European Journal of Computational Mechanics
Vol. 21, Nos. 3–6, June–December 2012, 300–311

Taylor & Francis
Taylor & Francis Group

A first step toward a PGD-based time parallelisation strategy

Fabien Poulhaon, Francisco Chinesta* and Adrien Leygue

Laboratoire GeM de l'Ecole Centrale de Nantes – UMR CNRS 6183, 1 Rue de la Noë, 44231 Nantes Cedex 3, France

This paper proposes a new method for solving the heat transfer equation based on a parallelisation in time of the computation. A parametric multidimensional model is solved within the context of the Proper Generalised Decomposition (PGD). The initial field of temperature and the boundary conditions of the problem are treated as extra-coordinates, similar to time and space. Two main approaches are exposed: a "full" parallelisation based on an off-line parallel computation and a "partial" parallelisation based on a decomposition of the original problem. Thanks to an optimised overlapping strategy, the reattachment of the local solutions at the interfaces of the time subdomains can be improved. For large problems, the parallel execution of the algorithm provides an interesting speedup and opens new perspectives regarding real-time simulation.

Nous proposons dans cet article une nouvelle approche pour la résolution de l'équation de la chaleur en régime transitoire aboutissant à une parallélisation en temps du calcul. Cette méthode est basée sur l'utilisation de la Proper Generalised Decomposition (PGD) qui permet de résoudre à moindre coût des problèmes mettant en jeu un nombre important de paramètres. Ainsi, nous montrons qu'il est possible d'introduire à la fois la condition initiale mais aussi les conditions aux limites d'un problème comme des coordonnées supplémentaires du modèle numérique. L'accent est plus particulièrement mis sur deux approches: une parallélisation dite "totale" pour laquelle le calcul parallèle est réalisé hors ligne et une parallélisation que l'on qualifie de "partielle" et qui repose sur une décomposition du problème de base. Le recollement des solutions locales aux interfaces des sous-domaines temporels est affiné grâce une stratégie de recouvrement. Le calcul parallèle s'avère tout à fait intéressant de par le gain en temps qu'il confère, dès lors que la complexité du problème augmente. Ce résultat prometteur ouvre de nombreuses perpespectives quant au calcul en temps réel.

Keywords: model reduction; Proper Generalised Decomposition; parametric models; separated representation; time parallelisation

Mots-clés: réduction de modèles; représentation séparée; parallélisation en temps; modèles paramétriques

1. Introduction: parametric Proper Generalised Decomposition

Many models encountered in computational science involve a tremendous amount of dimensions, as for example, kinetic theory description of complex materials, quantum chemistry involving the solution of the Schrödinger equation and financial mathematics for credit markets modelling. If we use a classical mesh-based method like finite elements method (or finite differences, finite volumes, etc.), it is well known that the complexity of the model, that is to

*Corresponding author. Email: francisco.chinesta@ec-nantes.fr

ISSN 1779-7179 print/ISSN 1958-5829 online

http://dx.doi.org/10.1080/17797179.2012.714985
http://www.tandfonline.com

say the number of degrees of freedom, increases exponentially with the number of dimensions. Thus, the increase in the number of dimensions in the model becomes very penalising. If we consider a model involving 50 dimensions, the complexity already reaches 10^{100} for a discretisation of only 100 elements per dimension. 10^{80} being the presumed number of particles in the whole universe, we will never be able to deal with such a quantity of degrees of freedom using computers; this is the so-called *curse of dimensionality*. Classical methods show their limits. Another strategy is required in order to deal with highly multidimensional models. We need a mathematical tool able to reduce the complexity of the model (Ammar and Chinesta, 2008).

A first solution would consist of using sparse grids methods, but this strategy fails for models defined in spaces whose dimensions are about 20 (Achdou & Pironneau, 2005). Another option to alleviate the curse of dimensionality is to consider a separated representation of the solution within the context of the Proper Generalised Decomposition (PGD). Many years ago, in the 80s, Ladeveze (1999) developed the so-called radial approximation as one of the key ingredients of the non-linear non-incremental LATIN solver. The radial approximation consists of approximating the solution by a space and time separated representation. For a space/time problem, each term of the decomposition is the product of a time function and a space function, so that the solution finally reads,

$$u(x,t) \approx \sum_{i=1}^{N} T_i(t) \cdot X_i(x) \tag{1}$$

where N is usually of some tens. Therefore, instead of solving P problems in space, as we would do with an incremental approach–where P would be the total number of time increments–we only need to solve about N space and time problems. The CPU time savings can be spectacular in some cases (Chinesta, Ammar, & Cueto, 2010). Moreover, in separated approximations, the complexity scales linearly with the number of dimensions and then the introduction of extra dimensions is much less penalising.

The parametric PGD includes also parameters as extra-coordinates and uses a separated representation for approximating the resulting multidimensional solution. If we consider that our solution depends on x, t but also Q parameters $(p^1, \ldots, p^Q)$, we can approximate it by a sum of modes, each mode being a product of functions. The separated representation–also known as finite sum decomposition–takes the form,

$$u(x,t,p^1,\ldots,p^Q) \approx \sum_{i=1}^{N} T_i(t) \cdot X_i(x) \cdot P_i^1(p^1) \cdots P_i^Q(p^Q) \tag{2}$$

The challenge is then to construct a separated representation of the solution without knowing a priori this latter. In other words, we wish to extract a reduced number of functions able to represent the whole space, time and parametric evolution of the unknown field. When employing PGD, the functions used to approximate the solution are built on the fly. In order to compute those functions, an alternated directions fixed-point algorithm is implemented in a Galerkin framework. The number of enrichments, or in other words the number of terms in the finite sum decomposition depends on the regularity of the problem. It can range from a few tens to some tens. For more details regarding both theoretical and practical aspects of the PGD, the interested reader can refer to (Ammar, Chinesta, Cueto, & Doblare, 2010; Chinesta et al., 2010; Nouy, 2010; Pruliere, Chinesta, & Ammar, 2010).

2. Principle of time parallelisation

Parallel computing has become a dominant paradigm in computer simulation. Clusters and multi-core processors have been widely developed during the past years. They offer a solution to speed up the computation by carrying out many calculations simultaneously. Large problems can then be split into smaller ones and treated "in parallel". For engineering applications, the common approach is to perform a cutting on the spatial domain, called domain decomposition. Each region is then treated separately using the same algorithm developed for the sequential approach, but the difficulty is then to ensure a transfer of information from one subdomain to the other, and several continuity issues have to be taken into account at the interfaces.

The approach that we propose in this article consists of dividing the global problem into several local problems. By "global" we refer to the problem on the whole time domain whereas the adjective "local" is used when the problem is only solved on a time interval. In other words, we want to split the time domain into several subdomains and perform in parallel the computation of the solution on each time interval using multi-core processors or a platform of processors, as depicted in Figure 1. At first sight, it is quite difficult or at least unconventional to imagine that we could solve a transient problem, by computing what happens in a time interval without knowing the beginning. Several questions and challenges arise.

Firstly, *how can I solve a transient problem on an interval if I do not know the initial condition to be used?* Since we compute concurrently the local solutions, we have no idea of the initial field of temperature on those intervals, except for the first one where the initial condition corresponds actually to the initial condition for the global problem. Secondly, once I know the solution on each interval, *which strategy should I use to obtain the solution on the entire time domain?*

Actually time parallelisation has been and is still the subject of an intensive research work. To our knowledge, the first promising approach is the so-called "para-réelle" approach (Lions, Maday, & Turinici, 2001). The main idea behind this strategy is the use of a trial initial condition on each time subdomain. The solution that jumps at the interfaces are used to update the prescribed initial conditions. The iterative process stops when the discontinuities across the time intervals boundaries are small enough according to a given convergence criterion. An alternative to this method was proposed in 2008 and developed in a first place for linear problems. This approach does not involve neither an iteration scheme nor the necessity of introducing a convergence criterion and is based on the use of the Proper Orthogonal Decomposition (POD) (Ryckelynck, Chinesta, Cueto, & Ammar, 2006). A reduced basis of functions is extracted by applying the POD to a pre-computed solution and is used to approx-

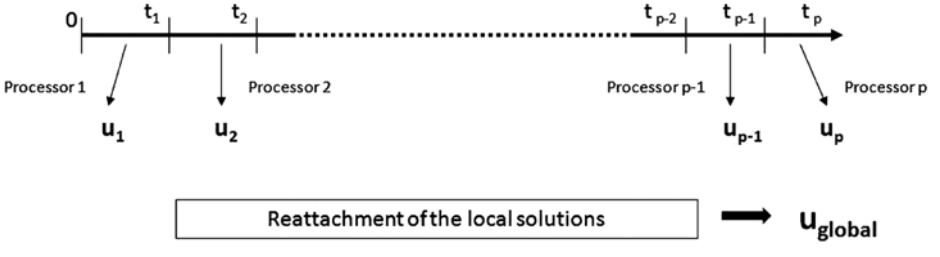


Figure 1. Principle of time parallelisation.

imate the solution and then the initial condition on each time interval. More details concerning this method are given in Chinesta, Ammar, Lemarchand, Beauchene, and Boust (2008).

The present work proposes a strategy based on a parametrisation of the initial condition. As mentioned previously, the main issue we have to tackle can be reformulated as follows: *How could one bypass the dependency of a transient problem on its initial condition?* We cannot solve the problem without specifying an initial condition because we would obtain an ill-posed problem. But, what if we could compute a solution valid for any initial condition? This is the key point of the method we expose here. Indeed, if for a given time-dependent problem, the initial condition is no longer treated in a classical way, but parametrised and considered as an extra-coordinate of the model, then the parallel time integration becomes possible.

3. Introducing the initial condition as an extra-coordinate of the model

The initial condition for a space/time problem being a field, a preliminary work has to be done: i.e. we have first to choose a set of independent parameters which are representative of the initial condition. These parameters can then be introduced as extra-coordinates of the model. The first idea would be to use the nodal values corresponding to the classical linear finite element approximation of the initial condition on the space mesh. But, this choice is not reasonable because the mesh used for the computation has to be fine enough to ensure a proper accuracy for the model. Thus, parametrising the initial condition through those nodal values would lead to the addition of some hundreds or even thousands of new coordinates; the dimensionality of the model would literally blow up! Even if the PGD is very good at dealing with highly multidimensional models, the associated computational cost would become unacceptable.

But actually, in many cases involving quite smooth solutions, the initial condition can be spatially represented by a reduced number of parameters, a few tens in general. Therefore, the strategy we propose consists of building up an auxiliary mesh much coarser than the one we use for the solution of the problem. The latter is only necessary to define the initial condition. Considering again a linear finite element approximation, the initial condition reads,

$$u_0(x) \approx \sum_{j=1}^{N_c} p^j \cdot \varphi_c^j(x) \tag{3}$$

where N_c is the number of nodes of the coarse mesh and $\left\{\varphi_c^j\right\}_{j=1}^{N_c}$ are the shape functions satisfying the delta Kronecker property defined on the coarse mesh. Finally, the introduction of this second mesh allows us to parametrise the initial field with a small number of scalars.

Thanks to the PGD, it is then possible to construct a separated representation of the solution by computing each mode on the fly. The strategy we employ is relatively simple. In order to calculate a solution valid for any initial condition or at least for a large range of configurations, we consider that each nodal value p^j has a domain of variability $\Omega_p = [p_{\min}, p_{\max}]$ where the bounds $p_{\min}$ and $p_{\max}$ are prescribed. For the sake of simplicity and without loss of generality, we consider that this domain is identical for all parameters p^j. Thus, the solution is sought under the following separated form,

$$u\left(x,t,p^1,\ldots,p^{N_c}\right) \approx \sum_{i=1}^{i=N} X_i(x) \cdot T_i(t) \cdot P_i^1\left(p^1\right) \cdots P_i^{N_c}\left(p^{N_c}\right) \tag{4}$$

where $x \in \Omega_x$, $t \in \Omega_t$ and $p^j \in \Omega_p$ for $j \in [1,\ldots,N_c]$.

Choosing the bounds $p_{\min}$ and $p_{\max}$ is not always straightforward. When the physics of the problem is known, it is not too difficult to define a lower and an upper bound. The choice is based on physical criteria or energy balance. In other cases, one could imagine that the length of this interval is another coordinate of the model, living in a domain large enough. Obviously, the accuracy and the convergence of the model is not affected by the values chosen for the bounds. On the other hand, the discretisation in the interval $[p_{\min}, p_{\max}]$ does have an impact on the computational cost, since a finer resolution leads to an increase of the complexity – that is to say the number of degrees of freedom – of the model. Nevertheless when the variation is linear, two points suffice to represent any intermediate configuration.

After presenting a method to introduce the initial condition as an extra-coordinate of the model, essential preliminary step to the progression toward our final objective, we detail in the next section two approaches for parallelising the computation of the solution of the 1D heat transfer equation. Issues related to the reattachment of the local solutions are also addressed and we propose methods for reducing the error and for guaranteeing a good accuracy.

4. Time parallelisation for heat transfer modelling

4.1. A general approach

Throughout this article we focus on a unique problem, the transient heat transfer. Considering a spatial domain $\Omega_x = [0, L]$ and a time domain $\Omega_t = [0, t_{\max}]$, the equation to be solved writes,

$$\begin{aligned} &\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = f(x,t) \quad \text{for} \quad (x,t) \in \Omega_x \times \Omega_t \\ &u(x,0) = u_0(x) \\ &u(0,t) = u_{\mathrm{a}} \quad u(L,t) = u_{\mathrm{b}} \end{aligned} \tag{5}$$

Let us imagine that we use a double core processor and we would like to take advantage of this architecture in order to save computing time. The principle of the method consists of splitting the time domain into two intervals, not necessarily of the same length, that we denote $I_1 = [0, t_1]$ and $I_2 = [t_1, t_{\max}]$. For each interval, we compute a multidimensional solution including the initial condition as an extra-coordinate. Each solution is sought under a separated form and finally we obtain a couple of solutions $u_1\left(x \in \Omega_x, t \in I_1, p^1 \in \Omega_{\mathrm{p}}, \ldots, p^{N_{\mathrm{c}}} \in \Omega_{\mathrm{p}}\right)$ and $u_2\left(x \in \Omega_x, t \in I_2, p^1 \in \Omega_{\mathrm{p}}, \ldots, p^{N_{\mathrm{c}}} \in \Omega_{\mathrm{p}}\right)$.

Let us remark here that the only reason why we have to solve the equation on both intervals is because in general the source term is time-dependent. If the latter is constant in time, then we only need to solve the equation on a unique interval and this local solution is sufficient to reconstruct the solution on the entire time domain.

The advantage of this method is that we can compute all the local solutions off-line. The only information we need is the expression of the source term. Then, the remaining part of the work only consists of specifying an initial condition (and boundary conditions) for the global problem, particularising and finally reattaching the local solutions. This work is done online and appears to be very fast compared to an incremental method especially for large problems. The associated computing time only depends on the number of intervals we consider, but is independent of the discretisation in space.

Nevertheless, this parallelisation approach does not offer a lot of flexibility. The source term cannot be modified during the simulation. It is a static input data and this becomes

problematic in a real time framework. Indeed, if we want our simulation to be able to incorporate data provided in real-time by measurement devices like sensors (Gonzalez et al., in press), then this approach is clearly not suited even if it is fast enough. Let us imagine that the source term evolves in a different way than expected – malfunction of a heating device for instance – then all the off-line work is not usable because it is based on the a priori knowledge of this term. The computation has to be run again on the time subdomains where the source term should be updated. Finally, the original gain in time provided by this method is completely lost.

Thereby, we would rather use a method allowing us to modify the source term on the fly if necessary. The idea is again to split the work into two parts. The first one, performed off-line, should not involve the source term. The second one, done on-line, should be fast enough to reach realtime. We propose a different approach called *partial parallelisation* to distinguish it from the *full parallelisation* we presented up to now. This latter is based on a decomposition and only applies to linear problems.

4.2. *Decomposition: a more flexible option*

The solution of a linear problem can be obtained from the solution of the homogeneous problem (h) and a particular solution of the complete one (p). It corresponds to the sum of both solutions, that is to say $u = u^{\mathrm{h}} + u^{\mathrm{p}}$. Since the homogeneous problem does not involve any time depending source term, we can solve it off-line once and for all on a unique time interval using the PGD method. This multidimensional solution stays valid for all the other subdomains.

The computation of the remaining part of the solution, that is to say the particular solution of the complete problem, is parallelised. Each local particular problem is solved online, concurrently on different processors using classical discretisation methods. Now let us consider that the source term has to be modified in realtime. With the decomposition strategy, we just have to identify on which processors the particular problem with the new source term has to be solved and to compute a new solution online. Even if this leads to an increase of the computing time, this latter is not very penalising compared to the previous approach. We only have to solve a new 1D problem, which is quite cheap if the problem is not too large, and the work that has been done off-line, that is to say the homogeneous solution, is still usable. For all these reasons, the decomposition approach seems to be better or at least much more flexible for parallel time integration of linear models.

Regardless of the method we choose, the essential step consists of merging those local solutions – from a temporal point of view – into a unique global solution on the entire time domain. Our capacity to reattach accurately the local solutions is determinant for the reliability of the model. The difficulty comes here from the fact that we introduced two discretisations in space. A fine one for the calculation and a coarse one only necessary to parametrise the initial field of temperature. In order to ensure the continuity of the global solution at the interface, both local solutions must match. The following equality should be fulfilled,

$$u_1(x, t = t_1) = u_2(x, t = t_1) \tag{6}$$

The profile of temperature computed at the end of interval I_1 has to be used as initial temperature profile for interval I_2. In other words, we will particularise the general solution u_2 by fixing the value of parameters $p^1, \ldots, p^{N_c}$ for the initial condition to fit at the best the temperatures obtained at the end of interval I_1. To do so, a projection from the fine to the coarse mesh is required in order to calculate the adequate nodal values to be used for the initial condition of the local problem defined on interval I_2.

4.3. From local to global solution: reattachment strategies

Throughout this section, we will use index c to refer to the auxiliary coarse mesh. As we mentioned previously, the work to be done to build the global solution from the local parts computed concurrently is a projection from one mesh to the other. We consider here only two time intervals but this can be extended to n intervals. For the sake of clarity, we denote u the solution $u_1(x, t = t_1)$ at the end of interval I_1 and u_c the solution $u_2(x, t = t_1)$ at the beginning of interval I_2. We would like those two solutions to be as close as possible.

4.3.1. Projection at the interface

In order to enforce this continuity condition, we use a least squares approach. Let us consider a functional θ: $E \rightarrow \Re$ where $E = \text{span}\left\{\varphi_c^i\right\}_{i=1}^{N_c}$ such that,

$$\forall\ u_c \in E,\ \theta(u_c) = \int_{\Omega_x} (u_c - u)^2 \mathrm{d}x \tag{7}$$

The minimisation of this functional leads to the solution of a linear system of N_c equations, we are not going to detail here, where N_c is the number of nodes of the auxiliary coarse mesh.

The least squares method provides the best choice, according to the L_2 norm, for the nodal values to be used for the initial condition on the second interval. Once we know those values, the method to obtain the global solution is relatively simple. On the first interval I_1, the values of the parameters $\{p^j\}_{j=1}^{N_c}$ are fixed by the initial condition of the global problem. For the second interval, the general solution computed off-line is particularised using the values $\left\{u_c^i\right\}_{i=1}^{N_c}$ resulting from the projection based on the least squares method.

Nevertheless, a reproach can be formulated against this kind of approach. The least squares method is purely mathematical. No physical consideration is taken into account. When dealing with a thermal problem, one can be interested in ensuring the conservation of internal energy or the conservation of heat flux at the interface. The method presented previously should be completed with a mathematical tool which is able to represent the physics of the phenomenon. We could introduce some constraints in the classical least squares formulation and obtain a different minimisation problem that could be solved using Lagrange multipliers.

But in any case, as soon as the number of nodes of the calculation and the auxiliary mesh is different – making the assumption that the nodes are equally spaced – the projection of the local solution at the interface leads inevitably to the introduction of an error. The control and the reduction of this error is essential to ensure a proper accuracy for the solution. One alternative is to refine the coarse mesh: the higher the number of nodes, the better the approximation of the solution at the interface. But we have to keep in mind that the addition of a node in the coarse mesh results in the introduction of a new coordinate p^j in the model. Therefore, the complexity of the model and the required computing time and resources to solve it increase. An adaptative refinement of the coarse mesh is an interesting alternative, in the sense that it is localised; the mesh is refined in the region of important error, and in reverse, coarsened in other regions if possible. The implementation of this method is a work in progress.

The alternative we are going to present here is based on the following question: *How could one reduce the error at the interface without modifying the number of parameters for*

the initial condition? One possible solution to this issue is partly based on the decaying character of the heat transfer equation.

4.3.2. Decaying of the error and overlapping

Because the equation we are solving is parabolic, we know that the initial condition has an impact on the solution essentially during a short period. Thus, in the absence of a source term, the initial condition decays exponentially. The characteristic time Δt_{d} associated to this decaying corresponds actually to the characteristic time for the diffusion of the heat on the auxiliary coarse mesh used to define the initial field of temperature. This time is given by $\Delta t_{\mathrm{d}} \approx \frac{\Delta x^2}{k}$ where Δx is the characteristic size of the coarse mesh and k is the thermal conductivity of the material. This is a very nice property from which we can take advantage when defining our reattachment strategy. Indeed, since we know that even if we introduce an error on the initial condition during the projection step, this latter will be quickly "forgotten", the projection should not be performed at the interface anymore but a certain time before. Proceeding in this manner, we know that the jump from one solution to the other and therefore the error at the interface will be much smaller.

The method we employ consists of computing solutions on each time interval independently. The only difference is that now the subdomains are not distinct anymore; they overlap in a region of length Δt. Consequently, a part of the local solutions computed upstream is never used in the global solution. For instance on interval I_1, the projection is performed at $t = t_1 - \Delta t$ but the local solution u_2 on interval I_2 is actually incorporated into the global solution at $t = t_1$. At first sight, this method seems to be less efficient in the sense that the computed information is not entirely used. But actually it proves to be much more powerful than the classical parallelisation approach as we demonstrate in the last section of this paper.

The "natural" overlapping presented up to now is based on the decaying behaviour of the solution. Instead of taking only advantage of this characteristic, we can ensure an optimal reattachment of the solutions at the interface. Previously, the reattachment was performed by using a least squares method at time $t_{\mathrm{projection}} = t_{\mathrm{interface}} - \Delta t_{\mathrm{d}}$. The new approach we discuss now consists of determining which values should be used for the initial condition at $t_{\mathrm{projection}}$ in order to optimise the reattachment at $t_{\mathrm{interface}}$.

To formulate this problem in a proper way, we consider again only two time subdomains $I_1 = [0, t_1]$ and $I_2 = [t_1 - \Delta t_{\mathrm{d}}, t_{\max}]$. Multidimensional solutions u_1 and u_2 have been computed and have to be reattached. Let us introduce the function L defined as,

$$L(p^1, \ldots, p^{N_{\mathrm{c}}}) = \int_{\Omega_x} \left(u_1(x, t_1) - u_2\left(x, t_1, p^1, \ldots, p^{N_{\mathrm{c}}}\right)\right)^2 \mathrm{d}x \tag{8}$$

Our objective is to find the values $\{p_i\}_{i=1}^{N_{\mathrm{c}}}$ minimising $L(p^1, \ldots, p^{N_{\mathrm{c}}})$. After replacing u_2 by its separated representation, we obtain a non-linear minimisation problem. Among several available options for the solution of this kind of minimisation problem, we compared two methods in our numerical experiments: the Matlab routine *fminsearch* and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method.

The *fminsearch* routine uses a simplex search method and proves to be an efficient tool for solving non-linear minimisation problems. The BFGS method is also particularly well suited to the solution of unconstrained optimisation problems. This is a quasi-Newton's method involving the Hessian matrix. However, the latter is not evaluated directly but an approxima-

tion is preferred, and in that sense, it constitutes a generalisation of the secant method. The corresponding algorithm is quite complex but the gain in accuracy is significant.

The optimisation of the overlapping results in a gain on the minimum size of the subdomains to be considered and consequently on the speedup provided by the parallel time integration while ensuring at the same time its primary function – to reduce discontinuities at the interfaces. Indeed thanks to this method, we are not restricted by the characteristic time for the diffusion of the error anymore; the time interval can now be smaller than $2\Delta t_{\mathrm{d}}$. Nevertheless, as we show in the last section devoted to numerical experiments, there exists a minimum size for the overlapping interval below which we are not able to optimise the reattachment.

5. Numerical results

Time parallelisation makes sense for problems involving several characteristic times. For instance, when studying the ageing of a composite material caused by oxidation, the chemical reaction whose characteristic time is relatively small (some milliseconds) has to be simulated for a long period (several months or years). Another example concerning composite materials is ultrasonic welding. This manufacturing process consists of welding two pieces by applying a sinusoidal force of frequency 20 kHz during a few seconds. If we try to solve those kind of problems with an incremental approach, the required number of time steps is very large and so is the associated computation time.

In order to highlight the advantages of using a parallel approach, we solve an academic problem consisting of the 1D heat transfer equation with a source term that exhibits a slow time variation with an over-imposed high frequency sinusoidal variation,

$$f(x,t) = 10 \cdot \left(1.05 - \exp\left(\frac{-t}{3}\right) + 0.05\sin(100\pi t)\right) \cdot (x \cdot (1-x)) \tag{9}$$

The two characteristic times involved here are $\tau_{\mathrm{slow}} = 3$ and $\tau_{\mathrm{fast}} = \frac{2\pi}{\omega} = \frac{2\pi}{100\pi} = 0.02$. In order to capture the high-frequency variations, the time step is chosen equal to $\Delta t = \frac{\tau_{\mathrm{fast}}}{10} = 0.002$. The total duration of the simulation is $t_{\max} = 5\tau_{\mathrm{slow}} = 15$. With an incremental method, the solution of this problem would require 7500 iterations, that is to say the solution of 7500 1D problems of size N_n.

If the size of the problem is relatively small, then the parallel approach cannot compete with the incremental methods because the reattachment of the local solutions is too time-consuming. But as soon as we increase the complexity of the model, the interest for the parallelisation grows.

5.1. *Influence of the number of subdomains*

The length of the 1D domain is chosen equal to 1 and the conductivity of the material we are modelling is 0.1. The initial field of temperature is parametrised with five nodal values, and therefore the characteristic size of the coarse mesh is 0.25. A priori, we know that the minimum size for a time interval is $\Delta t = \Delta t_{\mathrm{d}} = \frac{0.25^2}{0.1} = 0.625$ and thus the maximum number of subdomains is about 25. After computing a solution of the problem using a standard incremental discretisation technique, we compare the latter chosen as reference solution with the temperature field that we obtain with a parallel approach. The quantity of interest is the error associated to the L_2 norm at each time step, which is defined at time t as follows,

$$\text{Error}(t) = \frac{\int_{\Omega_x} (u_{\text{ref}}(x,t) - u_{\text{PGD}}(x,t))^2 \mathrm{d}x}{\int_{\Omega_x} u_{\text{ref}}(x,t)^2 \mathrm{d}x} \tag{10}$$

where u_{ref} is the reference solution and u_{PGD} is the global solution obtained from the reattachment of the solutions computed on the time subdomains.

Figure 2 depicts the error for different number of subdomains. As expected, if the size of the subdomain is large compared to the characteristic time, the error introduced by the projection at the interface vanishes before the end of the subdomain, Figure 2(a). The evolution of the error is similar on each time interval; an important jump at the interface followed by an exponential decaying. The error does not cumulate from one subdomain to the other; the maximum error is constant and equal in this case to about 2.5%. On the contrary, as soon as we consider intervals smaller than the critical value Δt_{d}, the error starts increasing and reaches more than 4%, as we can see in Figure 2(b).

5.2. *Overlapping*

In Figure 3(a), we can see how fast the error decreases as soon as overlapping applies. In order to emphasise the notion of decaying error, we purposely scaled the overlapping length with respect to the characteristic time. The main result we obtain here is the existence of a threshold value for the overlapping length corresponding to Δt_{d}. If we extend this length beyond this value, the resulting gain in accuracy is meaningless whereas the efficiency of the method is reduced; the quantity of unused computed information increases with the size of the overlapping interval. Moreover, the overlapping strategy results in a smoothing of the discontinuities at the interface as pointed out in Figure 3(b).

"Natural" overlapping is clearly more constraining regarding the number of subdomains. Indeed if we want to reduce significantly the error at the interface, one should consider an overlapping region of a length of about Δt_{d}, which means that a time interval should be at least of length $2\Delta t_{\text{d}}$. Thanks to an optimisation strategy previously described, it is possible to reduce the size of the overlapping region while keeping an equivalent quality for the reattachment at the interface.

As depicted in Figure 4, both optimisation methods, fminsearch and BFGS, are very efficient even for small overlapping length. They offer very similar results; an overlapping length corresponding to about 25% of the characteristic decaying time is sufficient. And, applying

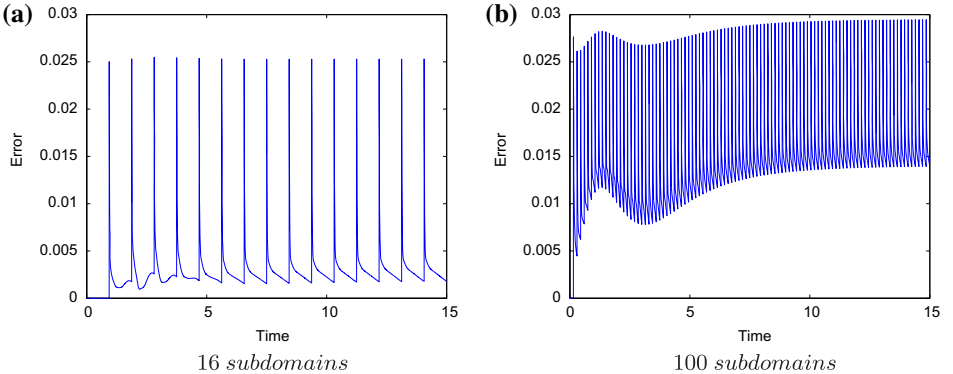
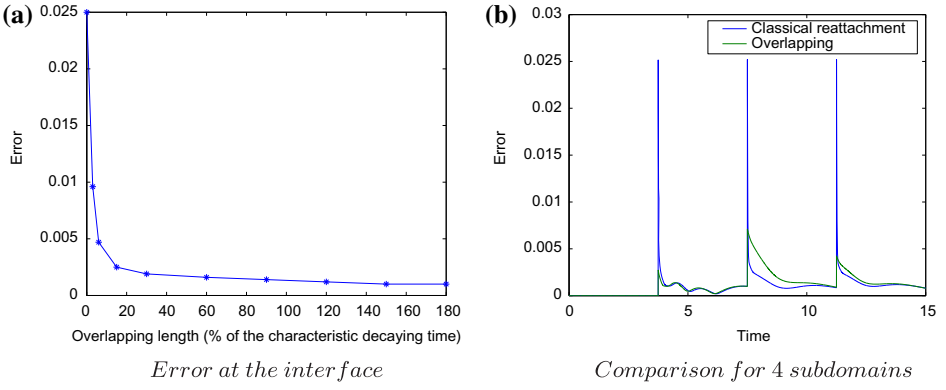


Figure 2. Error evolution vs. the number of subdomains.



Error at the interface *Comparison for 4 subdomains*

Figure 3. Improvement of the reattachment with a natural overlapping.

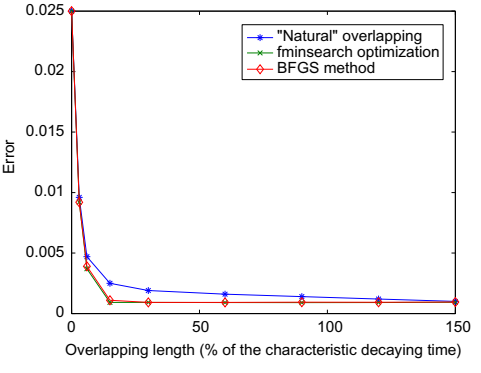


Figure 4. Optimisation of the reattachment.

the same rule which consists of restricting the overlapping length to 50% of the total length of the time interval, the minimum length we can afford for a subdomain appears to be $\frac{\Delta t_d}{2}$. Compared to a natural overlapping, the size of the subdomain is therefore divided by 4 or equivalently the maximum number of subdomains is multiplied by 4. This is particularly important from a speedup point of view. If the online workload can be distributed among a higher number of processors, the resulting gain in computation time will be more significant and the parallel time integration more interesting.

6. Conclusion

The PGD entails new abilities regarding numerical simulations. The uncertainty concerning the value of a parameter is no more a brake to the construction of a model since it can be included as an extra-coordinate. The separated representation of the solution is a way to circumvent the curse of dimensionality one faces when dealing with models defined in spaces of high dimension. Henceforth, a parallel approach can be considered for the solution of a transient problem, like the heat transfer problem. This requires a parametrisation of the initial condition which is done thanks to the construction of an auxiliary coarse mesh. A decomposition of the problem turns out to be the most adaptable option and represents a very promising

alternative especially for future developments on real-time simulation. The error due to the reattachment of the local solutions can be significantly reduced by implementing an optimised overlapping strategy. This opens new perspectives regarding the solution of problems involving multiple characteristic times. Results concerning the speedup have not been exposed explicitly but will be the subject of future publications.

References

Achdou, Y., & Pironneau, O. (2005). *Computational methods for option pricing*. SIAM series in Applied Math, Philadelphia.

Ammar, A., & Chinesta, F. (2008). Circumventing curse of dimensionality in the solution of highly multidimensional models encountered in quantum mechanics using meshfree finite sums decomposition. *Lecture Notes Computational Science Engineering, 65*, 1–17.

Ammar, A., Chinesta, F., Cueto, E., & Doblare, M. (2010). Proper Generalized Decomposition of time-multiscale models. *International Journal for Numerical Methods in Engineering, 83*, 1114–1132.

Chinesta, F., Ammar, A., & Cueto, E. (2010). Recent advances and new challenges in the use of the Proper Generalized Decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering, 17*(4), 327–350.

Chinesta, F., Ammar, A., Lemarchand, F., Beauchene, P., & Boust, F. (2008). Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. *Computer Methods in Applied Mechanics and Engineering, 197*, 400–413.

Gonzalez, D., Masson, F., Poulhaon, F., Leygue, A., Cueto, E., & Chinesta, F. (in press). Proper Generalized Decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation*.

Ladeveze, P. (1999). *Nonlinear computational structural mechanics*. New York: Springer.

Lions, J.-L., Maday, Y., & Turinici, G. (2001). Résolution d'EDP par un schéma en temps "para réelle". *Comptes Rendus Académie des Sciences, 332*, 661–668.

Nouy, A. (2010). A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering, 199*, 1603–1626.

Pruliere, E., Chinesta, F., & Ammar, A. (2010). On the deterministic solution of multidimensionnal parametric models using Proper Generalized Decomposition. *Mathematics and Computers in Simulation, 81*, 791–810.

Ryckelynck, D., Chinesta, F., Cueto, E., & Ammar, A. (2006). On the "a priori"model reduction: Overview and recent developments. *Archives of Computational Methods in Engineering, State of the Art Reviews, 13*, 91–128.