

Abstract

Answering the need of an industrial looking for a model order reduction to use on a parametric problem, we offer to adapt the proper generalized decomposition (PGD) for its use. The goal of this work is to adapt the PGD to dynamics problems. The proper orthogonal decomposition (POD) will be used as a comparison tool for the PGD basis, as they are both reduced order modeling methods using separated variables. In order to use these methods on dynamics problems, a choice has to be made regarding the integration scheme, therefore this paper will present different possibilities and show what they offer. Results on academical problems of the method will be presented.

Keywords: Reduced Order Modeling, Separated variables representation, PGD, POD, dynamics, Reduced Basis.

1 Introduction

This work originated from an industrial need, regarding a time consuming simulation. The industrial runs a complex numerical model simulating the behavior of a car on the road. This non-linear slow dynamics problem takes a time to be solved measured in days, and it has to be ran numerous times to evaluate the car response for different sets of parameters. This is why a reduced order modeling method is needed. The PGD has been chosen since it allows to introduce parameters inside the problem, and provides a solution for all values of the parameters.

This work takes place among the model order reduction methods. The PGD method is the center of many current studies, applying its use to various cases. This paper will be centered on dynamics problems.

The need to be competitive brings more and more industrials to head towards simulation tools, which allow, at least during conception, to reduce the prototypes try-outs and therefore to decrease the time and cost of development.

It also enabled technical research offices to use optimization on conception and to produce probabilistic models. These two processes have in common their high cost in resources and calculation time, since both need the assessment of numerous solutions of the numerical model. That is why they are currently applied only to quite simple models in the industrial world. To deal with large problems, it is necessary to reduce calculation time while preserving a reasonable quality through model order reduction strategies.

Among the various methods of model order reduction existing, we offer to work essentially on the PGD [1, 2, 3]. Originally developed to solve evolution problems within the LATIN [4], this method has known numerous evolutions the past few years [5, 6]: Multiscale in space and time, stochastic use, parametric problems, "optimal" PGD base construction... This family of model order reduction using a representation with separation of variables is said *a priori*, since the reduced basis are built on the fly without the need for the exact solution of the problem (not even an approximation of it). The principle is to find the decomposition of the solution of a problem as a sum of products of separated variables functions (variables can represent a parameter of the problem [2]). In our case, the problem depends on the values of a set of parameters, and using the PGD ability to add parameters as variables, the resolution will provide a solution for all values of each parameter introduced this way.

In this paper, our goal is to apply this method to dynamics problem presenting non-linearities. We will provide a comparison with another model order reduction method, the POD [7]. This method will allow us to compare solutions but also reduced basis. The POD can be attached to the singular value decomposition (SVD) and as such it guarantees a form of optimality of the reduced basis, but this method has the major drawback to be *a posteriori*, implying that the method is applied to a previous resolution of the complete system. This means that a first resolution is required on the complete basis and that the acquired reduced basis is linked to this resolution.

An application of the PGD in the case of linear dynamics problems on academic examples will be presented comparing it to POD results. It will illustrate the performances and advantages of this method, showing the influence of the loading velocity, emphasising the results of integration scheme choice such as Galerkin time discontinuous [8] and pointing out the importance of orthogonalization.

1.1 POD

The proper orthogonal decomposition (POD [7]), known previously as principal component analysis (PCA [9]) or Karhunen-Loeve (K-L [10]) expansion, enables to extract from great quantities of data, the principal modes: the best representing the whole set of data. It is linked to the SVD, which could be seen as its discrete version. The SVD has for example been used in former face recognition programs.

1.1.1 Singular Value Decomposition

$\mathbf{A}(n \times m)$ being a matrix, there are $\mathbf{U}(n \times n)$, $\mathbf{V}(m \times m)$ orthogonal and $\mathbf{S}(n \times m)$ diagonal with coefficients zeros or positive numbers, such as:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad A_{ij} = \mathbf{u}_i \mathbf{S} \mathbf{v}_j^T \quad \text{where } \mathbf{u}_i \text{ et } \mathbf{v}_j \text{ are the lines of } \mathbf{U} \text{ and } \mathbf{V} \quad (1)$$

This decomposition is not unique, but for \mathbf{S} to be, it is chosen as a convention to order the values on its diagonal decreasingly: $S_{ii} \geq S_{(i+1)(i+1)}$. These values are called singular values and for a matrix \mathbf{A} having for rank r : $S_{ii} = 0 \forall i > r$. The decomposition can then be truncated at the order r (see below). Note: $r \leq \min(m, n)$.

These singular values are not without reminding the eigenvalues, and they are linked to them. It can be noted that singular values can be calculated for rectangular matrices but eigenvalues only exist for endomorphisms so they are limited to square matrices. The relation between the two decompositions is that \mathbf{V} contains the eigenvectors of $\mathbf{A}^T \mathbf{A}$ (symmetrically \mathbf{U} contains the eigenvectors of $\mathbf{A} \mathbf{A}^T$). And the squares of singular values of \mathbf{A} are the modulus of the eigenvalues of $\mathbf{A} \mathbf{A}^T$. If \mathbf{A} is symmetrical positive, its eigenvalues are its singular values and $\mathbf{U} = \mathbf{V}$. If \mathbf{A} is symmetrical but has some negative eigenvalues, then its singular values associated to these columns will be their absolute values.

1.1.2 Truncated SVD

The goal of this decomposition is to reduce the quantity of data keeping as good as possible the information it was containing. The left and right singular vectors (respectively the columns of \mathbf{U} and \mathbf{V}) give modes as representative of the whole set of data as they are associated with higher singular values. An order k is chosen to keep only the highest singular values.

$$\mathbf{A}^{(k)} = \mathbf{U} \mathbf{S}^{(k)} \mathbf{V}^T \quad A_{ij} = \mathbf{u}_i \mathbf{S}^{(k)} \mathbf{v}_j^T \quad (2)$$

where $\mathbf{S}_{ij}^{(k)} = S_{ij} \quad \forall i, j < k$, and zero above

this gives the approximated matrix $\mathbf{A}^{(k)}$, but then a big part of the system $\mathbf{U} \mathbf{S}^{(k)} \mathbf{V}^T$ is useless since it brings no contribution to the result because of multiplication with zeros of $\mathbf{S}^{(k)}$. Therefore each matrix can be truncated. Here is to visualize the k order truncation effect:

$$\begin{bmatrix} U_{11} & \cdots & U_{1k} \\ \vdots & \ddots & \vdots \\ U_{n1} & \cdots & U_{nk} \end{bmatrix} \quad (n \times k) \quad \begin{bmatrix} S_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S_{kk} \end{bmatrix} \quad (n \times m) \quad \begin{bmatrix} V_{11} & \cdots & V_{m1} \\ \vdots & \ddots & \vdots \\ V_{1k} & \cdots & V_{mk} \end{bmatrix} \quad (n \times m) \quad (3)$$

\mathbf{U} and \mathbf{V} are truncated after their k^{th} column, and \mathbf{S} after its k^{th} line and its k^{th} column, but $\mathbf{A}^{(k)}$ stays the same size as \mathbf{A} .

Obviously the quicker the singular values decrease the more efficient the truncation will be.

The SVD assures the optimality of the approximation, because the matrix obtained after truncation of order k , is the matrix of rank k which minimizes the error: $\|\mathbf{A} - \mathbf{A}^{(k)}\|_F$. The Frobenius norm is equal to the square root of the sum of all the squared terms of a matrix, which is also equal to the highest singular value.

The truncated SVD still needs the computation of the full SVD, which represents a number of numerical operations the same order as $O(mn^2)$, directly linked to the calculation time. And even if it is transposed to exchange the role of m and n , it represents a handicap which amplifies at the same time as the size of the studied matrix. To fight this issue, approximated decomposition methods have been developed, the low-rank matrix approximation method (LRMA), and particularly the QUIC-SVD, which allows to control the error caused by the approximation [11].

The POD is generally used in its discrete version, because data basis are discrete (like for example due to experimental sampling), or simply because of the discretization used by numerical solvers. A use of the continuous form can be found in [12], it will be described in the next part.

1.1.3 Continuous formulation

The POD is a method which offers a solution with separated variables in the form of a sum of product of functions. $z(x, t)$ will be approximated in this form:

$$z(x, t) \approx \sum_{k=1}^m a_k(t) \varphi_k(x) \quad (4)$$

In the rest of this paper, x will represent the space coordinate (so it can be a vector) and t the time coordinate. The decomposition 4 is not unique since the space containing the functions $\varphi_k(x)$ has to be chosen. For example if x is inside an interval of real numbers Ω , then the functions $\varphi_k(x)$ can be chosen as Fourier series, Legendre polynomials, Tchebychev polynomials, etc... And this choice influences obviously the result obtained for the functions $a_k(t)$. The hypothesis is made that the space approximation for the space functions is well chosen and that the sum will approach the solution $z(x, t)$ when m approaches infinity (more precisely on the definition domain of $z(x, t)$, apart from potential ensembles of measure zero, reader which are not familiar with measurement theories can read [13]. This consideration will not be developed here. The same goes for the questions of integrability of the functions, which is no problem in discrete calculation)

If for $\varphi_k(x)$ functions an orthonormal basis is chosen, we have:

$$\int_{\Omega} \varphi_i(x) \varphi_j(x) dx = \delta_{ij} \quad (5)$$

We obtain the following formula for $a_k(t)$:

$$a_k(t) = \int_{\Omega} z(x, t) \varphi_k(x) dx \quad (6)$$

then the determination of the function $a_k(t)$ only depends on $\varphi_k(x)$.

If the decomposition is optimal, i.e. if for each term of the sum there is no better approximation, then the terms are called proper orthogonal modes, and the sum proper orthogonal decomposition.

To match with the discrete formulation, we define:

$$\mathbf{A} = \mathbf{Q}\mathbf{V}^T = \sum_{i=1}^m \mathbf{q}_i \mathbf{v}_i^T \text{ where } \mathbf{Q} = \mathbf{U}\mathbf{S} \quad (7)$$

In this discrete form of the equation 4, the matrix \mathbf{A} represents $z(x, t)$ and the columns \mathbf{q}_i represent the functions $a_k(t)$. The link between singular values and eigenvalues has been described previously, and this connection can help solving this problem. We define $\mathbf{B} = \mathbf{A}^T \mathbf{A}$, which will give in the case of continuous time but discrete space representation:

$$\mathbf{B}_{ij} = \frac{1}{T} \int_{t_0}^{t_0+T} z(x_i, t) z(x_j, t) dt \quad (8)$$

And if we change to infinite dimension formulation, there are an infinity of eigenvalues and \mathbf{B} is no longer a matrix but a function of two variables:

$$B(x_1, x_2) = \frac{1}{T} \int_{t_0}^{t_0+T} z(x_1, t) z(x_2, t) dt \quad (9)$$

where x_1 and x_2 are continuous variables defined on Ω .

ψ being an eigenvector of \mathbf{B} , of eigenvalue associated λ , then $\mathbf{B}\psi = \lambda\psi$. Which gives in discrete form:

$$\sum_{j=1}^m \mathbf{B}_{ij} \psi_j^T = \lambda \psi_i \quad (10)$$

and becomes an integral in continuous form:

$$\int_{\Omega} B(x_1, x_2) \psi(x_2) dx_2 = \lambda \psi(x_1) \quad (11)$$

One of the problem of this model order reduction method *a posteriori* is that it depends on the solution obtain previously. Indeed the modes found are representing the original data, and if in the calculation, the configuration of the considered system should change, the modes would not be correctly representing the solution. The separated variables representation used by the POD also has limits, in particular the bad representation of a local perturbation that moves, typically a wave propagating (the solution of the d'Alembert equation is not with separated variables $f(x \pm \beta t)$).

1.2 PGD

Like the POD, the proper generalized decomposition (PGD [2]) approximate the solution, using the hypothesis of separated variables, as a sum of products of functions. A function $z(x, t)$ would be represented in the following form:

$$z(x, t) \approx \sum_{k=1}^m a_k(t) \varphi_k(x) \quad (12)$$

It was originally introduced by [1] within the LATIN method. It differs from the POD since the function $z(x, t)$ is unknown at the start. The modes are not known previously to the resolution, they are calculated "on the fly" thanks to an algorithm as such :

Algorithm 1 PGD Resolution

```

1: for  $m = 1$  to  $m_{max}$  do
2:   initialize  $a_1$ 
3:   for  $k = 1$  to  $k_{max}$  do
4:      $\varphi_k \leftarrow S_m(a_k)$ 
5:     norm  $\varphi_k$  ( $\|\varphi_k\|_\Omega = 1$ )
6:      $a_k \leftarrow T_m(\varphi_k)$ 
7:     check the convergence of  $(\varphi_k a_k)$ 
8:   end for
9:    $\varphi_m \leftarrow \varphi_k$  and  $a_m \leftarrow a_k$ 
10:   $z_m = z_{m-1} + \varphi_m a_m$ 
11: end for

```

Where S_m is the application associating to a time function a , a space function φ . And T_m the application associating to a space function φ , a time function a . To find them we start with the weak formulation of the problem:

$$B(u, v) = L(v) \quad \forall v \in H \quad (13)$$

Where H is a Hilbert space, B a bilinear form, continuous and coercive on H and L a linear form continuous on H .

A classical process is applied for the PGD, called Galerkin progressive: z_{m-1} (the approximation of z at the order $m - 1$) is supposed known. The goal is to find a new couple $(\varphi, a) \in U \times V$. It is defined as the one verifying the criterion of double orthogonality of Galerkin, i.e.:

$$B(z_{m-1} + \varphi a, \varphi^* a) = L(\varphi^* a) \quad \forall \varphi^* \in U \text{ et } \forall a^* \in V \quad (14)$$

Then $S_m(a)$ is defined by:

$$B(z_{m-1} + \varphi a, \varphi^* a) = L(\varphi^* a) \quad \forall \varphi^* \in U \quad (15)$$

and $T_m(\varphi)$ by:

$$B(z_{m-1} + \varphi a, \varphi a^*) = L(\varphi a^*) \quad \forall a^* \in V \quad (16)$$

The couple (φ, a) verifies the equation 14 if and only if $\varphi = S_m(a)$ and $a = T_m(\varphi)$, which is a non-linear problem, of which the resolution by the PGD can be seen as an eigenvalues problem where φ and a are respectively the dominant eigenfunctions of $S_m \circ T_m$ and $T_m \circ S_m$. Allowing to draw our inspiration from algorithms used in eigenvalues problems and giving the algorithm 1, which provides the decomposition. It can be modified, in particular to add the orthogonalization which can improve the quality of the basis found (it works as long as there are only two variables). If we want to orthogonalize φ_m with the existing basis $\{\varphi_1, \varphi_2, \dots, \varphi_{m-1}\}$, a dot product needs to be defined, for example such as:

$$\langle u, v \rangle_\Omega = \int_\Omega uv \, d\Omega \quad (17)$$

2 Dynamics

The fundamental equation of dynamics is verified at any time:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}$$

It is a differential equation in \mathbf{u} . In order to solve a differential equation there are different methods. Ideally there is one that gives an exact and explicit solution, but it is limited to a few classical cases. But in general cases differential equations are solved approximately and numerically. It relies on the discretization of the response domain. In space it means creating a mesh, allowing to define quantities depending on space as matrices. Time representing only one dimension, a simple choice of a fixed time step to slice the time domain, enables to represent it more or less precisely.

2.1 Integration scheme

The discretization of the time axis allows to solve the equation at a given instant (provided that the solution is known at the previous instant*). In order to do so, a choice of a differentiation method has to be made to derive \mathbf{u} , therefore a numerical scheme is chosen.

*If a scheme only uses the solution at the previous time step and not the ones before it, it is said to use a unique step, otherwise it is said to use multiples steps.

In order to describe the differentiation of a discrete function, we can use the Euler scheme, Runge-Kutta's or Crank-Nicholson's. In the particular case of dynamics the Newmark method is used to define the links between acceleration, velocity and displacement. These links are defined by two equations using parameters which can make the algorithm explicit or implicit, and influence its stability.

A scheme is said to be stable if it can control the accumulation of calculation error. Some schemes are conditionally stable and the solution they provide can diverge. The condition of stability in dynamics is generally a condition on the discretization of time: choosing a time step that is not too small to define the propagation of information in the mesh elements. The critical size above which the time step must not be, is therefore depending on the size of the mesh elements.

2.1.1 Newmark method

The method introduced by [14] gives an integration scheme based on two equations:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + dt \cdot \dot{\mathbf{u}}_n + \frac{dt^2}{2} [(1 - 2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}] \quad (18)$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + dt[(1 - \gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}] \quad (19)$$

The scheme stability depends on the two parameters β and γ :

$$\begin{array}{ll} \gamma \leq 1/2 & \text{Unstable} \\ \gamma \geq 1/2 \text{ et } 2\beta \leq \gamma & \text{Conditionally stable} \\ \gamma \geq 1/2 \text{ et } 2\beta \geq \gamma & \text{Unconditionally stable} \end{array}$$

For particular sets of values we find the usual methods:

Name	γ	β	Properties
Centered Differences	1/2	0	Explicit and Conditionally stable
Fox Goodwin	1/2	1/12	Conditionally stable
Linear Acceleration	1/2	1/6	Conditionally stable
Average Acceleration	1/2	1/4	Unconditionally stable

The equation of dynamics can be written this way:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \quad (20)$$

Which gives, when associated with equations 18 and 19, if \mathbf{K} does not depend on \mathbf{u} :

$$\begin{array}{c} \overbrace{(\mathbf{M} + \mathbf{C}\gamma dt + \mathbf{K}\beta dt^2)}^{\mathbf{S}} \ddot{\mathbf{u}}_{n+1} = \\ \mathbf{f}_{n+1} - \mathbf{C} \overbrace{[\dot{\mathbf{u}}_n + dt(1 - \gamma)\ddot{\mathbf{u}}_n]}^{\dot{\mathbf{u}}_{n+1}^p} - \mathbf{K} \overbrace{[\mathbf{u}_n + dt\dot{\mathbf{u}}_n + \frac{dt^2}{2}(1 - 2\beta)\ddot{\mathbf{u}}_n]}^{\mathbf{u}_{n+1}^p} \end{array} \quad (21)$$

Allowing to find in three steps, the values of $\ddot{\mathbf{u}}$, $\dot{\mathbf{u}}$ and \mathbf{u} :

$$\begin{cases} \ddot{\mathbf{u}}_{n+1} = \mathbf{S}^{-1}(\mathbf{f}_{n+1} - \mathbf{C}\dot{\mathbf{u}}_{n+1}^p - \mathbf{K}\mathbf{u}_{n+1}^p) \\ \dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_{n+1}^p + dt\gamma\ddot{\mathbf{u}}_{n+1} \\ \mathbf{u}_{n+1} = \mathbf{u}_{n+1}^p + dt^2\beta\ddot{\mathbf{u}}_{n+1} \end{cases} \quad (22)$$

2.1.2 HHT

The Hilber-Hughes-Taylor (HHT or HHT- α [15]) method, which is based on the Newmark method, was introduced in order to add a parameter α which enables to put in place a controlled numerical dumping. The integration scheme is still defined by equations 18 and 19, but there is one more parameter to use, which can reduce the choice of value to α :

$$\begin{cases} \beta &= (1 - \alpha)^2/4 \\ \gamma &= 1/2 - \alpha \\ \alpha &\in [-1/3; 0] \end{cases} \quad (23)$$

The equation of dynamics given in [16] is formulated as such:

$$\mathbf{M}\ddot{\mathbf{u}}_{n+1} + (1 + \alpha)\mathbf{C}\dot{\mathbf{u}}_{n+1} - \alpha\mathbf{C}\dot{\mathbf{u}}_n + (1 + \alpha)\mathbf{K}\mathbf{u}_{n+1} - \alpha\mathbf{K}\mathbf{u}_n = (1 + \alpha)\mathbf{f}_{n+1} + \alpha\mathbf{f}_n \quad (24)$$

Which gives, if \mathbf{K} does not depend on \mathbf{u} :

Allowing to calculate in three steps:

$$\begin{cases} \ddot{\mathbf{u}}_{n+1} &= \mathbf{S}_{\text{HHT}}^{-1}(\mathbf{f}_{n+1}^{\text{HHT}} - \mathbf{C}\dot{\mathbf{u}}_{n+1}^{\text{pHHT}} - \mathbf{K}\mathbf{u}_{n+1}^{\text{pHHT}}) \\ \dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + dt[(1 - \gamma)\ddot{\mathbf{u}}_n + \gamma\ddot{\mathbf{u}}_{n+1}] \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + dt.\dot{\mathbf{u}}_n + \frac{dt^2}{2}[(1 - 2\beta)\ddot{\mathbf{u}}_n + 2\beta\ddot{\mathbf{u}}_{n+1}] \end{cases} \quad (25)$$

with:

$$\begin{aligned} \mathbf{f}_{n+1}^{\text{HHT}} &= (1 + \alpha)\mathbf{f}_{n+1} - \alpha\mathbf{f}_n \\ \mathbf{S}_{\text{HHT}} &= \mathbf{M} + (1 + \alpha)(\mathbf{C}dt\gamma + \mathbf{K}dt^2\beta) \\ \dot{\mathbf{u}}_{n+1}^{\text{pHHT}} &= \dot{\mathbf{u}}_n + (1 + \alpha)dt(1 - \gamma)\ddot{\mathbf{u}}_n \\ \mathbf{u}_{n+1}^{\text{pHHT}} &= \mathbf{u}_n + (1 + \alpha)(dt\dot{\mathbf{u}}_n + \frac{dt^2}{2}(1 - 2\beta)\ddot{\mathbf{u}}_n) \end{aligned} \quad (26)$$

2.1.3 Time discontinuous Galerkin

This method allows to introduce some discontinuity in the displacement or velocity field. In order to introduce such discontinuity in time, the quantities must be defined so that at each instant t_n they can have two values. The quantities will be represented as such:

$$\begin{aligned} \mathbf{a}_n^- &= \lim_{\varepsilon \rightarrow 0^-} \mathbf{a}(t_n + \varepsilon) \\ \mathbf{a}_n^+ &= \lim_{\varepsilon \rightarrow 0^+} \mathbf{a}(t_n + \varepsilon) \end{aligned} \quad (27)$$

The discontinuity introduced here would be:

$$[\mathbf{a}_n] = \mathbf{a}_n^+ - \mathbf{a}_n^- \quad (28)$$

It could be represented like in the Figure 1.

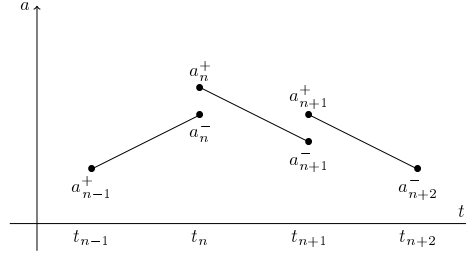


Figure 1: Representation of a variable allowing discontinuities

We need to go back to the fundamental equation of dynamics, and we introduce \mathbf{v} in the following way:

$$\begin{aligned} \mathbf{M} \dot{\mathbf{v}} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{u} &= \mathbf{f} \\ \dot{\mathbf{u}} - \mathbf{v} &= 0 \end{aligned} \quad (29)$$

In order to have this weak formulation:

$$\begin{aligned} \forall \mathbf{w}_1, \forall \mathbf{w}_2 \quad & \int_{t_m}^{t_{m+1}} \mathbf{w}_1 (\mathbf{M} \dot{\mathbf{v}} + \mathbf{C}\mathbf{v} + \mathbf{K}\mathbf{u} - \mathbf{f}) dt + \int_{t_m}^{t_{m+1}} \mathbf{w}_2 \mathbf{K} (\dot{\mathbf{u}} - \mathbf{v}) dt \\ & + \mathbf{w}_2(t_m) \mathbf{K}[\mathbf{u}]_m + \mathbf{w}_1(t_m) \mathbf{M}[\mathbf{v}]_m = 0 \end{aligned} \quad (30)$$

Choosing \mathbf{w}_1 and \mathbf{w}_2 among linear functions we can obtain the following system:

$$\begin{aligned} & \begin{bmatrix} \mathbf{K} & 0 & -\mathbf{K}\frac{\Delta t}{6} & \mathbf{K}\frac{\Delta t}{6} \\ 0 & \mathbf{K} & -\mathbf{K}\frac{\Delta t}{2} & -\mathbf{K}\frac{\Delta t}{2} \\ 0 & 0 & \mathbf{K}\frac{(\Delta t)^2}{3} + \mathbf{C}\frac{\Delta t}{2} & \mathbf{K}\frac{(\Delta t)^2}{6} + \mathbf{M} + \mathbf{C}\frac{\Delta t}{2} \\ 0 & 0 & \mathbf{K}\frac{(\Delta t)^2}{12} - \mathbf{M}\frac{1}{2} & \mathbf{K}\frac{(\Delta t)^2}{12} + \mathbf{M}\frac{1}{6} + \mathbf{C}\frac{\Delta t}{6} \end{bmatrix} \begin{bmatrix} \mathbf{u}_m^+ \\ \mathbf{u}_{m+1}^- \\ \mathbf{v}_m^+ \\ \mathbf{v}_{m+1}^- \end{bmatrix} \\ & = \begin{bmatrix} \mathbf{K}\mathbf{u}_m^- \\ \mathbf{K}\mathbf{u}_m^- \\ (\mathbf{M}\mathbf{v}_m^- + \frac{\Delta t}{2}(\mathbf{f}_m + \mathbf{f}_{m+1})) - \Delta t \cdot (\mathbf{K}\mathbf{u}_m^-) \\ -\frac{\Delta t}{6}(\mathbf{K}\mathbf{u}_m^- - \mathbf{f}_{m+1}) - \frac{1}{3}\mathbf{M}\mathbf{v}_m^- \end{bmatrix} \end{aligned} \quad (31)$$

Using only the two last lines, the velocity can be determined from the quantities of the previous state. And another step allows us to calculate the displacement using the two first lines. Here [17] is an example of the use of the time discontinuous Galerkin scheme with the PGD.

2.1.4 Comparisons

For reasons that will be detailed later on, it can be interesting to avoid the presence of high frequency phenomena. And unfortunately the use of an integration scheme can introduce such phenomena which are a purely numerical effect, depending on the time step Δt .

The Figure 2 presents the dynamics problem used to obtained the results hereafter. It is a beam fixed on one side solicited in traction/compression, by an effort depending

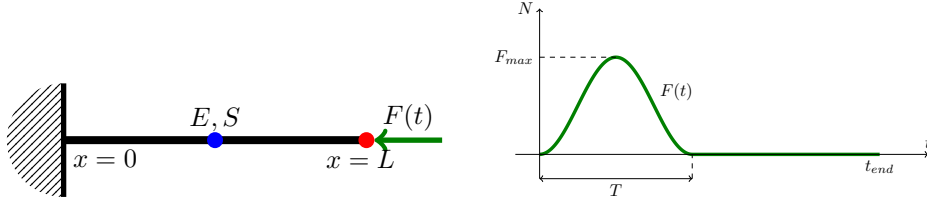


Figure 2: Problem studied

on time. The graph on the right represents the evolution of the effort in time, being a sinus verse for one period then staying at zero. The value F_{max} is always chosen the same. The period is given by the parameter T , which is used to change the solicitation. The shorter T is, the more abrupt is the loading, the closer it gets to an impulsions.

In the Figure 3 are displayed on each graph three curves. They are linked to the resolution of the beam problem. The red one represents the displacement at the end of the beam, the blue one the displacement in the middle and the green one represents the loading (disregarding the unities). An unconditionally stable scheme is used here: Newmark's average acceleration, and what changes is the loading velocity. Since the loading is just one period of a sinus verse, the velocity is controlled by the period T . And we can see that, the more violent the loading is, the more perturbations appear.

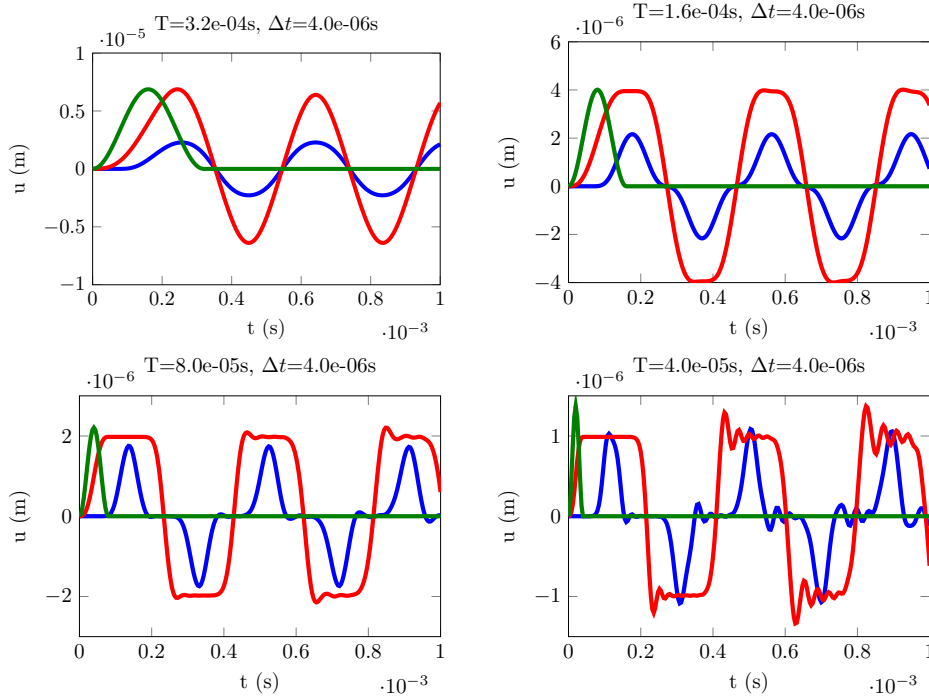


Figure 3: Influence of loading velocity on numerical integration
Newmark scheme - Average acceleration
Variable T fixed Δt

The Figure 4 shows that the perturbation is indeed a numerical effect. The integra-

tion scheme is the same as previously but what changes is the length of the time step. We can see that the shorter the time step is (the more precisely time represented), the less perturbation there are for a given loading. Increasing time representation precision to avoid perturbation presents drawbacks. First it increases as well the calculation cost, second it links the velocity of the loading to the time step, which supposes the velocity is previously known and supposes that the velocity is independent from the time step too (which can be false in worst-cases scenarios such as choc).

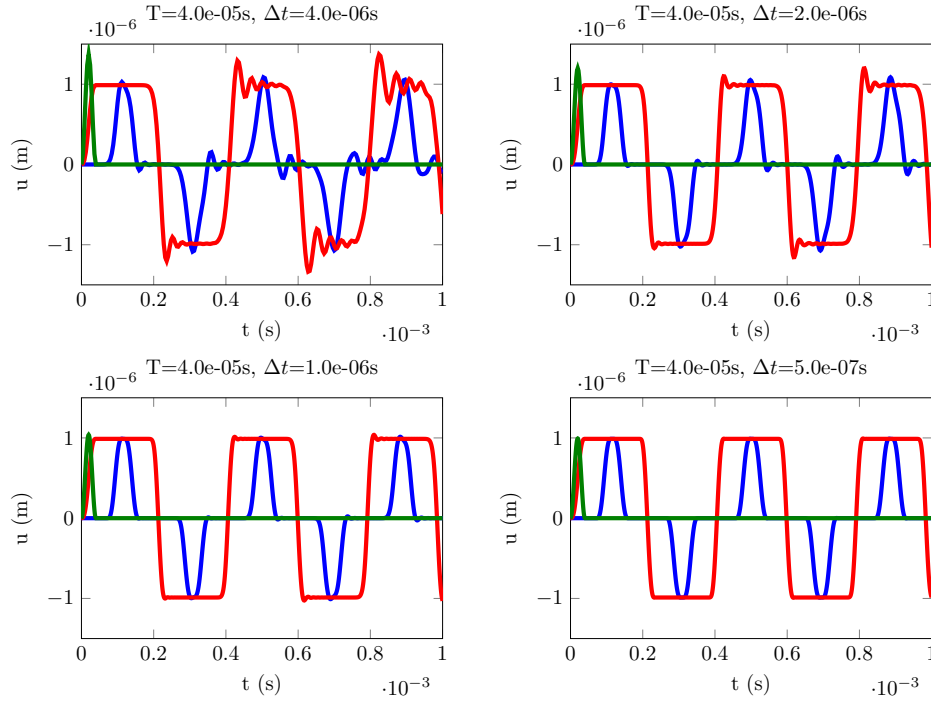


Figure 4: Influence of Δt on numerical integration,
Newmark scheme - Average acceleration
Fixed T variable Δt

In order to avoid or reduce the presence of such perturbation we can change the integration scheme. The curves in the Figure 5 represent the same as previously, but this time the loading and the time step stay the same and only the integration scheme changes. On the first graph then Newmark method : average acceleration is used. On the second it still is Newmark but it is called average acceleration modified, which can introduce dumping when the previous one is conservative. It does avoid to introduce high frequency perturbation but it changes the global solution by introducing a dumping effect regardless of the dumping defined for the mechanical system. The third graph is obtained with the HHT scheme, it allows to reduce the dumping effect to high frequency. Unfortunately the frequencies affected are linked to the time step which can be set by other parameters, moreover the result is a compromise between the two previous solutions, and it has just faded the perturbation slightly. Finally on the fourth part is the result given by the time discontinuous Galerkin method. It seems

to be the less perturbed solution and the closest to what was obtained by increasing the time representation precision.

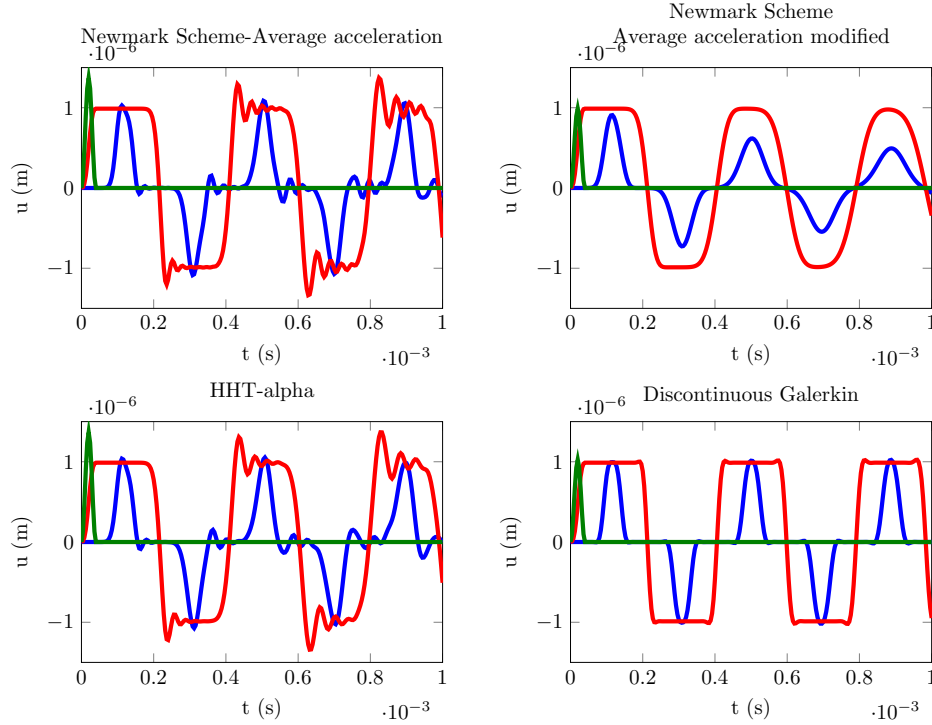


Figure 5: Use of different integration schemes
Fixed $T=4e-5s$ fixed $\Delta t=4e-6$

2.2 PGD

2.2.1 Problem definition

- Kinematic admissibility: $U \in [H^1(\Omega)]^3, U|_{\partial\Omega_U} = U_d$
- Static admissibility: $\text{div}\sigma + f_v = \rho\ddot{U} + \mu\dot{U}$ on Ω and $\sigma.n|_{\partial\Omega_f} = f_d$
- Behavior relation: $\sigma = H : \varepsilon(U)$ on Ω
- Initial conditions: $U(0) = U_0$ and $\dot{U}(0) = \dot{U}_0$

Variational formulation:

$$\begin{aligned} \forall U^* \text{ KA to } 0, \quad & \int_{\Omega} [\varepsilon(U^*) : \sigma + U^* \mu \dot{U} + U^* \rho \ddot{U} - U^* f_d] d\Omega \\ & - \int_{\partial\Omega_f} U^* f_d ds - \underbrace{\int_{\partial\Omega_U} U^* \sigma.n ds}_{=0} = 0 \end{aligned} \quad (32)$$

Where

- U is displacement
- Ω space, with the boundary $\partial\Omega = \partial\Omega_U \cap \partial\Omega_f$
- σ stress tensor
- ε strain tensor
- ρ volumetric mass density
- μ volumetric dumping (it is generally badly known)
- H Hook tensor
- f_v volumetric force
- f_d load on $\partial\Omega_f$

We use the PGD representation with separated variables:

$$U(x, t, \theta) = \sum_{k=1}^n \varphi_k(x) g_k(t) h_k(\theta) \quad (33)$$

Introducing this form in the problem defined in the equation 32, we can obtain the three separated problems that will be solved each at a time for every iteration of the fixed point. These problems are described above.

Let's discretize space to introduce the usual notations.

$$\varphi(x) = \sum_{i=1}^{Nb_{cx}} N_i(x) \varphi_i = N_\varphi(x) \varphi_q \quad (34)$$

In the variational formulation is replaced, U^* by $(\varphi gh)^* = (\varphi^* gh + \varphi g^* h + \varphi gh^*)$.

And to simplify equation the usual notations for quantities integrated on the space domain are defined, \mathbf{K} for the stiffness, \mathbf{C} for dumping, \mathbf{M} for the mass and \mathbf{f} for the effort.

And the result with all calculations done is : $\forall \varphi_q^*, \forall g^*, \forall h^*$

$$\begin{aligned} \int_T \int_\Theta (\varphi_q^* gh + \varphi_q g^* h + \varphi_q gh^*)^T & \left[\mathbf{K} \varphi_q gh + \mathbf{C} \varphi_q \dot{g} h + \mathbf{M} \varphi_q \ddot{g} h \right. \\ & + \mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k \\ & \left. + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f} \right] dt d\theta = 0 \end{aligned} \quad (35)$$

From which a problem is extracted for each variable.

2.2.2 Space problem

$$\begin{aligned} & \int_T \int_\Theta gh [\mathbf{K} gh + \mathbf{C} \dot{g} h + \mathbf{M} \ddot{g} h] dt d\theta \varphi_q \\ = & - \int_T \int_\Theta gh [\mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k \\ & + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f}] dt d\theta \end{aligned} \quad (36)$$

So here is basically a problem in the form of linear system : $\mathbf{A} \times \mathbf{x} = \mathbf{b}$

2.2.3 Time problem

$$\begin{aligned}
& \int_{\Theta} \varphi_q^T h [\mathbf{K} \varphi_q h] d\theta \, g + \int_{\Theta} \varphi_q^T h [\mathbf{C} \varphi_q h] d\theta \, \dot{g} \\
& + \int_{\Theta} \varphi_q^T h [\mathbf{M} \varphi_q h] d\theta \, \ddot{g} \\
= & - \int_{\Theta} \varphi_q^T h [\mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k \\
& + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f}] d\theta
\end{aligned} \tag{37}$$

What we find here is a differential equation in g , that can be solved like a classical dynamics problem, i.e. using an integration scheme like the ones discussed in the preceding section.

2.2.4 Parameter problem

$$\begin{aligned}
& \int_T \varphi_q^T g [\mathbf{K} \varphi_q g + \mathbf{C} \varphi_q \dot{g} + \mathbf{M} \varphi_q \ddot{g}] dt \, h \\
= & - \int_T \varphi_q^T g [\mathbf{K} \sum_{k=1}^{n-1} (\varphi_q)_k g_k h_k + \mathbf{C} \sum_{k=1}^{n-1} (\varphi_q)_k \dot{g}_k h_k \\
& + \mathbf{M} \sum_{k=1}^{n-1} (\varphi_q)_k \ddot{g}_k h_k - \mathbf{f}] dt
\end{aligned} \tag{38}$$

And finally, in this simple example where \mathbf{K} , \mathbf{C} and \mathbf{M} do not depend on the parameter, we find a scalar equation.

2.3 Application onto academical problems

2.3.1 POD

As said previously, the POD relies on a preceding resolution of a problem on the complete basis. The time sample on which the problem on complete basis is ran, is called snapshot. The reduce order modeling brought by the POD method allows to evaluate the system's response either on numerous similar problems or on the same problem as the snapshot but on a longer time. The snapshot, involving the model before reduction, can represent a heavy calculation, which would push toward a snapshot as short as possible. But its length is also linked to the quality of the modes found by the POD, which limits the will to shorten it. Here it was chosen long enough to capture several oscillations of the first mode. Beside its cost, the preceding resolution represents another drawback. Because it links the POD modes to a problem, and we have observed that changing parameters of the problem may reduce significantly the quality of the solution found. The POD being used here to give a comparison element for the PGD, its results will not be further developed.

2.3.2 PGD

The Figure 6 graphs are plotting the error of the solution found by the PGD to dynamics problems described Figure 3. What can be noticed is that the error level decreases slower as the loading is getting more "violent". Indeed the more violent the loading is, the closer it gets to a choc problem. Where the solution being a wave going through the material, it is not well represented by separated variables such as the PGD representation uses. It is showing a weakness of the PGD using a fixed point method to calculate new modes (which present an uncertainty of convergence), since we can see on the fourth graph that the method diverges. To avoid this possible problem the PGD can use another method called minimization [18] that is assured to converge but is more costly in calculation.

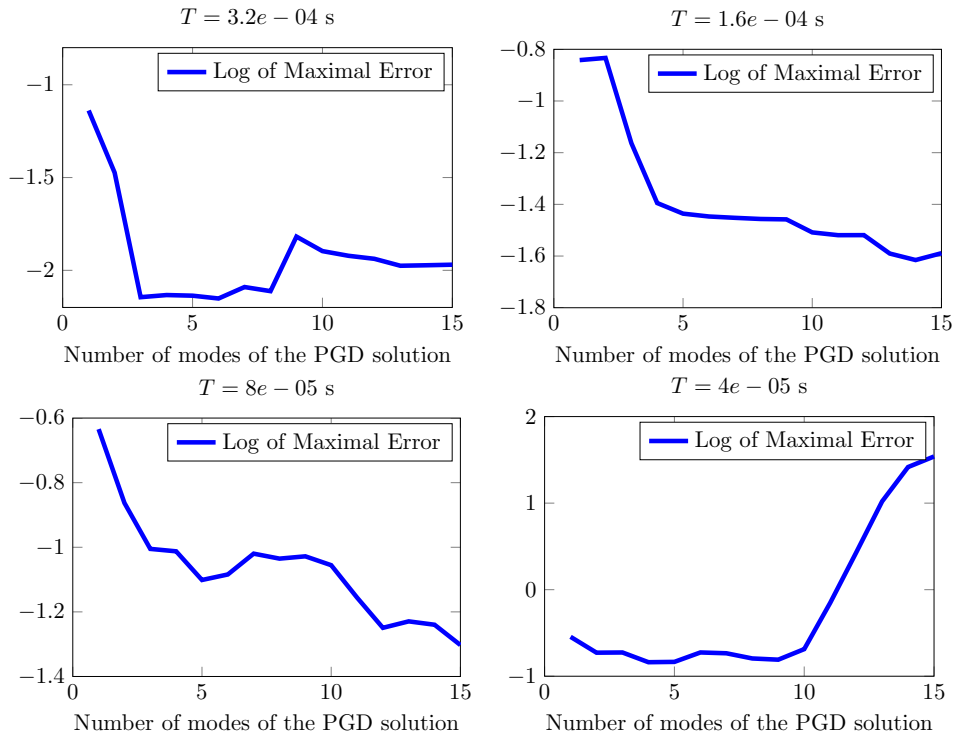


Figure 6: Influence of loading velocity on PGD convergence

2.3.3 Comparisons

On the Figure 7 is displayed the evolution of an error indicator : the maximum of the local difference between the resolution and a reference solution, which can be written : $e_{Max} = \left| \frac{\max(s_{Ref} - s_{Cat})}{\max(s_{Ref}) - \min(s_{Ref})} \right|$. The reference solution here is the solution obtained classically, without a model order reduction method.

It can be noticed on the left graph of Figure 7, that not every PGD mode brings the same contribution to the solution. When the right hand side graph shows for the POD

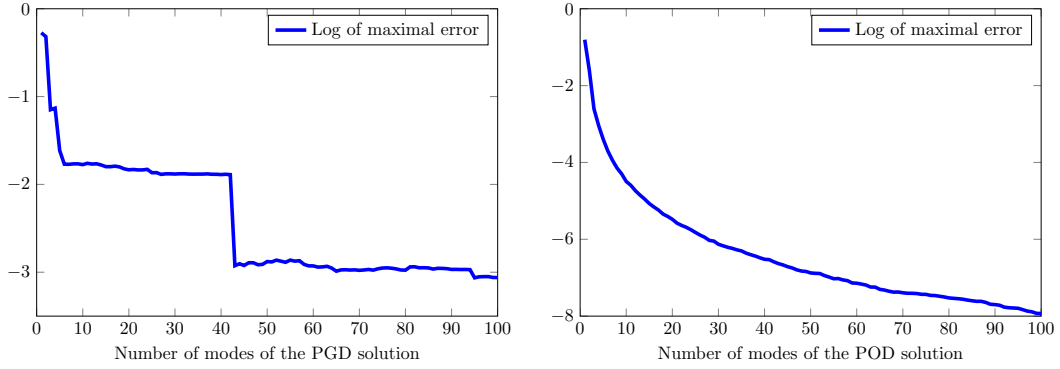


Figure 7: Evolution of an error indicator while increasing the number of modes

an error decreasing continuously and quickly, the PGD encounters stages. The aspect of the POD graph reminds the optimality of the basis it gives, and the speed at which the error decreases is linked to the way singular values decrease. Now this graph is not discarding PGD against POD, it should be kept in mind that adding a mode to the PGD is just run another fixed point, when each point of the POD graph here has needed a separate resolution. The presence of stages on the PGD graph suggests that several modes produced by the method do not contribute much to the solution, probably because those modes must be highly correlated with preceding ones. And eventually, every once in a while a new mode is found that is not correlated to the preceding ones, and the error decreases drastically.

To see how the modes are correlated to each other, we can visualize the Modal Assurance Criterion (MAC [19]) analysis of a family of modes. In the case of the POD it would be useless since as the name of the method says the modes given are orthogonal to each other. On the contrary, visualizing the MAC analysis of the PGD modes reveals how much they are correlated and reinforce the idea of a stagnation of the method by finding new modes that could be expressed by a combination of the previous ones. What is shown on Figure 8 is actually the MAC analysis allowing to visualize the correlation between the modes obtained by PGD and those obtained by POD. What we see is that on the hundred modes found here by the PGD, they only correspond to the first dozen of the ones found by the POD (Reminder : when the MAC value is 1 the correlation is total, i.e. the modes are identical.). In numerous cases in the use of the PGD method, the modes can be orthogonalized, which would prevent those correlations. But unfortunately, the use of more than two variables in the resolution (like the addition of parameters), makes orthogonalization difficult or practically impossible, so it will not be implemented here.

3 Conclusion

The results presented showed that the PGD without the use of orthogonalization can present stagnation periods in the evolution of the solution, due to the creation of highly

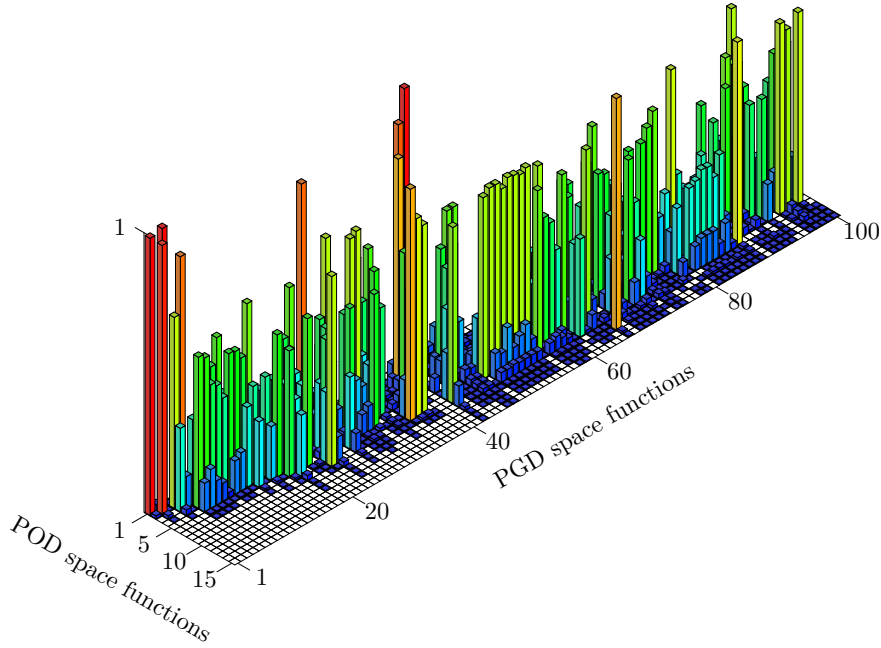


Figure 8: Correlation between POD and PGD modes

correlated modes. It was also highlighted that PGD presents a weakness, the more violent the loading gets the more modes are needed to reach a fixed quality for the solution. Moreover when high frequencies are appearing in transient dynamics the PGD can diverge, due to an inappropriate representation of waves with separated variables. And if we want to avoid to introduce perturbation that could cause convergence issues, the choice of an appropriate integration scheme must be done, depending on the problem treated. It seems that time discontinuous Galerkin is the one supporting the most abrupt cases. But as the industrial problem from which the project originated is in slow dynamics, it is not for sure the most relevant scheme to be used here. The other way to prevent divergence is to use the minimization method which will be used in the project in the future.

3.1 Prospect

This work is part of a PhD, which takes place inside the MECASIF project, and its goal is to add the possibility to solve cases presenting localized non-linearities using the PGD. It is a real challenge because of the peculiar way the PGD finds a solution, and because the non-linearities will induce intrusive changes in the current program. The introduction of parameters is also planned in the near future, to treat the industrial case described in the introduction.

Acknowledgment

This work was carried out as part of Project FUI AAP15 MECASIF supported by the French "Fond Unique Interministriel".

References

- [1] P. Ladevèze, *Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation*, Springer Berlin, New York, 1999.
- [2] F. Chinesta, A. Ammar, E. Cueto, "Recent Advances And New Challenges In the Use Of The Proper Generalized Decomposition For Solving Multidimensional Models", *Archives of Computational Methods in Engineering*, 17: 327–350, 2009.
- [3] F. Chinesta, P. Ladeveze, E. Cueto, "A Short Review on Model Order Based on Proper Generalized Decomposition", *Computational Methods in Engineering*, 18(4): 395–404, 2011.
- [4] P. Ladevèze, "The large time increment method for the analyze of structures with nonlinear constitutive relation described by internal variables", *CR Acad Sci Paris*, 309(1095-1099), 1989.
- [5] N. Relun, D. Neron, P.A. Boucard, "A model reduction technique based on the PGD for elastic-viscoplastic computational analysis", *Computational Mechanics*, 51: 83–92, 2013.
- [6] C. Heyberger, P.A. Boucard, D. Neron, "A rational strategy for the resolution of parametrized problems in the PGD framework", *Computer Methods in Applied Mechanics & Engineering*, 259: 40–49, 2013.
- [7] A. Chatterjee, "An introduction to the Proper Orthogonal Decomposition", *Current science*, 78(7): 808–817, 2000.
- [8] W.H. Reed, T.R. Hill, "Triangular mesh methods for the neutron transport equation", *Tech. Report LA-UR-73-479*, 1973, Los Alamos Scientific Laboratory.
- [9] I.T. Jolliffe, *Principal Component Analysis*, Springer, Second edition, 2002.
- [10] J. Kittler, P.C. Young, "A new approach to feature selection based on the Karhunen-Loeve expansion", *Pattern Recognition*, 5(4): 335–352, 1973.
- [11] P. Holmes, A. Gray, C.I. Jr., "QUIC-SVD: Fast SVD using cosine trees", *NIPS*, pages pp 673–680, 2008.
- [12] P. Holmes, J. Lumley, G. Berkooz, "Turbulence, Coherent Structures, Dynamical Systems and Symmetry", *Cambridge Monographs on Mechanics*, 1996.

- [13] W. Rudin, *Principles of Mathematical Analysis*, Volume 3, McGraw-Hill New York, 1976.
- [14] N.M. Newmark, “A method of computation for structural dynamics”, *Journal of Engineering Mechanics*, 85: 67–94, 1958.
- [15] H. Hilber, T. Hughes, R. Taylor, “Improved Numerical Dissipation for the time integration algorithms in structural dynamics”, *Earthquake engineering and structural dynamics*, Vol 5: pp 283–292, 1977.
- [16] C. Chen, J. Ricles, “Analysis of implicit HHT-alpha integration algorithm for real-time hybrid simulation”, *Earthquake engineering and structural dynamics*, 41: pp 1021–1041, 2012.
- [17] L. Boucinha, A. Gravouil, A. Ammar, “Space–time proper generalized decompositions for the resolution of transient elastodynamic models”, *Computer Methods in Applied Mechanics and Engineering*, 255: 67–88, 2013.
- [18] A. Nouy, “A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations”, *Computer Methods in Applied Mechanics and Engineering*, 199(23): 1603–1626, 2010.
- [19] R.J. Allemang, “The modal assurance criterion—twenty years of use and abuse”, *Sound and vibration*, 37(8): 14–23, 2003.