A data compression technique for PGD reduced-order modeling

Matteo Capaldo, David Néron, Pierre Ladevèze, Pierre-Alain Guidault

YIC2013
Second ECCOMAS Young Investigators Conference
2–6 September 2013, Bordeaux, France

A data compression technique for PGD reduced-order modeling

M. Capaldo [a,*], David Néron [a], Pierre Ladevèze [a], Pierre-Alain Guidault [a]

[a] LMT - ENS Cachan/CNRS/UPMC/PRES UniverSud Paris
61 av. du Prsident Wilson, F-94230 Cachan, France

*capaldo@lmt-ens.cachan.fr

Abstract. *This work concerns the Proper Generalized Decomposition (PGD), an a priori model reduction technique used to solve problems, eventually nonlinear, defined over the time-space domain. PGD seeks the solution of a problem in a reduced-order basis generated by a dedicated algorithm. This is the LATIN method, an iterative strategy which generates the approximations of the solution over the entire time-space domain by successive enrichments. Herein an algebraic framework adapted to PGD is proposed. It defines a compressed version of the data making less expensive the elementary algebraic operations.*

Keywords: PGD; LATIN; model reduction technique; data compression technique.

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 unsuitable. Model reduction techniques constitute an efficient way to circumvent this obstacle 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 assuming this ROB by the decomposition of the solution of a surrogate model relevant to the initial model (see e.g. [1, 2, 5, 8]). A priori methods follow a different path by building progressively an approximate separated representation of the solution, without assuming any basis (see e.g. [3, 4, 6, 8]).

This work focusses on the Proper Generalized Decomposition (PGD) which belongs to the second family and is used herein to solve problems defined over the time-space domain and which are eventually nonlinear. For solving such problems, the PGD, originally introduced as the radial loading approximation in [7], 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 [8]. 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 [4, 6]).

However, model reduction techniques are more particularly efficient when the ROM needs to be constructed only once, which is not the case with the greedy algorithm, especially for nonlinear problems. In that case, the various operators must be updated along the iterations and the calculation of the ROM represents a large part of the global CPU cost (see e.g. [5]). A new algebric framework is, herein, proposed to simplify the elementary operations on PGD fields, making the ROM generation less expensive. It is based on the concept of *reference* times, points and parameters and allows to define a *compressed* version of the data. The space of *compressed* data shows interesting properties dealing the elementary algebric operations. That should lead to important improvements in terms of calculation performance.

2 DATA COMPRESSION

For the sake of semplicity, in this artcile, we will restrict the development of the proposed method to the case of fields depending on time and space, but it could be easily extended to the space of parameters.

2.1 Representation in a separated variables framework

In our algorithms previously developed, PGD has been employed to represent only few quantities which solution needed the resolution of problems global in space and defined on the whole time domain. These quantities used to cohabit with functions defined in a standard form (called "*full*" below). To take advantage of the PGD representation in terms of stockage and calculation cost (in particular, to compute an integral over the time-space domain it allows to separate the integration over the space from the intergation over the time), PGD representation must be extended the to the others fields of the problem.
The aim of this work is to semplify the calculation of the standard algebric operation (products, additions, integration, ..) for the fields defined in the PGD form and to avoid the artificial increasing of the PGD modes representing a function. An exemple is presented to clarify. Let's consider the calculation of the product $F = f\ f'$, where the functions are described on a reduced order basis of the order of 20. Formal representations yields to:

$$f(t,\underline{M}) = \sum_{i=1}^{m} \lambda_i(t)\ \Lambda_i(\underline{M}) \qquad and \qquad f'(t,\underline{M}) = \sum_{i=1}^{m'} \theta_i(t)\ \Theta_i(\underline{M}) \tag{1}$$

where m and m' are of the order of 20. We intend obtain their product F in the separeted variables form, in order to remain in the PGD framework :

$$F(t,\underline{M}) = f(t,\underline{M})\ f'(t,\underline{M}) \approx \sum_{i}^{M} \alpha_i(t)\beta_i(\underline{M}). \tag{2}$$

A first way is to perform the product term by term. It leads to a function described by the sum of $M = m \times m'$ products, where M is of the order of 400. These are, clearly, much more than the required functions to represent it and this number can increase swiftly over the executing operations.
The second approach is to rebuild the *full* version of the function f and f', and then to perform the product getting F. After that it's necessary to carry out the *Singular Value Decomposition* to achieve a separated variable representation of F with the desired approximation. This way is expensive as well as the first one due to the SVD that can lead to a vectorial space of large dimension.
Herein a different algebric framework is proposed. It is capable to build a first approximation of the function, solution of the operation which has to be executed. The concept of *reference times* and *reference points* is introduced [7] (cf. **Fig. 1**). The time domain is splitted in m_1 sub-intervals I_i of the size Δt_i. The centers t_i of the sub-interval I_i are called reference times. For the space domain m_2 points $\underline{M}_j$ are introduced and the domain Ω is divided in the sub-domains Ω_j. The $\underline{M}_j$ are called reference points and the size of the patch are indicated ω_j.

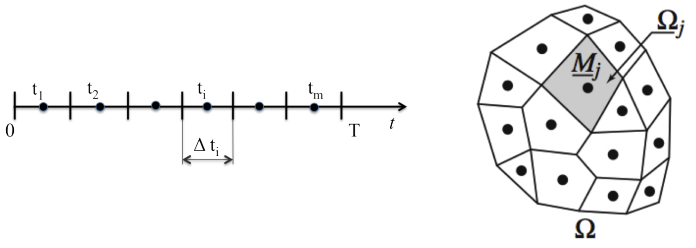

Figure 1: Reference times and reference points

A function f defined on the domain $I \times \Omega$ is compressed into the form $\bar{f}$, which consists of its *generalized components*. $\bar{f} = \{(\bar{a}_{ij}, \bar{b}_{ij})\}$ is defined below :

$$\bar{f} := \left\{ \begin{array}{rcccc|rcccc} \bar{a}_{ij}(t) & = & f(t,\underline{M}_j) & \text{si } t \in I_i & | & \bar{b}_{ij}(\underline{M}_j) & = & f(t_i,\underline{M}) & \text{si } \underline{M} \in \Omega_j \\ \bar{a}_{ij}(t) & = & 0 & \text{sinon} & | & \bar{b}_{ij}(\underline{M}_j) & = & 0 & \text{sinon} \end{array} \right\}_{i=1,..,m_1;\ j=1,..,m_2} \tag{3}$$

3 ALGEBRA IN THE COMPRESSED FRAMEWORK

The space of compressed fields shows interesting properties when the elementary operations must be executed. **Tab. 1** shows some operations on two compressed field f and f'.

Table 1: Elementary operations in the compressed space.

Addition	$\overline{f + f'} = \bar{f} + \bar{f}'$
Multiplication	$\overline{f\, f'} = \bar{f}\, \bar{f}'$
Derivation	$\overline{\partial f/\partial t} = \partial \bar{f}/\partial t$
Operator	$\overline{H\, f} = \bar{H}\, \bar{f}$

Returning, for instance, on the multiplication seen in the equation (2) between the two fields f and f' described in the PGD form. This will be computed following the procedure below :

1. $\bar{f} = \{(\bar{a}_{ij}, \bar{b}_{ij})\}$ and $\bar{f}' = \{(\bar{a}'_{ij}, \bar{b}'_{ij})\}$, compressed version of f and f', are calculated as shown in the equation (3).

2. $\bar{F} = \{(\bar{A}_{ij}, \bar{B}_{ij})\}$, compressed version of the operation $F = f\ f'$ is calculated as shown in **Tab. 1**. It yields :

$$\bar{A}_{ij} = \bar{a}_{ij}\ \bar{a}'_{ij} \qquad \text{and} \qquad \bar{B}_{ij} = \bar{b}_{ij}\ \bar{b}'_{ij} \tag{4}$$

3. A reconstruction of F in a separated variables form is sought. It arises from its compressed version $\bar{F}$.

4. Whether the quality of the approximation is poor, this is improved adding new PGD pairs.

4 RECONSTRUCTION BY THE COMPRESSED VERSION

The generation of a first approximation of F in the separated variables form gives just one pairs of functions for each time-space patch $I_i \times \Omega_j$. It produces :

$$F(t, \underline{M}) \approx F_0(t, \underline{M}) = \alpha_{ij}(t)\beta_{ij}(\underline{M}) \qquad \forall (t, \underline{M}) \in I_i \times \Omega_j \tag{5}$$

where the set of $\alpha_i j(t)\beta_i j(\underline{M})$ are defined by the set of $\bar{A}_i j(t)\bar{B}_i j(\underline{M})$ resulting of a minimization problem. Practically, for the mechanics problems, which is addressed this work, the chosen reconstruction method gives more importance to space domain. Normally, in fact, variations of the quantities over the space domain are stronger than variations over the time. The minimization problem leads to the explicit formulas : $\forall (t, \underline{M}) \in I_i \times \Omega_j$,

$$\alpha_{ij}(t) = \frac{\sum_{k=1}^{m_2} \omega_k \bar{A}_{ik}(t)\bar{B}_{ik}(\underline{M}_k)\lambda_k^2}{\sum_{k=1}^{m_2} \omega_k \bar{B}_{ik}(\underline{M}_k)\bar{B}_{ik}(\underline{M}_k)\lambda_k^2} \qquad \text{and} \qquad \beta_{ij}(\underline{M}) = \bar{B}_{ij}(\underline{M}) \tag{6}$$

The computational cost to generate F_0 is moderate since the explicit form of the formulas (6). To quantify the quality of this first approximation the error and the related norm are introduced below :

$$e = \frac{\|F - F_0\|}{\|F_0\|}, \qquad \|F\|^2_{I_i \times \Omega_j} = \int_{I \times \Omega} F^2 d\Omega dt = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \int_{I_i \times \Omega_j} F^2 d\Omega dt. \tag{7}$$

This evaluation can be executed cheaply exploiting that F is the product by two functions described in the PGD form, thus, their modes are sorted descending with respect their own importance. It could be sufficient to take in account the first terms to have a good forecast of e. Whether e exceeds a given threshold some PGD pairs can be added to the first approximation F_0 in order to have a finer representation of F. Mostly the quality of the first approximation is sufficient.

4.1 Numerical example

Let's consider two random functions f and f', described in the PGD form on a base of 10 functions ($m = m' = 10$). The goal is to approximate their product in the compressed form in order to have a PGD representation of it by only

one pair of time-space functions. Reference solution, that is exacte, (**Fig.2**, left), is obtained performing the product of the *full* fields of f and f'. These are gained by their PGD representation :

$$F(t,\underline{M}) = f(t,\underline{M})\ f'(t,\underline{M}) = \left(\sum_{i=1}^{10} \lambda_i(t)\Lambda_i(\underline{M})\right)\left(\sum_{i=1}^{10} \theta_i(t)\Theta_i(\underline{M})\right). \tag{8}$$

The reconstruction by the *compressed* representation on $m_1 = 17$ reference times and $m_2 = 17$ reference points (**Fig.2**, right) leads to an error $e = 1\%$ with respect to the exact solution. This is obtained generating only one PGD pair of F reconstructed for each time-space patch.

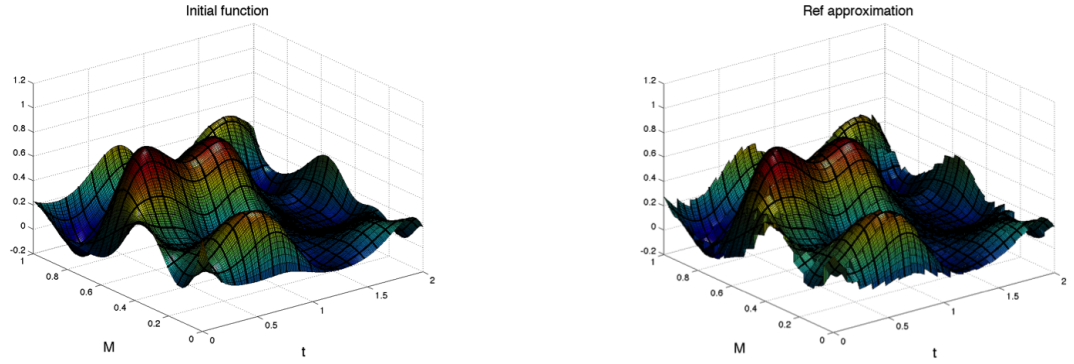

Figure 2: The function $F = f\ f'$ exact (letf) and its aprroximation by $m_1 = 17$ reference times and $m_2 = 17$ reference points (right).

5 CONCLUSION

Models reduction techniques based on PGD can be the framework to build up high performance calculation codes in order to simulate complex models, with high numbers of degrees of freedom and eventually parametric. Nevertheless in the current codes, only a limited amount of the fields is represented in the PGD form. That leads to manage fields with different representations and it hinders the potential gain of the PGD in terms of calculation and storage.

In this work the concept of a "*compressed*" representation of the fields has been introduced. It allows to manage all the fields in the PGD form. All the operations can be redefined in this framework, in order to have a first approximation in a separated variables form of themselves. Implementation in a complete calculation code is a work in progress. It should allow to gain a drastically reduction of CPU costs.

REFERENCES

[1] Kunisch, K., Volkwein, S. *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math., 90 (1), 117148, 2001.

[2] Cizmas P.G.A., OBrien T. J., et al. *Acceleration techniques for reduced-order models based on proper orthogonal decomposition* Journal of Computational Physics, 227, 77917812, 2008.

[3] Chinesta, F., Ladevèze, P., and Cueto, E.. *A short review on model order reduction based on proper generalized decomposition*. Archives of Computational Methods in Engineering, 18(395-404), 2011.

[4] E. Pruliere, F. Chinesta, A. Ammar, *On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition*. Mathematics and Computers in Simulation, 81 791810, 2010.

[5] Carlberg, K., Bou-Mosleh, C., and Farhat, C. *Efficient nonlinear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations*. International Journal for Numerical Methods in Engineering, 2010.

[6] Ladevèze, P., Néron, D. and Passieux, J.-C. *On a multiscale computational mechanics with time-space homogenization*, In Multiscale methods - Bridging the scales in Science and Engineering., Oxford University Press. Ed. J. Fish, ch. Space Time Scale Bridging methods, 26(1), pp. 247282, 2009.

[7] Ladevèze, P. *A computational technique for the integrals over the time-space domain in connection with the LATIN method (in french)*. Tech. Rep. 193, LMT-Cachan, 1997

[8] Ladevèze, P., *Nonlinear Computationnal Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation.*, Springer Verlag, 1999.