Computational Mechanics manuscript No.
(will be inserted by the editor)

A model reduction technique based on the PGD for elastic-viscoplastic computational analysis

N. Relun · D. Néron · P.A. Boucard

Received: date / Accepted: date

Abstract In this paper a model reduction approach for elastic-viscoplastic evolution problems is considered. Enhancement of the PGD reduced model by a new iterative technique involving only elastic problems is investigated and allows to reduce CPU cost. The accuracy of the solution and convergence properties are tested on an academic example and a calculation time comparison with the commercial finite element code Abaqus is presented in the case of an industrial structure.

Keywords PGD · LATIN method · elastic-viscoplastic · model reduction

1 Introduction

Numerical simulation has been playing an increasingly important role in science and engineering. However, when dealing with high-fidelity models, the number of degrees of freedom can lead to systems so large that direct techniques are inapplicable. Model reduction techniques constitute an efficient way to circumvent this difficulty by seeking the solution of a problem in a reduced-order basis (ROB), whose dimension is much lower than the original vector space. A posteriori methods usually consist in defining this ROB by the decomposition of the solution of a surrogate model relevant to the initial model (see e.g. [2,12,17,10,3]). A priori methods follow a different path by building progressively an approximate separated representation of the solution, without assuming any basis (see e.g. [13,1,23, 15,19,6,25,16,7]).

This work focuses on the Proper Generalized Decomposition (PGD) which belongs to the second family and is used herein to solve elastic-viscoplastic evolution problems defined over the time-space domain. This type of problems, especially when they involve a large number of time steps and degrees of freedom are particularly CPU expensive [18]. To make these computations affordable, the PGD, originally introduced as the "radial loading approximation" in [13], consists in seeking a separated time-space representation of the unknowns and the iterative LATIN method is used to generate the approximation by successive enrichments [14]. At a particular iteration, the ROB which has been already formed is first used to compute a reduced-order model (ROM) and find a new approximation of the solution. If the quality of this approximation is not sufficient, the ROB is enriched by determining a new functional product using a greedy algorithm. The PGD has been applied for solving many types of problems in the context of the LATIN method and allowed to decrease the CPU cost drastically. Elastic-viscoplastic problems have already been solved with this method in [8,9] but, since these works, more efficient algorithms have been introduced to built the PGD approximation. These algorithms (a review can be found in [21]) have only been applied to visco-elasticity [22,15] and this paper demonstrates their use in the elastic-viscoplastic case.

However, model reduction techniques are more particularly efficient when the ROM needs to be constructed only once, which is not the case when dealing with nonlinear aspects. In that case, the various operators must be updated along the iterations and the calculation of the ROM and its inversion represents a large part of the global CPU cost. This issue has been highlighted

P.A. Boucard
LMT-Cachan
(ENS Cachan / CNRS / UPMC / PRES UniverSud Paris)
61 avenue du Président Wilson, F-94235 Cachan Cedex, France
Tel.: +0033-147402233
Fax: +0033-147402785
E-mail: boucard@lmt.ens-cachan.fr

and some dedicated techniques proposed in the context of a posteriori methods (eg. [20,5,4]).

This work focusses on a new technique which allows, in the context of the PGD method, to avoid the previous drawback by dealing only with the inversion of the equilibrium equation, instead of the inversion of the operator corresponding to the ROM. Taking advantage of the time independence of the equilibrium equation, the associated discrete problem can be assembled and factorized only one time, which leads to significant decrease in CPU cost.

The present paper is organized as follows. The reference problem to be solved, the LATIN algorithm and the introduction of the PGD for elastic-viscoplastic constitutive formulation are presented in Section 2. Section 3 is focused on the new algorithm to determined the PGD. A gain of 4 in term of CPU time is estimated by the introduction of the new algorithm. In Section 4, some properties of this new reduced model are studied. The quality of the PGD obtained along LATIN iterations is compared with a direct PGD of the solution which is obtained at the convergence of the algorithm, showing that this decomposition is of good quality. The influence of the mesh refinement and the number of time steps on the number of pairs in the decomposition is shown. Our conclusion is that the PGD is linked to the problem to be solved, not on the discretization, as soon as the discretization is accurate enough to capture all the physics. A comparison of the results and CPU times obtained with our research software and the industrial code Abaqus is then presented, showing a gain of 30% in favor of the method developed herein.

2 The reference problem and the LATIN method

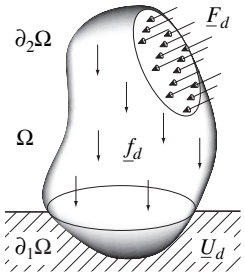

Fig. 1 The reference problem in domain Ω

The reference problem is a quasi-static isothermal evolution of a structure defined over the time-space domain $\mathcal{I}\times\Omega$, with $\mathcal{I}=[0,T]$ the interval of study, assuming small perturbations. The structure is subjected to prescribed body forces $\underline{f}_d$, to traction forces $\underline{F}_d$ over a part $\partial_2\Omega$ of the boundary, and to prescribed displacements $\underline{U}_d$ over the complementary part $\partial_1\Omega$ (see Fig. 1).

The state of the structure is defined by the set of fields $\mathbf{s}(\boldsymbol{\sigma},\dot{\boldsymbol{\varepsilon}}_p)$ (where the upper point denotes the derivative with respect to time), in which:

- $\boldsymbol{\varepsilon}_p$ designates the inelastic part of the strain field $\boldsymbol{\varepsilon}$ corresponding to displacement field $\underline{U}$, with the classic additive decomposition into an elastic part $\boldsymbol{\varepsilon}_e$ and an inelastic part $\boldsymbol{\varepsilon}_p=\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}_e$;
- $\boldsymbol{\sigma}$ designates the Cauchy stress field.

All these quantities are defined over the time-space domain $\mathcal{I}\times\Omega$ and assumed to be sufficiently regular.

2.1 Constitutive formulation

To fix the idea, we consider herein the case of a Chaboche's type elastic-visco-plastic formulation. The state equation of this formulation is

$$\boldsymbol{\sigma}=\mathbf{H}\boldsymbol{\varepsilon}_e \tag{1}$$

where $\mathbf{H}$ is the Hooke's operator, and the evolution equation is

$$\dot{\boldsymbol{\varepsilon}}_p=\dot{p}\,\frac{3/2\,\boldsymbol{\sigma}^D}{J_2(\boldsymbol{\sigma})} \tag{2}$$

where $\dot{p}=\langle(J_2(\boldsymbol{\sigma})-\sigma_y)/K\rangle_+^n$, with K, n and σ_y material dependent scalars. Superscript D denotes the deviatoric part of the tensor, and $J_2(\boldsymbol{\sigma})$ the Von-Mises equivalent stress. The evolution equation is nonlinear and we formally introduce operator $\mathbf{B}$ such as $\dot{\boldsymbol{\varepsilon}}_p=\mathbf{B}(\boldsymbol{\sigma})$. Let us define the space $\boldsymbol{\Gamma}$ of fields $\mathbf{s}$ such that (2) is fulfilled.

2.2 Admissibility conditions

Let us introduce the following spaces and the corresponding vector spaces (with superscript $\star$):

- the space $\mathcal{U}$ of the kinematically admissible fields $\underline{U}$ such that

$$\underline{U}_{|t=0}=\underline{U}_0 \text{ and } \underline{U}=\underline{U}_d \text{ on } \partial_1\Omega \tag{3}$$

- the space $\boldsymbol{\mathcal{F}}$ of the statically admissible fields $\boldsymbol{\sigma}$ such that $\forall\underline{U}^\star\in\mathcal{U}^\star$

$$\begin{aligned}&-\int_{\mathcal{I}\times\Omega}\boldsymbol{\sigma}:\boldsymbol{\varepsilon}(\underline{U}^\star)\,\mathrm{d}\Omega\,\mathrm{d}t\\&+\int_{\mathcal{I}\times\Omega}\underline{f}_d\cdot\underline{U}^\star\,\mathrm{d}\Omega\,\mathrm{d}t+\int_{\mathcal{I}\times\partial_2\Omega}\underline{F}_d\cdot\underline{U}^\star\,\mathrm{d}S\,\mathrm{d}t=0\end{aligned} \tag{4}$$

– the space $\boldsymbol{\mathcal{E}}$ of the kinematically admissible fields $\boldsymbol{\varepsilon}$ such that $\exists \underline{U} \in \mathcal{U}, \quad \boldsymbol{\varepsilon} = \nabla_{sym}\underline{U}$ that is, in a weak form,

$$\forall \boldsymbol{\sigma} \in \boldsymbol{\mathcal{F}}^\star,$$
$$-\int_{\mathcal{I}\times\Omega} \boldsymbol{\sigma}^\star : \boldsymbol{\varepsilon} \ \mathrm{d}\Omega\,\mathrm{d}t + \int_{\mathcal{I}\times\partial_1\Omega} \boldsymbol{\sigma}^\star \underline{n} \cdot \underline{U}_d = 0 \quad (5)$$

– the space $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$ of the admissible fields $\mathbf{s}$ such that $(\mathbf{H}^{-1}\boldsymbol{\sigma} + \boldsymbol{\varepsilon}_p) \in \boldsymbol{\mathcal{E}}$ and $\boldsymbol{\sigma} \in \boldsymbol{\mathcal{F}}$

2.3 Brief overview of the LATIN method

Solving such a nonlinear problem arises both difficulties of satisfying the linear equilibrium of the structure and solving the local nonlinear equations of the material constitutive formulation. With the spaces introduced before, the difficulty is to find the solution field $\mathbf{s}$ such that $\mathbf{s} \in \boldsymbol{\mathcal{A}}_{\mathbf{d}} \cap \boldsymbol{\Gamma}$. Usual nonlinear solvers are incremental in time. The LATIN method is non-incremental. It is an iterative procedure which consists in finding alternatively an approximation of the solution over the entire time-space domain between the space $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$ (linear equations eventually global in space) and the space $\boldsymbol{\Gamma}$ (local equations eventually nonlinear). More details can be found in [14].

A local stage and a linear stage are defined in the method as follows:

– knowing a linear solution $\mathbf{s}$ in $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$, the local stage consist in finding a local solution $\hat{\mathbf{s}}$ in $\boldsymbol{\Gamma}$ such as a local search direction (6) is fulfilled,

$$\mathbf{G}(\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}) + (\hat{\dot{\boldsymbol{\varepsilon}}}_p - \dot{\boldsymbol{\varepsilon}}_p) = 0 \quad (6)$$

where $\mathbf{G}$ is an operator, parameter of the method;

– knowing a local solution $\hat{\mathbf{s}}$ in $\boldsymbol{\Gamma}$, the linear stage consist in finding a linear solution $\mathbf{s}$ in $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$ such as a linear search direction (7) is fulfilled,

$$\mathbf{A}(\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) - (\dot{\boldsymbol{\varepsilon}}_p - \hat{\dot{\boldsymbol{\varepsilon}}}_p) = 0 \quad (7)$$

where $\mathbf{A}$ is an operator, parameter of the method.

The choice of $\mathbf{G}$ and $\mathbf{A}$ will be discussed later.

An iteration of the LATIN method is made of a local stage and a linear stage. A simple initialization is an elastic solution, which is in $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$.
An indicator of the quality of the solution can be calculated at each iteration of the LATIN method as the distance between the local solution and the linear one

$$\eta_{LATIN} = \frac{\left\| \hat{\dot{\boldsymbol{\varepsilon}}}_p - \dot{\boldsymbol{\varepsilon}}_p \right\|}{\left\| \hat{\dot{\boldsymbol{\varepsilon}}}_p + \dot{\boldsymbol{\varepsilon}}_p \right\|} + \frac{\| \hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma} \|}{\| \hat{\boldsymbol{\sigma}} + \boldsymbol{\sigma} \|} \quad (8)$$

We made the following choice for the search directions, similar to the search directions in a Newton tangent algorithm:

– the local search direction (which can be seen as a local approximation of the $\boldsymbol{\mathcal{A}}_{\mathbf{d}}$ space) is defined by,

$$\mathbf{G} = \infty \quad \Rightarrow \quad \hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma} \quad (9)$$

– the linear search direction (which can be seen as a linear approximation of the $\boldsymbol{\Gamma}$ space) is defined at each time-space point $(t, \underline{M})$ by,

$$\mathbf{A}(t,\underline{M}) = \left.\frac{\partial \mathbf{B}}{\partial \boldsymbol{\sigma}}\right|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}} = \frac{\partial \dot{p}}{\partial \boldsymbol{\sigma}} \otimes \frac{3/2\,\boldsymbol{\sigma}^D}{J_2(\boldsymbol{\sigma})} + \dot{p} \cdot \frac{3/2\left(J_2(\boldsymbol{\sigma})\mathbb{I} - 3/2\frac{\boldsymbol{\sigma}^D \otimes \boldsymbol{\sigma}^D}{J_2(\boldsymbol{\sigma})}\right)}{J_2(\boldsymbol{\sigma})^2} \quad (10)$$

where $\frac{\partial \dot{p}}{\partial \boldsymbol{\sigma}} = n \left\langle (J_2(\boldsymbol{\sigma}) - \sigma_y)/K \right\rangle_+^{(n-1)} \frac{3/2\,\boldsymbol{\sigma}^D}{J_2(\boldsymbol{\sigma})}$. One can notice that this operator evolves along LATIN iterations.

2.4 PGD model reduction technique

At the linear stage of the LATIN method, a linear approximation of the elastic-viscoplastic problem is known. The PGD technique can be introduced easily in this context as the superposition principle is valid for linear problems.

The unknowns are searched as sums of products of time functions $\alpha_i(t)$ and space functions ($\mathbb{C}_i(\underline{M})$, $\mathbb{E}_{p,i}(\underline{M})$), for example with $m-1$ pairs,

$$\boldsymbol{\sigma}(t,\underline{M}) = \sum_{i=1}^{m-1} \alpha_i(t)\mathbb{C}_i(\underline{M}) \quad (11a)$$

$$\dot{\boldsymbol{\varepsilon}}_p(t,\underline{M}) = \sum_{i=1}^{m-1} \dot{\alpha}_i(t)\mathbb{E}_{p,i}(\underline{M}) \quad (11b)$$

The interested reader can refer to [21] for a review of the different algorithms to solve a linear problem with the PGD. A progressive algorithm with update is considered in our case, based on the minimization of a constitutive error. The linear search direction (7) is used to define the error.

Let first consider the case of the update of the time functions, $m-1$ space functions ($\mathbb{C}_i$,$\mathbb{E}_{p,i}$) and time functions α_i being known, which corresponds to seek a better approximation of the solution in the ROB which has

been already built,

$$\boldsymbol{\sigma} = \sum_{i=1}^{m-1} (\alpha_i + \Delta\alpha_i)\mathbb{C}_i \tag{12a}$$

$$\dot{\boldsymbol{\varepsilon}}_p = \sum_{i=1}^{m-1} (\dot{\alpha}_i + \Delta\dot{\alpha}_i)\mathbb{E}_{p,i} \tag{12b}$$

Writing (7) as a minimization problem, one seeks $\{\Delta\alpha_i\}_{i=1}^{m-1}$ such that,

$$\{\Delta\alpha_i\}_{i=1}^{m-1} = \arg \min_{\{\Delta\overline{\alpha}_i\}_{i=1}^{m-1} \in \mathcal{H}_1^{m-1}} \left\| \mathbf{A} \sum_{i=1}^{m-1} \Delta\overline{\alpha}_i \mathbb{C}_i - \sum_{i=1}^{m-1} \Delta\dot{\overline{\alpha}}_i \mathbb{E}_{p,i} - \mathbb{\Delta}_1 \right\|_{\mathbf{M}} \tag{13}$$

where $\mathcal{H}_1$ is the space of continuous functions with continuous first derivative, $\mathbb{\Delta}_1 = \mathbf{A}\left(\hat{\boldsymbol{\sigma}} - \sum_{i=1}^{m-1} \alpha_i \mathbb{C}_i\right) - \left(\hat{\dot{\boldsymbol{\varepsilon}}}_p - \sum_{i=1}^{m-1} \dot{\alpha}_i \mathbb{E}_{p,i}\right)$ is known and $\|\square\|_{\mathbf{M}} = \int_{\mathcal{I}\times\Omega} \square : \mathbf{M}\square \, \mathrm{d}\Omega \, \mathrm{d}t$.

The operator $\mathbf{M}$, defining the norm, is chosen to be $\mathbf{A}^{-1}$ to ensure a good convergence rate of the PGD. The minimization problem (13) can be rewritten as a multivariables differential equation, which can be solved by using the algorithm presented in [15,24].

Knowing the correction of the time function, an indicator of the quality of the approximation with the PGD is defined,

$$\eta_{PGD} = \frac{\left\| \mathbf{A} \sum_{i=1}^{m-1} \Delta\alpha_i \mathbb{C}_i - \sum_{i=1}^{m-1} \Delta\dot{\alpha}_i \mathbb{E}_{p,i} - \mathbb{\Delta}_1 \right\|_{\mathbf{M}}}{\|\mathbb{\Delta}_1\|_{\mathbf{M}}} \tag{14}$$

If this indicator exceeds a threshold value after the update of the time functions (chosen equal to 0.95 after our numerical studies) a new pair of functions is added.

The addition of a new pair $(\alpha_m, \mathbb{C}_m, \mathbb{E}_{p,m})$ corresponds to an enrichment phase of the ROB,

$$\boldsymbol{\sigma} = \sum_{i=1}^{m-1} (\alpha_i + \Delta\alpha_i)\mathbb{C}_i + \alpha_m \mathbb{C}_m \tag{15a}$$

$$\dot{\boldsymbol{\varepsilon}}_p = \sum_{i=1}^{m-1} (\dot{\alpha}_i + \Delta\dot{\alpha}_i)\mathbb{E}_{p,i} + \dot{\alpha}_m \mathbb{E}_{p,m} \tag{15b}$$

The new pair is built by solving,

$$(\alpha_m, \mathbb{C}_m, \mathbb{E}_{p,m}) = \arg \min_{\substack{\overline{\alpha}_m \in \mathcal{H}_1 \\ \overline{\mathbb{C}}_m \in \boldsymbol{\mathcal{F}}^\star \\ (\mathbf{H}\overline{\mathbb{C}}_m + \overline{\mathbb{E}}_{p,m}) \in \boldsymbol{\mathcal{E}}^\star}} \left\| \mathbf{A}\overline{\alpha}_m \overline{\mathbb{C}}_m - \dot{\overline{\alpha}}_m \overline{\mathbb{E}}_{p,m} - \mathbb{\Delta}_2 \right\|_{\mathbf{M}} \tag{16}$$

where

$$\mathbb{\Delta}_2 = \mathbf{A}\left(\hat{\boldsymbol{\sigma}} - \sum_{i=1}^{m-1} (\alpha_i + \Delta\alpha_i)\mathbb{C}_i\right) - \left(\hat{\dot{\boldsymbol{\varepsilon}}}_p - \sum_{i=1}^{m-1} (\dot{\alpha}_i + \Delta\dot{\alpha}_i)\mathbb{E}_{p,i}\right)$$

is known.

A fixed point algorithm is used to solve problem (16). The algorithm is initialized with a random time function, then

- knowing the time function α_m, the space functions are searched as

$$(\mathbb{C}_m, \mathbb{E}_{p,m}) = \arg \min_{\substack{\overline{\mathbb{C}}_m \in \boldsymbol{\mathcal{F}}^\star \\ (\mathbf{H}\overline{\mathbb{C}}_m + \overline{\mathbb{E}}_{p,m}) \in \boldsymbol{\mathcal{E}}^\star}} \left\| \mathbf{A}\alpha_m \overline{\mathbb{C}}_m - \dot{\alpha}_m \overline{\mathbb{E}}_{p,m} - \mathbb{\Delta}_2 \right\|_{\mathbf{M}} \tag{17}$$

  whose resolution is detailed in the next section,
- knowing the space functions $(C_m, \mathbb{E}_{p,m})$, the time function is searched as

$$\alpha_m = \arg \min_{\overline{\alpha}_m \in \mathbb{H}_1} \left\| \mathbf{A}\overline{\alpha}_m \mathbb{C}_m - \dot{\overline{\alpha}}_m \mathbb{E}_{p,m} - \mathbb{\Delta}_2 \right\|_{\mathbf{M}} \tag{18}$$

  which can be rewritten as a differential equation and solved [15,24].

This algorithm converges quickly and in practice only 3 to 10 iterations are necessary to reach a stagnation of the time function.

At each linear stage of LATIN iteration, we decided to enrich the ROB by at most one vector. At the first iteration, a new vector is generated, then for the next iterations, an update of the time functions is first performed and new vectors added only if necessary. The way the PGD pairs are obtained could be called LATIN progressive PGD.
The PGD of the solution obtained at the end of LATIN iterations is not optimal for this last iteration as the search direction $\mathbf{A}$ evolves along the iterations. This point is detailed in section 4.1.

3 A new algorithm to find the space functions

A major difficulty in the fixed point method to determine a new pair of functions consists in ensuring the static and kinematic admissibilities of the space functions (17), which is mandatory to reuse the PGD pairs along LATIN iterations.

Removing the subscript m for the sake of simplicity, let us define $e^2 = \|\mathbf{A}\alpha\mathbb{C} - \dot{\alpha}\mathbb{E}_p - \mathbb{\Delta}_2\|_{\mathbf{M}}$, which must be minimized under the constrains that $\mathbb{E} \in \boldsymbol{\mathcal{E}}^\star$ and $\mathbb{C} \in \boldsymbol{\mathcal{F}}^\star$.

3.1 Previous algorithm

The method proposed in previous works [15,24] consisted in a fixed point algorithm. The space function $\mathbb{E}$ associated to the total strain was introduced and e^2 written as a function of $\mathbb{E}$ and $\mathbb{C}$ through the use of the Hooke equation and the partition of strain $\mathbb{E}_p = \mathbb{E} - \mathbf{H}^{-1}\mathbb{C}$, $e^2 = \left\|\mathbf{A}\alpha\mathbb{C} + \dot{\alpha}\mathbf{H}^{-1}\mathbb{C} - \dot{\alpha}\mathbb{E} - \mathbb{A}_2\right\|_{\mathbf{M}}$.

The fixed point algorithm consisted in finding the minimum of e^2 ensuring alternatively the kinematic admissibility and the static admissibility.

- $\mathbb{E}$ being known find $\mathbb{C} = \arg \min_{\overline{\mathbb{C}} \in \boldsymbol{\mathcal{F}}^\star} e^2(\overline{\mathbb{C}}, \mathbb{E})$, whose resolution leads to one assembly of a finite elements problem and one inversion the obtained rigidity matrix after dualization.
- $\mathbb{C}$ being known find $\mathbb{E} = \arg \min_{\overline{\mathbb{E}} \in \boldsymbol{\mathcal{E}}^\star} e^2(\mathbb{C}, \overline{\mathbb{E}})$, whose resolution leads to one assembly of a finite elements problem and one inversion.

It is important to note that these minimization problems involve both the search direction $\mathbf{A}$, which is updated at each iterations of the LATIN to capture the nonlinear behavior, and the time function α, which varies during the iterations of the space-time fixed point algorithm.

In that case no convergence property was proved and after discretization numerous assemblies of matrices must be performed, which induced a huge computational cost.

The calculation cost of this algorithm can be estimated on an academic example. Let consider the example of a plate with a hole with imposed displacement. The symmetry conditions allow to consider only a eight of the plate (Fig. 2). A 2 periods sinusoidal displacement is prescribed.

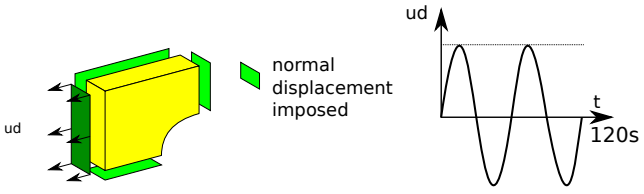

Fig. 2 Boundary conditions

One can notice on Fig. 3 that the method converges to the solution. LATIN indicator of 10^{-2} is reached in 34 iterations. 17 pairs of functions are generated during these iterations. In 17 others iterations, the solution is improved only through the update of the time functions, *i.e.* by the use of the ROM.

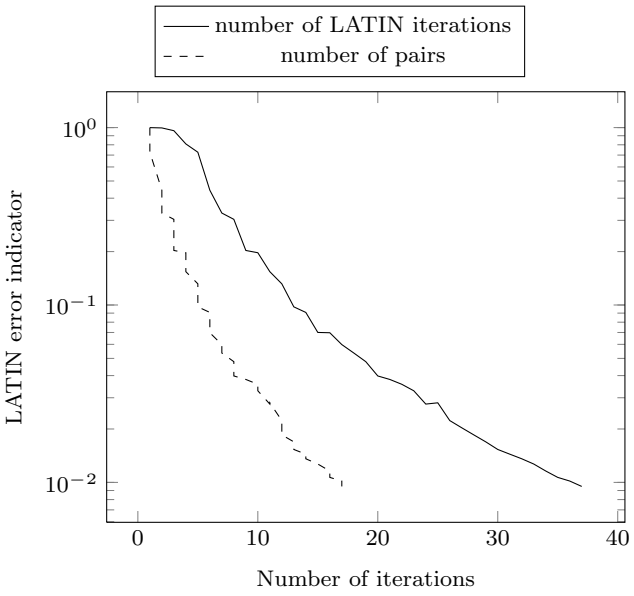

Fig. 3 LATIN indicator versus number of iterations and number of functions, plate with a hole

Let now detailed the calculation cost of space function in the enrichment of the ROB with 17 pairs. The number of pairs and the number of iterations in the space-time fixed point (Eq. 17 and Eq. 18) determines the number of space problems to be solved in total. 10 iterations in the time-space fixed point algorithm were used in the case of the plate, witch makes a total of 170 space problems to be solved.

Two assemblies and factorizations, one for the kinematic problem and one for the static problem of the linear systems are necessary at each space resolution ($170 \times 2 = 340$). The number of inversions depends on the number of iterations in the static-kinematic fixed point, which was 10 in the example ($340 \times 10 = 3,400$). Let us introduce a time unit (tu) as the time of one iteration of a classic Newton tangent algorithm. During one iteration we evaluate the cost as follows 10% to assemble the matrice associated to the finite elements problem, 88% to factorize it and 2% to solve the linear system of equations. With this orders of magnitude the total time dedicated to the space problem (Eq. 17) with the old algorithm can be evaluated.

operation	number	unitary cost (tu)	total (tu)
assembly	340	0.1	34
factorization	340	0.88	299.2
inversion	3,400	0.02	68
total			401.2

A total of 401.2 time units are necessary to generate the 17 spaces functions with this algorithm.

This number can be compared with a classic Newton tangent algorithm. For 120 time steps and a mean of

two iterations per time step, calculation cost is $2\times120 = 240$ time units, which is less than 401.2.

3.2 New algorithm

To avoid the drawbacks of previous algorithm we propose here a new algorithm that will be shown to involve matrices that do not depend on the search direction and the time function and then will decrease the computational cost. With the new algorithm, the solution of a nonlinear problem is solved using only the discrete matrices related to the same linear elastic problem, taking advantage at the time independence of the equilibrium equation.

Let us first remark that the plastic strain rate space function $\mathbb{E}_p$ is the main unknown of the minimization problem (17). Indeed, introducing the state law (1) in the static admissibility condition of $\mathbb{C}$ (4) reads,

$$\forall \mathbb{E}^\star \in \boldsymbol{\mathcal{E}}^\star, \quad \int_\Omega \mathbf{H}(\mathbb{E}-\mathbb{E}_p) : \mathbb{E}^\star d\Omega = 0 \tag{19}$$

which allows to define an operator $\mathbf{E}$ and $\mathbf{C}$ such that $\mathbb{E} = \mathbf{E}\mathbb{E}_p$ and $\mathbb{C} = \mathbf{C}\mathbb{E}_p$ ($\mathbf{C} = \mathbf{H}(\mathbf{E}-\mathbf{I})$). The error e^2 is then only a function of $\mathbb{E}_p$ and the minimization problem (17) consists in finding $\mathbb{E}_p$ such that,

$$\mathbb{E}_p = \arg \min_{\mathbb{E}_p} \|\mathbf{A}\alpha\mathbf{C}\mathbb{E}_p - \dot\alpha\mathbb{E}_p - \mathbb{\Delta}_2\|_{\mathbf{M}} \tag{20}$$

$\mathbb{E}$ and $\mathbb{C}$ are obtained as a post-treatment by

$$\mathbb{E} = \mathbf{E}\mathbb{E}_p, \tag{21}$$

$$\mathbb{C} = \mathbf{C}\mathbb{E}_p \tag{22}$$

One can notice that the admissibility conditions $\mathbb{E} \in \boldsymbol{\mathcal{E}}^\star$ and $\mathbb{C} \in \boldsymbol{\mathcal{F}}^\star$ are fulfilled by construction.

3.2.1 Residual problem

Minimization problem (20) is formally rewritten by introducing an intermediate variable $\mathbb{Z}$ that only depends on $\mathbb{E}_p$, $\mathbb{Z} = \begin{bmatrix}\mathbb{E}-\mathbb{E}_p & \mathbb{E}_p\end{bmatrix}^T = \begin{pmatrix}\mathbf{E}-\mathbf{I}\\ \mathbf{I}\end{pmatrix}\mathbb{E}_p$. The error e^2 is then,

$$e^2 = 1/2\int_\Omega \mathbb{Z}^T\mathbf{Q}\mathbb{Z}\,\mathrm{d}\Omega + \int_\Omega \mathbb{L}^T\mathbb{Z}\,\mathrm{d}\Omega + a^2/2 \tag{23}$$

where , $\mathbf{Q}$ is a symmetric matrix and $\mathbb{L}^T$ a linear form and a a scalar (24), which are known quantities at this stage,

$$\mathbf{Q} = \int_{\mathcal{I}} \begin{bmatrix}\alpha\mathbf{A}\mathbf{H}\mathbf{M}\alpha\mathbf{A}\mathbf{H} & -\alpha\mathbf{A}\mathbf{H}\mathbf{M}\dot\alpha\\ -\alpha\mathbf{A}\mathbf{H}\mathbf{M}\dot\alpha & \dot\alpha\mathbf{M}\dot\alpha\end{bmatrix}\mathrm{d}t \tag{24a}$$

$$\mathbb{L} = \int_{\mathcal{I}} \begin{bmatrix}-\alpha\mathbf{A}\mathbf{H}\mathbf{M}\mathbb{\Delta}_2\\ \dot\alpha\mathbf{M}\mathbb{\Delta}_2\end{bmatrix}\mathrm{d}t \tag{24b}$$

$$a^2/2 = \int_{\mathcal{I}} \mathbb{\Delta}_2\mathbf{M}\mathbb{\Delta}_2\,\mathrm{d}t \tag{24c}$$

One can recognize a classic quadratic form in e^2. The stagnation of e^2 with respect to $\mathbb{E}_p$ is found for,

$$\frac{\partial e^2}{\partial \mathbb{E}_p} = \frac{\partial e^2}{\partial \mathbb{Z}}\frac{\partial \mathbb{Z}}{\partial \mathbb{E}_p} = 0$$
$$\iff \int_\Omega (\mathbf{Q}\mathbb{Z}+\mathbb{L})^T\begin{pmatrix}\mathbf{E}-\mathbf{I}\\ \mathbf{I}\end{pmatrix}\mathbb{E}_p^\star\,\mathrm{d}\Omega = 0, \quad \forall \mathbb{E}_p^\star \tag{25}$$

Let us defined $q = \int_\Omega (\mathbf{Q}\mathbb{Z}+\mathbb{L})^T\begin{pmatrix}\mathbf{E}-\mathbf{I}\\ \mathbf{I}\end{pmatrix}\mathbb{E}_p^\star\,\mathrm{d}\Omega$; split into two integrals

$$q = \int_\Omega (\mathbf{Q}_1\mathbb{Z}+\mathbb{L}_1) : (\mathbb{E}^\star - \mathbb{E}_p^\star)\,\mathrm{d}\Omega + \int_\Omega (\mathbf{Q}_2\mathbb{Z}+\mathbb{L}_2) : \mathbb{E}_p^\star\,\mathrm{d}\Omega \tag{26}$$

where $\mathbf{Q}_1$, $\mathbb{L}_1$ and $\mathbf{Q}_2$, $\mathbb{L}_2$ are respectively the first and the second line of operators $\mathbf{Q}$ and $\mathbb{L}$. In this expression, the stagnation still bring in $\mathbb{E}^\star$. Using the equilibrium, it can be expressed only with $\mathbb{E}_p^\star$. Indeed, as the static admissibility $C^\star \in \boldsymbol{\mathcal{F}}^\star$ is enforced, $\forall \mathbb{\Sigma} \in \boldsymbol{\mathcal{E}}^\star$,

$$\int_\Omega \mathbb{C}^\star : \mathbb{\Sigma}\,\mathrm{d}\Omega = 0 \Leftrightarrow \int_\Omega \mathbf{H}(\mathbb{E}^\star - \mathbb{E}_p^\star) : \mathbb{\Sigma}\,\mathrm{d}\Omega = 0$$
$$\Leftrightarrow \int_\Omega \mathbf{H}\mathbb{\Sigma} : \mathbb{E}^\star\,\mathrm{d}\Omega = \int_\Omega \mathbf{H}\mathbb{\Sigma} : \mathbb{E}_p^\star\,\mathrm{d}\Omega \tag{27}$$

That means that any kinematically admissible to zero field has the same virtual energy with respect to $\mathbb{E}^\star$ or $\mathbb{E}_p^\star$ when the Hooke operator is involved.

A particular $\mathbb{\Sigma} \in \boldsymbol{\mathcal{E}}^\star$ is chosen, such that $\forall \mathbb{E}^\star \in \boldsymbol{\mathcal{E}}^\star$,

$$\int_\Omega \mathbf{H}\mathbb{\Sigma} : \mathbb{E}^\star\,\mathrm{d}\Omega + \int_\Omega (\mathbf{Q}_1\mathbb{Z}+\mathbb{L}_1) : \mathbb{E}^\star\,\mathrm{d}\Omega = 0 \tag{28}$$

The calculation of $\mathbb{\Sigma}$ is performed through a classic finite elements problem. By using (27), (28) is written as,

$$\int_\Omega (\mathbf{Q}_1\mathbb{Z}+\mathbb{L}_1) : \mathbb{E}^\star\,\mathrm{d}\Omega = -\int_\Omega \mathbf{H}\mathbb{\Sigma} : \mathbb{E}_p^\star\,\mathrm{d}\Omega \tag{29}$$

This last equation enables to write q (26) only as a function of $\mathbb{E}_p^\star$,

$$q = \int_\Omega \mathbb{R}(\mathbb{Z},\mathbb{\Sigma}) : \mathbb{E}_p^\star\,\mathrm{d}\Omega \tag{30}$$

where

$$\mathbb{R}(\mathbb{Z},\mathbb{\Sigma}) = -(\mathbf{Q}_1 Z + \mathbb{L}_1) - \mathbf{H}\mathbb{\Sigma} + (\mathbf{Q}_2\mathbb{Z}+\mathbb{L}_2) \tag{31}$$

$\mathbb{R}(\mathbb{Z},\mathbb{\Sigma})$ is a residual, only function of $\mathbb{E}_p$. The solution is found when $\mathbb{R}$ is equal to zero.

3.2.2 Iterative procedure

We use an iterative procedure, similar to a conjugate gradient algorithm, to find $\mathbb{E}_p$ such as e^2 is minimum (Algorithm 1).

This iterative procedure is initialized with $\mathbb{E}_{p,0} = \mathbb{0}$, which gives through the equilibrium (19) that $\mathbb{E}_0 = \mathbb{0}$ and then $\mathbb{Z}_0 = \begin{bmatrix}\mathbb{0} & \mathbb{0}\end{bmatrix}^T$, with $\mathbb{0}$ the field with null value at any point.
At iteration n, a correction $\mathbb{Z}_n$ of $\mathbb{Z}$ is searched. One must first determined the residual $\mathbb{R}$ with (31), where $\mathbb{\Sigma}$ is calculated through problem (28), $\mathbb{Z}$ being known from previous iteration,

$$\mathbb{Z} = \sum_{j=0}^{n-1} \mathbb{Z}_j$$

Then a direction $\mathbb{E}''_{p,n}$ must be chosen,

$$\mathbb{E}''_{p,n} = (\mathbf{Q}_{22} - \mathbf{Q}_{12})^{-1}\mathbb{R} \tag{32}$$

The equilibrium (19) enables to determine $\mathbb{E}''_n$ associated with this direction (33), with a finite element method.

$$\forall \mathbb{E}^\star \in \boldsymbol{\mathcal{E}}^\star, \quad \int_\Omega \mathbf{H}(\mathbb{E}''_n - \mathbb{E}''_{p,n}) : \mathbb{E}^\star d\Omega = 0 \tag{33}$$

The direction $\mathbb{Z}''_n = \begin{bmatrix}\mathbb{E}''_n - \mathbb{E}''_{p,n} & \mathbb{E}''_{p,n}\end{bmatrix}$ is then set orthogonal to previous iterations directions by a Graam-Schmitt procedure to find $\mathbb{Z}'_n$ such as

$$\begin{cases} \mathbb{Z}''_n = \sum_{i=0}^{n-1} \gamma_i \mathbb{Z}'_i + \mathbb{Z}'_n \\ \int_\Omega (\mathbf{Q}\mathbb{Z}'_j)^T \mathbb{Z}'_n \,\mathrm{d}\Omega = 0 \quad \forall j < n \end{cases} \tag{34}$$

In practice orthogonalization with the direction of iteration $n-1$ is enough. With this search direction, one searched the intensity of descent λ

$$\mathbb{Z}_n = \lambda \mathbb{Z}'_n \tag{35}$$

such as e^2 is minimum. The new value of $\mathbb{Z}$ is

$$\mathbb{Z} = \sum_{j=0}^{n-1} \mathbb{Z}_j + \mathbb{Z}_n$$

which gives an evolution of the error,

$$e^2(\mathbb{Z}_n) = e^2(\lambda \mathbb{Z}'_n) = \frac{1}{2}\lambda^2 \int_\Omega (\mathbf{Q}\mathbb{Z}'_n)^T \mathbb{Z}'_n \,\mathrm{d}\Omega + \lambda \int_\Omega \mathbb{L}^T \mathbb{Z}'_n \,\mathrm{d}\Omega \tag{36}$$

The minimum of $e^2(\lambda \mathbb{Z}'_n)$ is found for,

$$\lambda = -\frac{\int_\Omega \mathbb{L}^T \mathbb{Z}'_n \,\mathrm{d}\Omega}{\int_\Omega (\mathbf{Q}\mathbb{Z}'_n)^T \mathbb{Z}'_n \,\mathrm{d}\Omega} \tag{37}$$

With this value of λ,

$$e^2(\mathbb{Z}_n) = -\frac{1}{2}\frac{\left(\int_\Omega \mathbb{L}^T \mathbb{Z}'_n \,\mathrm{d}\Omega\right)^2}{\int_\Omega (\mathbf{Q}\mathbb{Z}'_n)^T \mathbb{Z}'_n \,\mathrm{d}\Omega} \tag{38}$$

One can remarks that $e^2(\mathbb{Z}_n)$ is always negative, while $e^2(\mathbb{Z})$ is always positive. The correction by $\mathbb{Z}_n$ always induces a diminution of $e^2(\mathbb{Z})$. Iterations of the algorithm are stopped when $e^2(\mathbb{Z}_n)/\left\|\mathbb{\Delta}_2\right\|_{\mathbf{M}}$ is less than a threshold value. A value of 10^{-2} appears in the numerical test to be small enough and is reached in average in 10 iterations.

Algorithm 1: Space function determination

```
Initialization E_{p,0} = 0
while e^2(Z_n)/||Δ_2||_M > 10^{-2} do
    Σ (28) (FE problem)
    Residual calculation R (31)
    Descent direction E''_{p,n} (32)
    Descent direction E''_n (33) (FE problem)
    Graam-Schmitt Z'_n (34)
    Descent intensity λ (37)
    Correction Z_n = λ Z'_n
    Update of Z ← Z + Z_n
    Error evolution e^2(Z_n) (38)
end
```

3.2.3 Evaluation of the calculation cost

The calculation cost can be estimated on the academic example with this new algorithm. The discrete matrix associated to $\int_\Omega \mathbf{H}\mathbb{E}''_n : \mathbb{E}^\star \,\mathrm{d}\Omega$ (33) and $\int_\Omega \mathbf{H}\mathbb{\Sigma} : \mathbb{E}^\star \,\mathrm{d}\Omega$ (28) is assembled and factorized only one time. The number of inversions depend on the number of pairs, the number of iterations in the time-space fixed point algorithm and the number of iterations in the iterative procedure of the space function algorithm ($17 \times 10 \times 10 \times 2 = 3,400$).

operation	number	unitary cost (tu)	total (tu)
assembly	1	0.1	0.1
factorization	1	0.88	0.88
inversion	$3,400$	0.02	68
total			68.98

The total number of time units is 68.98 which is around 6 times less than using the previous algorithm and 3.5 times less than using a Newton tangent algorithm. This total is just an estimation of the calculation time with the proposed algorithm, the real one is higher because of all operations not taken into account.

4 Some numerical properties of the PGD

Some properties of the PGD, as the fact that it is obtained along LATIN iterations, the number of functions depending on the mesh refinement or the time step size, remain unclear. We tested the influence of these parameters on the number of generated functions and on the computational time to reach a constant quality of solution.

4.1 LATIN PGD

The LATIN PGD consists in finding the pairs along the iterations of the LATIN method. The linear problem on which the PGD is build evolves along the iterations as the PGD is introduced at the linear stage of the LATIN. The LATIN PGD is then not optimal with respect to the linear stage at the last LATIN iteration, which is the best linear approximation of the problem. This fact is highlighted by comparing the convergence rate of the PGD with space functions generated along LATIN iterations or generated at the last iteration.

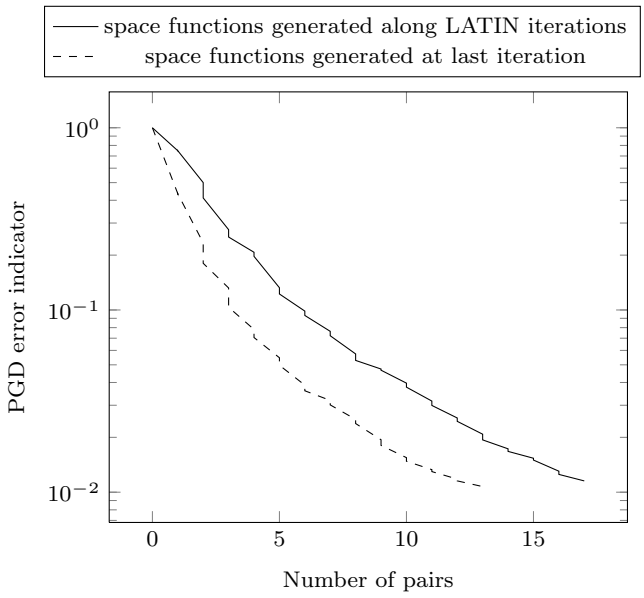

Fig. 4 Comparison PGD error versus number of pairs, when approximating the last LATIN iteration with space functions generated along LATIN iterations or generated at the last iteration

One can remark on Fig. 4 that the convergence rate of the LATIN PGD is less than the progressive PGD for the first functions. For the last functions the convergence rate is the same. This can be explained by the fact that the tangent evolution law operator evolves quickly in the first iterations of the LATIN method and becomes constant in next iterations. Nevertheless the number of pairs do not increases dramatically (16 using LATIN PGD compared to 14 using space function generated at the last iteration).

4.2 Influence of mesh characteristic length and time step

We compared the number of pairs to reach a LATIN indicator of 10^{-2} when using different meshes and time steps. One can observe on Fig. 5 that the number of pairs does not evolve when increasing the number of degrees of freedom. However, it depends on the number of time steps until it reaches a plateau. This is related to the influence of the time step on the calculation of the evolution law of the constitutive relation.

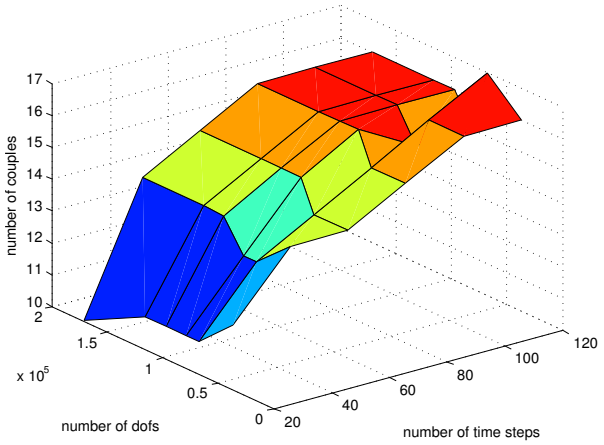

Fig. 5 Number of pairs versus number and dofs and time steps

4.3 Performance of the algorithm, comparison with Abaqus

Model reduction techniques do not always allow to save computational time. The calculation of the reduced model, integral in time or space, can lead to be less efficient than a direct method. The control of the error added by the reduced model is also a critical point. To check this, the LATIN method has been implemented in a C++ code linked to standard library of linear algebra in the code platform of the LMT-Cachan. Then the efficiency of the method is compared to the commercial software Abaqus on an example on an example provided with its documention. The elastic-viscoplastic constitutive formulation is calculated in Abaqus through a Umat generated by the Zmat program [1].

[1] http://www.nwnumerics.com/Z-mat/

The example considered herein is the upper part of a vessel (FIG. 6) from the Abaqus documentation. The symmetry conditions allow us to take a quarter of the part. An internal pressure is imposed with a sinusoidal variation in time (FIG. 6). The part is meshed with 160, 587 quadratic tetrahedrons (10 nodes), which leads to 740, 097 degrees of freedom. An Euler implicit time scheme is used with 60 time intervals.

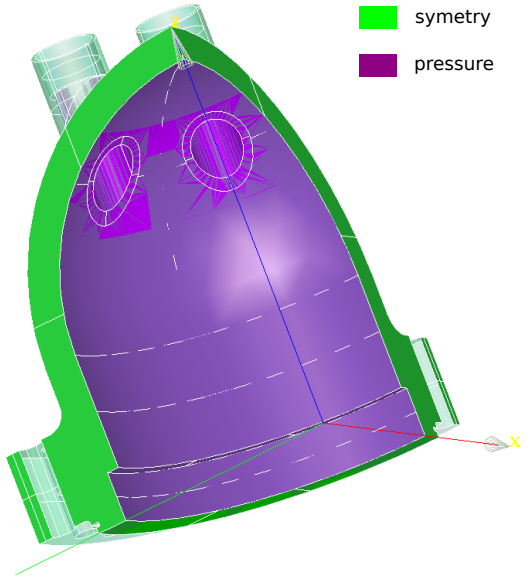

Fig. 6 Boundary conditions

100 iterations are necessary to reach a LATIN error indicator of 10^{-2} (FIG. 7). 20 pairs of time and space functions are generated, the quality of the solution is improved only by an update of the time functions in most of the iterations reuse of the ROB without enrichment).

As shown in Table 1, calculation times obtained with Abaqus and the LATIN program are similar. The Abaqus calculation time is increased by the use of the Zmat program, but the elastic-viscoplastic behavior law is not present natively in this program.

	Wallclock time (sec)
Abaqus + Zmat	12, 308
LATIN	8, 920

Table 1 Comparison of LATIN and Abaqus calculation times

5 Conclusions

In this paper, we showed that it is possible to build a reduced model with the PGD in the case of a non-

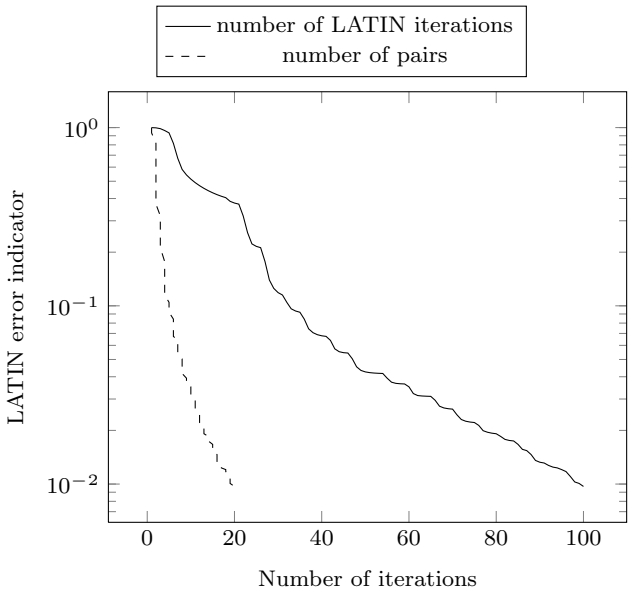

Fig. 7 LATIN indicator versus number of iterations and number of pairs, vessel head

linear constitutive law through the use of the LATIN method, which allows to build a linear approximation of the problem at each of its iterations on which the PGD is computed easily.

We have presented a new algorithm for the calculation of the ROB of the Proper Generalized Decomposition for the *a priori* construction of a separated variable representation of the solution of time-dependent problem, based on a minimum residual problem. This new algorithm is 6 times less computationally expensive than fixed point techniques used in previous works, as the discrete operator involved in the finite element problems does not change along the LATIN iterations, even if the solution of a problem with an elastic-viscoplastic formulation is searched.

We have highlighted some properties of the reduced model built along LATIN iterations. We checked the quality of the PGD reduced model obtained along LATIN iterations compared to a PGD performed on the best linear approximation of the problem. We concluded that the PGD model built along LATIN iterations is not optimal but of good quality as very few additional pairs are generated in that case compared to the PGD performed on the best linear approximation of the problem. On a simple academic example, we showed that the number of pairs does not depend on the mesh size, and depends on the number of time steps until convergence of the time integration scheme.

We finally have illustrated the performances of the LATIN method with this new algorithm on an example of the Abaqus documentation, by comparing the calculation time obtained with the LATIN and with Abaqus

in the case of an elastic-viscoplastic material formulation. We obtained a gain of 30% in CPU time in favor of the LATIN method.

This work shows that it is possible to generate a good quality PGD reduced model associated to a design configuration defined by a set of parameters (material parameters, boundary conditions...). The next step is is to deal with a parametric study and then to compute the solution for several sets of parameters. For a new set of parameters, the idea is to initialize the algorithm using the ROB already generated for the previous set. It is quite inexpensive and generally sufficient if the two sets of data are closed. The error indicators already defined allow to know if the required quality is achieved and to automatically enrich the approximation by adding new functional products by performing new LATIN iterations if it is not the case. Using this strategy based on the PGD reduced model, we can expect to drastically reduce the computational time associated to a parametric study [11].

Acknowledgements This work was supported by the *Agence Nationale pour la Recherche* as part of project MATETPRO07: Approche mécano Probabiliste Pour la conception RObuste en FatIgue (APPROFI).

References

1. Ammar, A., Mokdad, B., Chinesta, F., Keunings, R.: A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids: Part II: Transient simulation using space-time separated representations. Journal of Non-Newtonian Fluid Mechanics **144**, 98–121 (2007)
2. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique **339**(9), 667–672 (2004)
3. Bergmann, M., Cordier, L.: Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. Journal of Computational Physics **227**(16), 7813–7840 (2008)
4. Carlberg, K., Bou-Mosleh, C., Farhat, C.: Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. International Journal for Numerical Methods in Engineering **86**(2), 155–181 (2011)
5. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. Siam Journal On Scientific Computing **32**(5), 2737–2764 (2010)
6. Chinesta, F., Ammar, A., Cueto, E.: Proper generalized decomposition of multiscale models. International Journal for Numerical Methods in Engineering **83**(8-9), 1114–1132 (2010)
7. Chinesta, F., Ladevèze, P., Cueto, E.: A short review on model order reduction based on proper generalized decomposition. Archives of Computational Methods in Engineering **18**(4), 395–404 (2011)
8. Cognard, J.Y., Ladevèze, P.: A large time increment approach for cyclic viscoplasticity. International Journal of Plasticity **9**(2), 141–157 (1993)
9. Cognard, J.Y., Ladevèze, P., Talbot, P.: A large time increment approach for thermo-mechanical problems. Advances in Engineering Software **30**(9-11), 583–593 (1999)
10. Gunzburger, M.D., Peterson, J.S., Shadid, J.N.: Reduced-order modeling of time-dependent pdes with multiple parameters in the boundary data. Computer Methods in Applied Mechanics and Engineering **196**(4-6), 1030–1047 (2007)
11. Heyberger, C., Boucard, P.A., Nron, D.: Multiparametric analysis within the proper generalized decomposition framework. Computational Mechanics pp. 1–13. DOI 10.1007/s00466-011-0646-x
12. Kunisch, K., Xie, L.: Pod-based feedback control of the burgers equation by solving the evolutionary hjb equation. Computers & Mathematics with Applications **49**(7-8), 1113–1126 (2005)
13. Ladevèze, P.: New algorithms: mechanical framework and development (in french). Compte rendu de l'académie des Sciences **300(2)**, 41–44 (1985)
14. Ladevèze, P.: Nonlinear Computationnal Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation. Springer Verlag (1999)
15. Ladevèze, P., Passieux, J.C., Néron, D.: The LATIN multiscale computational method and the proper generalized decomposition. Computer Methods in Applied Mechanics and Engineering **199**, 1287–1296 (2010)
16. Leygue, A., Verron, E.: A first step towards the use of proper general decomposition method for structural optimization. Archives of Computational Methods in Engineering **17**(4), 465–472 (2010)
17. Lieu, T., Farhat, C., Lesoinne, M.: Reduced-order fluid/structure modeling of a complete aircraft configuration. Computer Methods in Applied Mechanics and Engineering **195**(41-43), 5730–5742 (2006)
18. Miyamura, T., Noguchi, H., Shioya, R., Yoshimura, S., Yagawa, G.: Elastic–plastic analysis of nuclear structures with millions of dofs using the hierarchical domain decomposition method. Nuclear Engineering and Design **212**(1-3), 335–355 (2002)
19. Néron, D., Ladevèze, P.: Proper generalized decomposition for multiscale and multiphysics problems. Archives of Computational Methods in Engineering **17** (2010)
20. Nguyen, N.C., Peraire, J.: An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. International Journal for Numerical Methods in Engineering **76**(1), 27–55 (2008)
21. Nouy, A.: A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Computer Methods in Applied Mechanics and Engineering **199**(23-24), 1603–1626 (2010)
22. Nouy, A., Ladevèze, P.: On a multiscale computational strategy with time and space homogenization for structural mechanics. Computer Methods in Applied Mechanics and Engineering **192**, 3061–3087 (2004)
23. Nouy, A., Maître, O.P.L.: Generalized spectral decomposition for stochastic nonlinear problems. Journal of Computational Physics **228**(1), 202–235 (2009)
24. Relun, N., Néron, D., Boucard, P.A.: Multiscale elastic-viscoplastic computational analysis: a detailed example. European Journal of Computational Mechanics (to appear 2012)
25. Sarbandi, B., Cartel, S., Besson, J., Ryckelynck, D.: Truncated integration for simultaneous simulation of sintering using a separated representation. Archives of Computational Methods in Engineering pp. 1–9 (2010)