j=1}^s B^{(j)} \hat{w}_i^{(j)} \quad (19)$$

$$\hat{w}_i^{(j)} = \left(1 - \frac{i}{k^{(j)}}\right) w_0^{(j)} + \frac{i}{k^{(j)}} w_k^{(j)} \quad \text{and} \quad \Delta t^{(j)} = k^{(j)} \min(\Delta t^{(j)}) \quad (20)$$

The resolution of the problem for each minimal time-step is done in the same way as for one unique time-step on the structure.

2.2.2 Second order transient problems

As previously, the second order transient problem is formulated using a dual DDM:

$$\begin{cases} M_n^{(j)} \ddot{u}_n^{(j)} + C_n^{(j)} \dot{u}_n^{(j)} + K_n^{(j)} u_n^{(j)} = f_n^{(j)} & \text{for } j \in \{1, \dots, s\} \quad (\text{a}) \\ \sum_{j=1}^s B^{(j)} w_n^{(j)} = 0 & (\text{b}) \end{cases} \quad (21)$$

Here we use a Newmark integration scheme:

$$\begin{cases} \dot{u}_n^{(j)} = \underbrace{\dot{u}_{n-1}^{(j)} + (1 - \gamma^{(j)}) \Delta t^{(j)} \ddot{u}_{n-1}^{(j)}}_{p \dot{u}_n^{(j)}} + \gamma^{(j)} \Delta t^{(j)} \ddot{u}_n^{(j)} \\ u_n^{(j)} = \underbrace{u_{n-1}^{(j)} + \Delta t^{(j)} \dot{u}_{n-1}^{(j)} + \left(\frac{1}{2} - \beta^{(j)}\right) (\Delta t^{(j)})^2 \ddot{u}_{n-1}^{(j)}}_{p u_n^{(j)}} + \beta^{(j)} (\Delta t^{(j)})^2 \ddot{u}_n^{(j)} \end{cases} \quad (22)$$

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

$$\begin{cases} \text{if } w_n^{(j)} = \ddot{u}_n^{(j)}, \quad p w_n^{(j)} = 0 & \text{and } \mu^{(j)} = 1 \\ \text{if } w_n^{(j)} = \dot{u}_n^{(j)}, \quad p w_n^{(j)} = p \dot{u}_n^{(j)} & \text{and } \mu^{(j)} = \gamma^{(j)} \Delta t^{(j)} \\ \text{if } w_n^{(j)} = u_n^{(j)}, \quad p w_n^{(j)} = p u_n^{(j)} & \text{and } \mu^{(j)} = \beta^{(j)} (\Delta t^{(j)})^2 \end{cases} \quad (23)$$

The second order transient problem is then solved as for the first order transient problem, for both mono and multi-time-steps in the structure.

2.2.3 Continuity at the interface

Stability of the integration schemes for a multi-domain and mono-time-step analysis Basing on the stability studies done for dynamics by Combescure and Gravouil^{3, 5} using the energy methods developed by Hughes^{11, 10}, we computed¹³ an interfacial energy term :

$$E_{interface} = \frac{1}{\Delta t} [\lambda_n]^T \sum_{j=1}^s B^{(j)} [\dot{T}_n^{(j)}] \quad (24)$$

If $E_{interface} < 0$, the system can lead to numerical dissipations, if $E_{interface} = 0$, the scheme is as stable as in a mono-domain analysis and if $E_{interface} > 0$, the system can lead to numerical instabilities.

For a first order problem, we have the following results:

- for a continuity of the $T_n^{(j)}$ at the interface, $E_{interface} = 0$ if all the $\alpha^{(j)} = 1/2$ and if the continuity of all the $\dot{T}_0^{(j)}$ is supposed,
- for a continuity in $\dot{T}_n^{(j)}$ the term $E_{interface}$ vanishes unconditionnally null.

For second order problems, we recall here the results obtained by Combescure and Gravouil:

- for a continuity in $\ddot{u}_n^{(j)}$, $E_{interface} = 0$ if all the $\gamma^{(j)}$ are equal,
- for a continuity in $\dot{u}_n^{(j)}$, $E_{interface}$ vanishes unconditionnally null,
- for a continuity in $u_n^{(j)}$, $E_{interface} = 0$ if we choose $\gamma^{(j)} = \frac{1}{2}\beta^{(j)}$ and if the continuity of all the $\dot{u}_0^{(j)}$ is supposed.

For these reasons, we consider in this work continuities in terms of $\dot{T}_n^{(j)}$ and $\dot{u}_n^{(j)}$.

Precision of the integration schemes in multi-time-step analysis Taking account the options previously retained, we interested in the error introduced by the multi-time-stepping. For this reason, we adapted the expression (24) to the multi-time-steps case, according to Combescure and Gravouil work, and we finally determined that:

$$E_{interface} \leq 0 \tag{25}$$

So, it confirms what has ever been shown for dynamics, i.e that for continuities in $\dot{T}_n^{(j)}$ and in $\dot{u}_n^{(j)}$, integration schemes keep stable even if they dissipate energy at the interface. Moreover, if the variations of the continuity quantities are linear, there is no dissipation in the interface.

3 IMPLEMENTATION

3.1 Structure of the code

We implemented the multi-domain solver by adding new classes librairies in the home developed FEM large deformation code *DynELA*¹⁵. The whole model is an instance of the class *Structure* composed of several instances of the class *Physic*. Each physic has one or several subdomains (class *Domain*) which are made of one or several meshes (class *Grid*). Finally, one or more solvers (class *Solver*) are related to each subdomain: it 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 1). Currently, this solver, named *MulPhyDo* (for Multi-Physic and Multi-Domain) is composed of an electrical solver, a thermal one and a mechanical one.

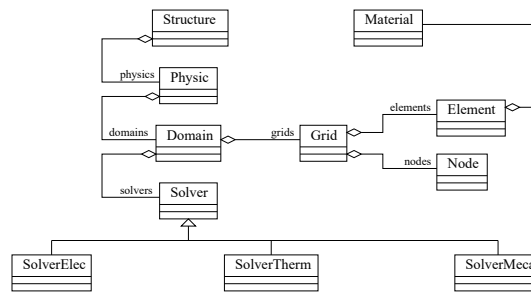


Figure 1: UML diagram of the solver

3.2 Parallelization

3.2.1 The DDM step by step

Starting from a calculation domain which has been partitionned into several subdomains and one interface, the computation steps are the following ones:

1. building the Finite Elements problems related to the several subdomains,
2. building the interfacial problem,
3. solving the interfacial problem,
4. solving the problems on each subdomain.

These four steps may be parallelized, but the steps (1 and 4) concerning the several subdomains are more time-consuming than the two other ones (2 and 3); thus, we parallelized only those two steps.

3.2.2 The OpenMP standard

OpenMp is a recently developed programming standard (1997) offering a standard interface for softwares developed in FORTRAN and C/C++^{1,9,16}. It can only be integrated on SMP 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 instructions 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 datas 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 in several elementary threads that can be done simultaneously by several processors. So it is a parallelism of control: several operations done simultaneously by several processors (see figure 2). 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 arbitrary defined as the “master”, the other ones being created from it.

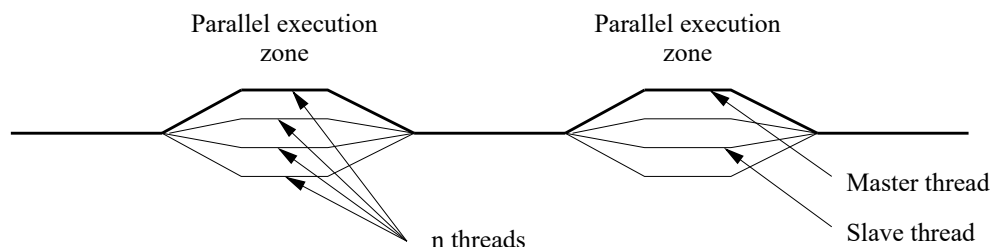


Figure 2: Parallel computing with OpenMP

All the tests and validation of parallelized code have been done on a Compaq ProLiant 8000 with 8 processors Intel Xeon PIII 550/2Mb and 5 Gb of shared memory. This computer works under Linux Redhat 8.0 and uses an Intel C++ 7.1 compiler (for the parallel version of the code) or a g++ compiler (for the non-parallel one).

4 VALIDATION BENCHMARKS

4.1 Steady state validation

In this example, a clamped beam, subjected to plan flexion, is divided into 8 subdomains. The dimensions of the beam are $10\text{m} \times 1\text{m} \times 1\text{m}$, the applied force is 49000 N, the Young modulus of the material is $E = 10^9\text{Pa}$ and the Poisson’s ratio is $\nu = 0.25$ (figure 3).

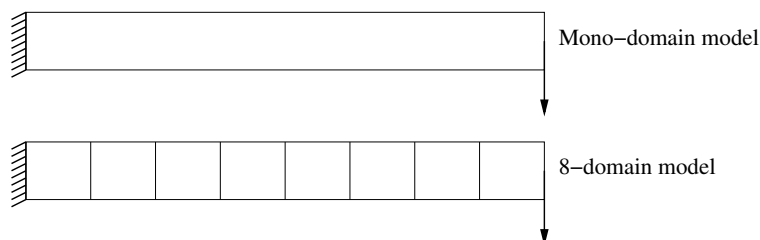


Figure 3: Clamped beam with the several subdomains

The analytic solution¹² is compared to the mono-domain 3D computations (MulPhyDo and Abaqus) and to the 8-domain 3D analysis. The same mesh is used for the three FE models. As one can see on (table 1), Abaqus gives a close response (0,3 %).

4.2 Influence of the number of subdomains

In this benchmark, we study a beam (with the same geometrical characteristics as in section 4.1) (figure 4) submitted to thermal sollicitations: the first quarter of the beam is submitted to

MulPhyDo 1 and 8 subdomains	Abaqus 6.3	Analytic beam solution
0.1844 m	0.1838 m	0.196 m

Table 1: Maximal end beam displacements for different computations

heat generation (a volumic flux $\Phi_v = 9000 \text{ W.m}^{-3}$), whereas the opposite wall is submitted to a convection condition (heat coefficient $h_c = 500 \text{ W.m}^{-2}.K^{-1}$, fluid temperature $T_{inf} = 0^\circ\text{C}$). The other walls are adiabatic (so, the flux going through the bar is $\Phi = \Phi_v \times (\frac{L}{4}) = 22500 \text{ W}$). The thermal conductivity of the material is $\lambda = 400 \text{ W.m}^{-1}.K^{-1}$.

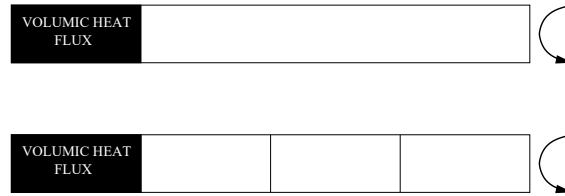


Figure 4: Beam submitted to a volumic heat generation and cooled by convection and one example of division into 4 subdomains

We focus here on the influence of the number of subdomains in the model (when it is divided into 1, 2, 3, 4, 6 and 8 subdomains) on the computation time. Figure 5 shows an important reduction of the computation time (85 % when the structure is divided in 8 subdomains). In the same time, we remark an increase of the time needed for the resolution of the interfacial problem (note that the time scales are different).

4.3 Transient first order problem (mono-time-step)

The validation of the transient first order mono-time-step analysis is done with the same beam. In this example, the whole beam presents an initial temperature of 20°C and one of its ends is suddenly set at 100°C . A temperature gradient occurs in the beam. The calculations are done for a thermal conductivity $\lambda = 40000 \text{ W.m}^{-1}.^\circ\text{C}^{-1}$, a specific heat $c = 5 \text{ J.kg}^{-1}.^\circ\text{C}^{-1}$ and a volumic mass $\rho = 1000 \text{ kg.m}^{-3}$. The analytic model best-matching this benchmark is the semi-infinite solid model¹² submitted to a temperature of 100°C suddenly applied on its surface (figure 6). The temperature dependance with the distance x to the surface and the time t is given by the equation:

$$T(x, t) = T_s + (T_i - T_s) \text{erf}\left(\frac{x}{2\sqrt{at}}\right) \quad (26)$$

where erf is the Gaussian error function, and a is the thermal diffusivity coefficient ($a = \frac{\lambda}{\rho c}$).

The beam is divided into 8 equal subdomains. On figure 7, we compare the evolution of the temperature with time at the node P (situated on the neutral fiber at 40 cm of the face subjected

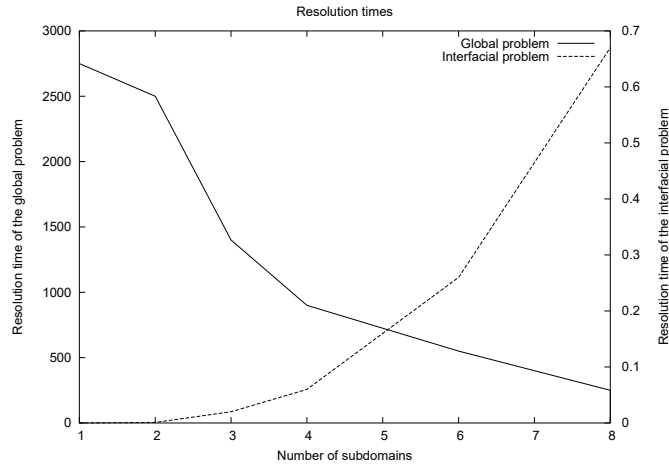


Figure 5: Resolution times of the global problem and of the interfacial problem for different numbers of subdomains

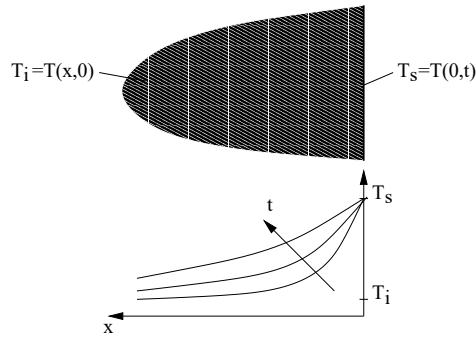


Figure 6: Semi-infinite solid model

to the thermal load), for different time-steps and two different values for the parameter α in the numerical Euler's integration scheme ($\alpha = 1$ on the left side and $\alpha = 0.5$ on the right side). We remark an increase of the precision while we decrease the time-step value.

4.4 Transient second order test (mono-time-step)

The validation of the transient second order benchmark computation is based on the dynamic analysis of the beam previously defined. Here, this beam is clamped at one of its ends and submitted to a uniform traction force on the opposite end ($45000N/node$ and 49 nodes, i.e. $2.205 \cdot 10^6 N$ globally) (see figure 8). This load is applied via the time-dependent function given on the figure 9. The Young's modulus is $E = 10^9 Pa$ and the Poisson's coefficient is $\nu = 0.25$.

Simulations have been conducted with MulPhyDo for 8 subdomains and Abaqus 6.4 (figure

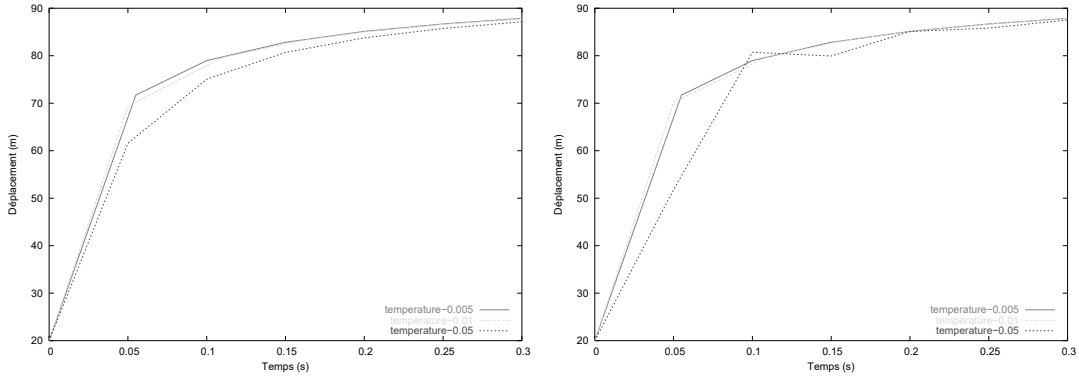


Figure 7: Evolution of the temperature at node P for $\alpha = 1$ (left) et $\alpha = 0,5$ (right)

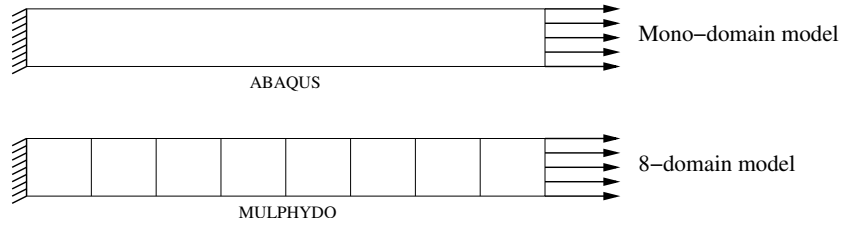


Figure 8: Beam submitted to an uniaxial and transient traction effort

10). 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 time-step $\Delta t = 0.5$ ms. The general aspect of the two curves is the same: the period and the displacement levels are similar. The small differences in the oscillations 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^{6,7} (which corresponds to $\beta = 0.276$ and $\gamma = 0.55$) and ensures an optimal precision of the integration schemes (periods, displacement levels).

4.5 Mono-time-stepping versus multi-time-stepping

The decomposition strategies allow optimization of the transient computations as in steady state analysis. But in the case of the multi-time-step transient analysis, it is necessary to adapt these strategies to the characteristic times of the different zones. Typically, the zones submitted to fast phenomena and highly non-linear have smallest time-steps than those which are further. Tests have been performed to define the best time-steps. We use the same example as in the subsection 4.3. The ratios between the time-steps of the subdomains are shown on figure 11 for the two analysis. In figure 12, the temperature of the node P is picked for the two cases and compared to the mono-steptime evolution. In the table 2, we compare the solutions obtained in

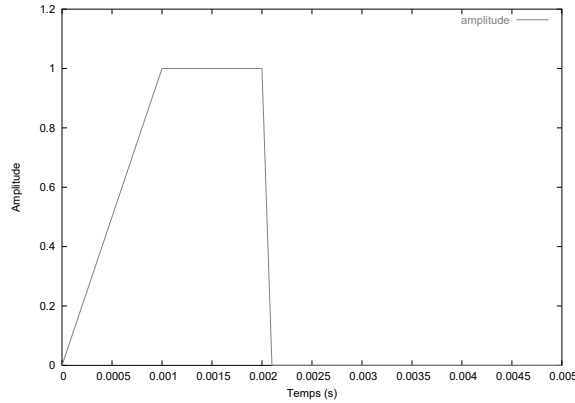


Figure 9: Amplitude of the effort on the beam.

multi and mono-time analysis. These results show the interest of the development of the multi-time computation: small error (less than 1%) and significant computational time reducing (up to 24%).

	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 gain	reference	-17%	-24%

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

5 INDUSTRIAL BENCHMARK

5.1 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 connectives (one lying copper cylinder with top and bottom brazes). Here, the bumps are replaced by parallelepipeds defined using a material with equivalent properties (figure 13). 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 (figure 14).

As the repartition of the heat between the top and the bottom of the chip is not known, it is assumed that half of heat escapes by the top and half by the bottom. Moreover, the bump behaviour is the most significative. So we only modelize the top of the assembly (figure 15).

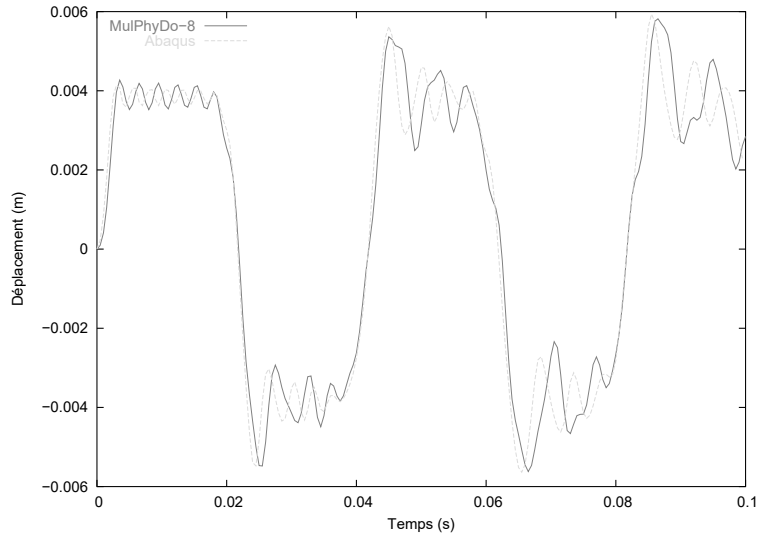


Figure 10: Déplacement en bout obtenu avec MulPhyDo et Abaqus 6.4

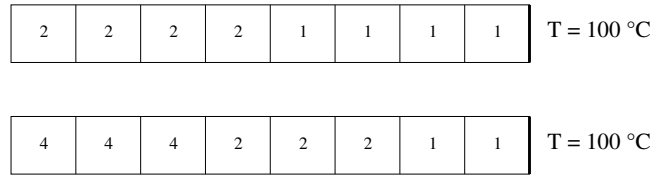


Figure 11: Mapping of the differents time-step ratios for the 2 test-cases

5.2 Electro-thermal model

The volumic calorific power (in $W m^{-3}$) dissipated by the chip at each time-step is given by the following formula:

$$P = -4.41 \cdot 10^8 - 2.04 \cdot 10^8 \alpha - 0.533I + 5.606I\alpha + 9.92 \cdot 10^{-10}I^2 + 2.28 \cdot 10^7T \quad (27)$$

where T (en degrés Celsius) is the temperature of the chip, I (in $A m^{-3}$) is the volumic current density and α (adimensionné, 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, and developed by L. Mussard¹⁴.

For each time-step, 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°C.

5.3 Decomposition strategies

We retained two kinds of decompositions with a specific idea for each one:

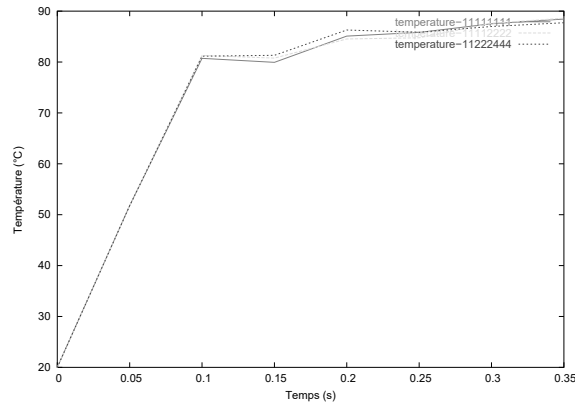


Figure 12: Evolution of the temperature at node P for the 2 multi-time configurations and comparison to the mono-time evolution.

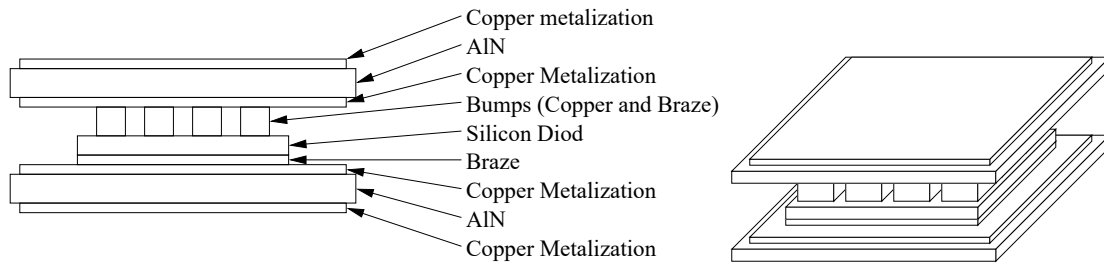


Figure 13: The differents parts of the assembly

- a “slice” tearing: each part or each group of parts is a subdomain. In this case, the interfacial problem has a long size because the interface involves large contact surfaces. We show two possible slice-tearing strategies in figure 16.
- 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 16).

For a transient multi-time-step analysis, the slice-tearings are more adapted, since they allow different time-steps on the different parts.

5.4 Material datas

The properties of the material used in the model are resumed in the table 3: coefficients of dilatation α , initial temperatures T_{ini} , thermal conductivities λ , Young’s modulus E , Poisson’s ratio ν , densities ρ and electrical resistivities ρ_{elec} .

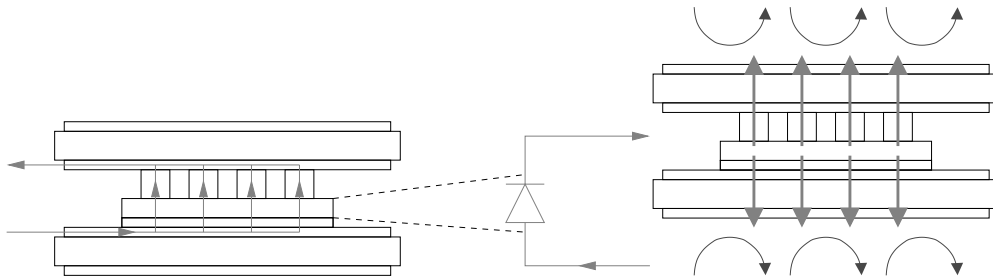


Figure 14: Current circulation (left) and heat circulation with convective cooling (right)

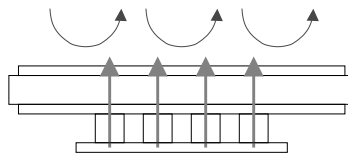


Figure 15: Model

5.5 Mono and multi-time-steps computation

5.5.1 Time-steps definition

Going back to the slice-tearing 1, we chose to set time-steps in subdomains 1 and 2 that are twice the time-step in the subdomain 3. In fact, this one includes the chip and is submitted to the fastest electro-thermal phenomenon (figure 17).

5.5.2 Results in mono and multi-time-steps analysis

We compare the evolution of the temperature for a node situated on the bottom surface of the chip for mono and multi-time-step analysis. We note very small differences between the two methods: less than 2% for the whole computation (see figure 18).

The time computations are reported in tables 4 for two different computers.

6 CONCLUSION

This work shows the interest of developing transient multi-time-stepping Finite Element codes, leading to important computation time reductions. Moreover, MulPhyDo includes now three physics (resistive electrics, thermics and structural dynamics), which permit multi-time-stepping 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^{8,2}.

This first application leads to a plan for further developments:

- introduction of non-linearities by using the collaboration of explicit (for zones with fine

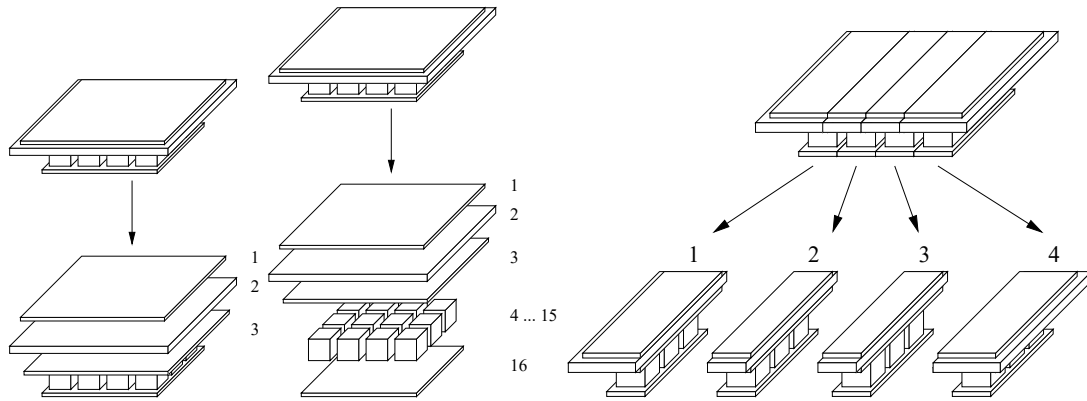


Figure 16: Slice-tearings 1 (left) and 2 (middle) and blocks-tearing

	α	T_{ini} ($^{\circ}\text{C}$)	λ ($\text{W}/\text{m}\cdot^{\circ}\text{C}$)	E (Pa)	ν	ρ (kg/m^3)	ρ_{elec} (Ω/m)
Copper	$1.64 \cdot 10^{-5}$	20	385	$1.17 \cdot 10^{11}$	0.343	8700	$1 \cdot 10^{-8}$
Silicon	$3.24 \cdot 10^{-6}$	20	150	$1.5 \cdot 10^{11}$	0.278	2330	$3.8 \cdot 10^{-8}$
AlN	$4.5 \cdot 10^{-6}$	20	173	$3.3 \cdot 10^{11}$	0.25	3260	$1 \cdot 10^3$
Bump	$2.19 \cdot 10^{-5}$	20	30	$5.27 \cdot 10^{10}$	0.3	7360	$1 \cdot 10^{-6}$

Table 3: Table of the material properties

time-steps) and implicit (for zones with larger time-steps) schemes,

- finest taking into account of the silicon chips heat generation in the power converters. In fact, the use of multi-time-stepping allows the use of time-steps very close to the commutation frequency of the chips

THANKS

The work of the authors has been supported by the funds given by the P.E.A.R.L. (Power Electronics Associated Research Laboratory).

REFERENCES

- [1] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. *Parallel Programming in OpenMP*. Academic Press, 2001.
- [2] P. Chow, C. Bailey, and C. Addison. Solving non-linear electronic packaging problems on parallel computers using domain decomposition. In *12th International Conference on Domain Decomposition Methods*, 2001.

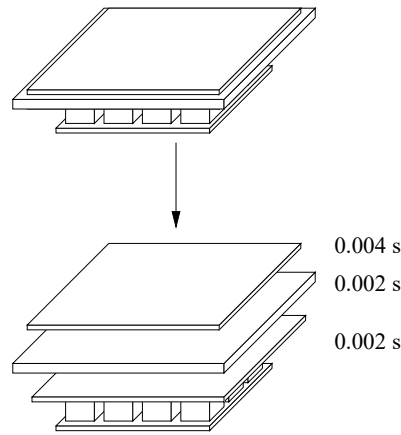


Figure 17: Time-steps definition for the multi-time-step analysis

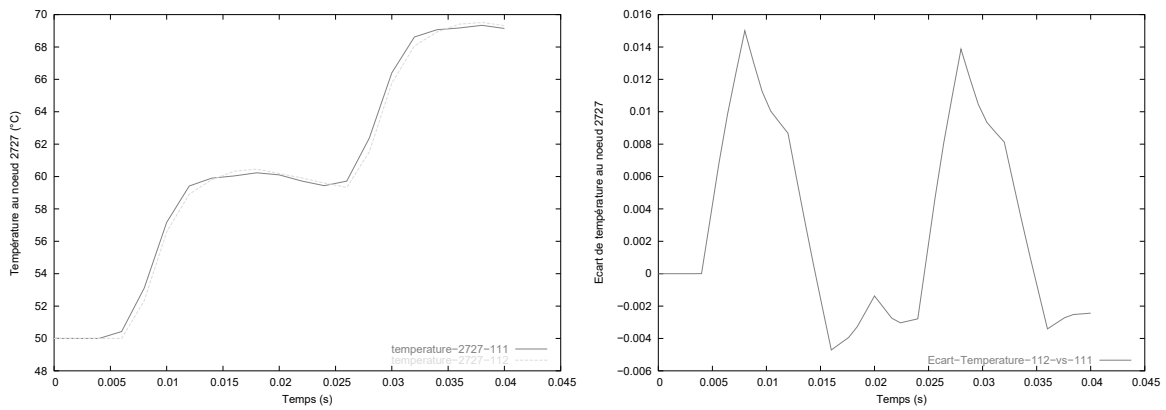


Figure 18: Temperature in mono and multi-time (left) and difference of temperatures (right) evolutions at node 2727

- [3] A. Combescure and A. Gravouil. A numerical scheme to couple subdomains with different time-steps for predominantly linear transient analysis. *Comput. Methods Appl. Mech. Engrg.*, 191:1129–1157, 2002.
- [4] C. Farhat. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal of Numerical Methods in Engineering*, 32:1205–1227, 1991.
- [5] A. Gravouil. *Méthode multi-échelles en temps et en espace avec décomposition de domaines pour la dynamique non-linéaire des structures*. PhD thesis, LMT - ENS Cachan, 2000.
- [6] Hibbitt, Karlsson, and Sorensen. *Theory Manual Version 5.7*. HKS, 1997.

PC Intel Processor Pentium III 1 GHz, 512 Mo SDRAM:

	Slice tearing 1 mono-time-step	Slice tearing 1 multi-time-step
Time in s	5778.5	5609.3
Ratio	1 (reference)	0.97

PC Intel Processor Pentium IV 2.6 GHz, 512 Mo DDRAM

	Découpage en tranches 1 monotemps	Slice tearing 1 mono-time-step
Time in s	1903	1830
Ratio	1 (reference)	0.96

Table 4: Computation times on two different computers

- [7] H. M. Hilber and T. J. R. Hughes. Collocation, dissipation and 'overshoot' for time integration schemes in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 6:99–117, 1978.
- [8] R.H.W. Hoppe, Y. Iliash, S. Ramminger, and G. Wachutka. Domain decomposition methods in electrothermomechanical coupling problems. In *15th International Conference on Domain Decomposition Methods*, 2003.
- [9] Y.C. Hu, A.L. Honghui Lu, A.L. Cox, and W. Zwaenepoel. Openmp for networks of smps. *Journal of parallel and distributed computing*, 60(12):1512–1530, 2000.
- [10] T.J.R. Hughes and W.K. Liu. Implicit-explicit finite elements in transient analysis: Implementation and numerical examples. *Journal of Applied Mechanics*, 45:375–378, June 1978.
- [11] T.J.R. Hughes and W.K. Liu. Implicit-explicit finite elements in transient analysis: Stability theory. *Journal of Applied Mechanics*, 45:371–374, June 1978.
- [12] F.P. Incropera and D.P. Dewitt. *Introduction to Heat Transfert - 2nd Edition*. Wiley, 1990.
- [13] L. Menanteau. *Développement d'un module de prototypage virtuel multiphysique, multidomaine et multitemps. Application aux convertisseurs de puissance*. PhD thesis, Institut Nationale Polytechnique de Toulouse, 2004.
- [14] L. Mussard, P. Tounsi, P. Austin, J.-M. Dorkel, and E. Antonini. New electro-thermal modeling method for igbt power module. *IEEE Bipolar/BiCMOS Circuit and Technology Meeting 2004*, 2004.

- [15] O. Pantalé. An object-oriented programming of an explicit dynamics code: application to impact simulation. *Advances in Engineering Software*, 33:297–306, 2002.
- [16] O. Pantalé. Parallelization of an object-oriented fem dynamics code: Influence of the strategies on the speedup. *To appear in Advances In Engineering Software*, 2005.