COMPDYN 2015
5th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, V. Papadopoulos, V. Plevris (eds.)
Crete Island, Greece, 25–27 May 2015

SVD FAMILY FOR ANALYSING CAR CRASH BEHAVIOUR

T.T.T. VUONG [1,2], L. JEZEQUEL[1] and Y. TOURBIER[2]

[1]LTDS, Ecole Centrale de Lyon
36 avenue Guy de Collongue, 69134 Ecully Cedex, France
e-mail: louis.jezequel@ec-lyon.fr

[2] Renault S.A.S, 1 avenue du Golf, 78084 Guyancourt cedex, France
e-mail: {thuy-thithanh.vuong,yves.tourbier}@renault.com

Keywords: Reduced Order Model, Optimization, Crash, Singular Value Decomposition, Generalized Singular Value Decomposition, Higher-Order Generalized Singular Value Decomposition, Proper Orthogonal Decomposition, a posteriori, non-intrusive

Abstract. *The car crash simulation describes the deformation of the car structure under high velocity (up to 65 km/h) during a very short simulation time (about 100 ms). The simulation represents many complexities of the finite element method in structural analysis: (i) high number of degrees of freedom (from 5 to 20 M finite elements for a whole vehicle) in order to well take into account of the geometrical and material nonlinearities; (ii) use of direct time-integration, explicit algorithm, which needs very small time step (due to Courant's condition) and consequently long time simulation (till* 24 *hours); (iii) high performance software needed (solver). Furthermore, the crash simulation is transient during all simulation; the contact management is sophisticated and the results are scattered. We are interested in elaborating a reduced crash model in order to cut down the total cost of an optimization study. The method should be a posteriori allowing predicting a new simulation interpolated between existing simulation(s) in a Design Of Experiments (DOE). We choose methods in the Singular Value Decomposition (SVD) family in order to find temporal correlation among crash data in one simulation, spatial and parametric correlation among crash data in several simulations. Two almost industrial use cases (BoxBeam — representing the most important car part for energy absorption and TBeam — representing the engine compartment) are tested with Matlab software.*

1 INTRODUCTION

In the design of a vehicle structure, the crash requirement for the security of occupants must be satisfied before any other requirements such as Durability or Noise Vibration Harshness (NVH). The simple Lumped Mass Spring (LMS) model of which properties are deduced from static crush tests, like Kamal's model [1], was firstly used for a crash simulation. Then, the multibody model MADYMO [2] was developed for both car parts and occupants [3]. Since 20 years, thanks to advances of computer technologies, the nonlinear FE model including geometries and material laws of vehicle parts become more and more popular. It becomes the high-fidelity model because it is believed to be the best approximated model for a car simulation : the physical phenomena are well modeled in particularly at the local level such as plasticity or crack. The crash solver at Renault is Pam-Crash developed by ESI Group. It uses the explicit algorithm which demands very small time steps, about μs (due to the stability condition – Courant's condition) so long time simulation (till 24 hours). As a result, the system is considered in balanced condition for each step. The mass matrix modeled by lumped-mass is already diagonal so there is no need to inverse the mass matrix during in the simulation. That makes the important difference to implicit algorithm.

Renault uses multi-objectives optimization in order to design the vehicle : there are a lot of parameters in a vehicle such as the shapes, the material laws and so on ; engineers must choose the parameters which match the technical specifications (crash, NVH ...) and some objectives such as mass minimization. While searching the best solution for an optimization study with the FE model, it is interesting to increase the number of parameters in order to improve the quality of the optimal solution but it rises dramatically the number of simulations. Nevertheless, it is not possible for the industrial context because of three reasons :

— Firstly, we will not transform our model to a simplified model like LMS or MADYMO because it will loose the "high-fidelity".

— Secondly, methods for parameters reduction are not already available so we can not have fewer parameters than that defined in the car technical specifications. In fact, there are perhaps correlations between parameters which allow us to reduce their number before creating the parametric domain. However, no method is found until now for crash simulation.

— Thirdly, the deadline must be respected : one or two days is needed for an engineer to receive and analyze the results of a simulation ; so there is a limited number of points in the parametric domain which can be explored.

As a result, our objective is to find out a method to reduce the cost of an optimization study, with the same optimal solution if we use the high-fidelity model.

There are many ways to reduce the optimization cost. The first one is to use the meta-model in order to approximate the solution in all parametric domain. However, this method is purely statistical and not really efficient in crash simulation because of the very strong nonlinear characters of car structure. The second one is to reduce the cost of a unique simulation by changing the crash code. As a user of Pam-Crash software, Renault will not tend to reduce a single simulation, which is an intrusive reduction, because it is not belong to Renault skills but the software developers. The last one is the method that we aim for : we are interested in a **non-intrusive** and **a posteriori** method, i.e., no modification of the code is made and some points (crash simulations) are necessarily realized before the reduction, respectively. We will take advantage of the maximum physical outputs of these simulations (displacements, stress, strains, sectional efforts ... in function of time) in order to interpolate other points in parametric domain and/or to indicate which are next interesting points to simulate. The following reasons help us decide to chose a **non-intrusive** and **a posteriori** method

- The first reason is due to the unpredictable behaviors of crash simulation when we change parameters. It is impossible to calculate the crash results with analytic methods. That's why we should have some simulations done to see at least the crash behaviors of some points in parametric domain .
- The second reason is that we can not modify the crash solver but the number and type of outputs results (we call "**outputs**") are easily controlled so more outputs we get more information we will explore. We hope to find their potential relations that the solver can provide actually .
- The last reason is that the requirement of precision of the reduced model for optimization is not the same as that for a unique simulation. More precisely, we only need to known more exactly where is the optimal solution in the parametric domain without increasing the number of high-fidelity simulations. The precision for each simulation is not really necessary.

Recently, the Model Order Reduction (MOR) methods have been developed in many fields like control systems, dynamical systems, fluid mechanics, heat transfer and structural mechanics. The aim is to reduce the complexity and the time to resolve mathematical systems, usually systems of (partial) differential equations. In industry, a system can have a great number of dimensions or degrees of freedom (about millions) and consequently a big number of equations. In the dynamic context, the resolution is more and more difficult because these degrees of freedom depend on time. The reduction techniques involve the research of subspace that minimize the projection errors of the initial space (space containing the degrees of freedom of the initial problem). The dimension of subspace is much more smaller so the resolution is faster. The subspace is represented by its basis which can approximate the initial space. The non-intrusive and a posteriori reduction techniques are then both physically and mathematically interesting for us because they tend to conserve the physical properties and allow quantifying reduction errors, which are not the same things concerning the simplified models (LMS and MADYMO).

There are a variety of reduction methods of which the major can be found in [17]. Theses methods can be both applied for quasi-static and dynamic analyses, linear and nonlinear cases. We can cite some recent and popular methods such as LArge Time INcrement (LATIN) [4, 5] providing an iterative and non-incremental algorithm to resolve nonlinear problems, Proper Generalized Decomposition (PGD) [6, 7, 8] in which the solution is decomposed in a sum of products of variables (time, space and parameters), Reduced Basis (RB) [9, 10] which find a discretization for parametrized problems , Empirical Interpolation Method (EIM) [12] in which the base is built from points giving the maximum errors of projection, A Priori Hyper-Reduction (APHR)[16] which tends to reduce the number of elements in the high-fidelity model and the POD method [13, 14, 15] that we will detail after. These methods, firstly tested in academic use cases, have been introduced in the industrial contexts. However, the method for car crash simulation has not already been available. Our goal is not to invent another reduction method but try to test and adapt existing ones for our context. The POD method is possibly the best candidate for our **non-intrusive** and **a posteriori** reduction.

The POD method [21] is similar to Karhunen-Loève Decomposition [18, 19] and Principal Component Analysis (PCA) [20] in statistics and Singular Value Decomposition (SVD) [22] in linear algebra. From now on, we call SVD instead of POD because our data for reduction are directly in a matrix form (see section 4).

2 Short review of Singular Value Decomposition family

The SVD allows decomposing one matrix in a product of three ones. The extension to two, there matrices and more than three constitute a family of SVD while staying on the fundamental definition of SVD. In this section, we will review some significant methods. Their application in crash simulation will be detailed in the section 5.

2.1 Singular Value Decomposition

While the eigenvalues and eigenvectors problems deal with square matrix, the SVD try to do the same thing with a rectangle matrix.

Theorem 1. ***The singular value decomposition for real matrices [23, 21].***

If A is a $N_t \times N_x$ real matrix, then there exists real orthogonal matrices U ($N_t \times N_t$) and V ($N_x \times N_x$), i.e. $U.U_t = I_{N_t}$ and $V.V_t = I_{N_x}$

$$U = [u_1 \quad u_2 \ldots u_{N_t}]$$

and

$$V = [v_1 \quad v_2 \ldots v_{N_x}]$$

such that

$$U^t.A.V = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$$

where

$$p = min(N_t, N_x)$$

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r > \sigma_{r+1} = \cdots = 0 \qquad rank(A) = r \leq p$$

The σ_i are the singular values of A and the vectors u_i and v_i are respectively the i-th left and i-right singular vector.

Then A can be expressed as :

$$\boxed{A = U.\Sigma.V^t} \tag{1}$$

where

$$\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$$

If $r < N_x < N_t$:

$$\begin{aligned} A_{N_t \times N_x} &= U_{N_t \times N_t}.\Sigma_{N_t \times N_x} . V^t_{N_x \times N_x} \\ &= (U_{N_t \times r} \quad U_{N_t \times (N_t - r)}). \begin{pmatrix} \Sigma_{r \times r} & 0 \\ 0 & 0 \end{pmatrix} .(V^t_{N_x \times r} \quad V^t_{N_x \times (N_x - r)}) \\ &= U_{N_t \times r} . \Sigma_{r \times r} . V^t_{Nx \times r} \end{aligned}$$

In all case A can be decomposed by the sum of r matrices of the same dimensions as A :

$$\boxed{A_{N_t \times N_x} = \textstyle\sum_{i=1}^{r} u_i .\sigma_i .v_i^t = \sigma_1 \, u_1 \, v_1^t + \sigma_2 \, u_2 \, v_2^t + \cdots + \sigma_r \, u_r \, v_r^t} \tag{2}$$

Then

$$\text{Frobenius norm of A :} \qquad \| A \|_F^2 \quad = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 = \sum_{i=1}^{r} u_i \,.\sigma_i \,.v_i^t$$

and

$$\text{2-norm of A :} \qquad \| A \|_2 \quad = max_{\|x\|=1} \ \| Ax \| = \sigma_1 \qquad \text{where } x \text{ is an real } n \times 1 \text{ vector}$$

We can prove easily that U and V are respectively eigenvectors of hermitian matrices $A.A^t$ and $A^t.A$, respectively. $\sigma_{i,(i=1,\ldots,r)}$ are eigenvalues of both two matrices $A.A^t$ and $A^t.A$ because

$$A.A^t = U \,.\, \Sigma^2 \,.\, U^t$$

$$A^t.A = V \,.\, \Sigma^2 \,.\, V^t$$

2.2 Generalized Singular Value Decomposition

Theorem 2. ***The generalized singular value decomposition for REAL MATRIX pairs***
If A is a $N_t \times N_{x_1}$ matrix with $Nx_1 \geq N_t$ and B is a $N_t \times N_{x_2}$ matrix, then there exists orthogonal matrices $U(N_{x_1} \times N_{x_1})$ and $V(N_{x_2} \times N_{x_2})$ and an invertible $X(N_t \times N_t)$ such that :

$$X \,.\, A \,.\, U = D_A = diag(\alpha_i) \quad \alpha_i \geq 0 \quad i = 1, \ldots, N_t$$

and

$$X \,.\, B \,.\, V = D_B = diag(\beta_i) \quad \beta_i \geq 0 \qquad i = 1, \ldots, q = min(N_t, N_{x_2})$$

where

$$\beta_1 \geq \beta_2 \geq \cdots \geq \beta_r \geq \beta_{r+1} = \cdots = \beta_q = 0 \qquad r = rank(B)$$

The elements of the set $\sigma(A, B)$ are generalized singular values of A and B :

$$\sigma(A, B) = \{\alpha_1/\beta_1, \ldots, \alpha_r/\beta_r\}$$

The generalized singular values corresponding to $\beta_i = 0$ are infinite. While the SVD is to find $\sigma \geq 0$ such as :

$$det(A \,.\, A^t - \sigma I_{N_t}) = 0$$

the GSVD is to find $\sigma(A, B)$ such as :

$$det(A \,.\, A^t - \sigma \,.B \,.\, B^t) = 0$$

Then A and B can be expressed as :

$$\boxed{\begin{array}{ll} A & = X^{-1} \,.\, D_A \,.\, U^t \\ B & = X^{-1} \,.\, D_B \,.\, V^t \end{array}} \tag{3}$$

We remark that the the decomposition above allows A and B, which contain the same number of rows, to have the same left matrix X . If we have 2 matrix A and B having the same number of columns, with $A \in R^{N_{t_1} \times N_x}$, $B \in R^{N_{t_2} \times N_x}$ and $N_{t_1} \geq N_x$, their properties stay the same and their decomposition is call "**BSVD**" by Van Loan[24] :

$$U^t \, A \, X = D_A = diag(\alpha_i)$$

$$V^t \, B \, X = D_B = diag(\beta_i)$$

where $U(N_{t_1} \times N_{t_1})$ and $V(N_{t_2} \times N_{t_2})$ are orthogonal matrices and $X(N_x \times N_x)$ non-singular matrix. $q = min\{N_{t_2}, N_x\}$. Bart de Moor [25, 26] extended to matrices with complex values with Quotient Singular Value Decomposition (**QSVD**) – generalized form of **BSVD** and Product Singular Value Decomposition (**PSVD)** that we will not detail now for the sake of time.

2.3 Higher-Order Generalized Singular Value Decomposition

In many case, we need to work with more than two matrices simultaneously, so the GSVD can not be applied. Van Loan et *al.* [27] developed a new method called "Higher-Order Generalized Singular Value Decomposition" (HO GSVD) which allows the factorization for $N \geq 2$ matrices.

Suppose we have a set of N real matrices $D_i \in R^{m_i \times n}$ and $rank(D_i) = n$, so the HO GSVD of N matrices is :

$$\begin{aligned} D_1 &= U_1 \, . \, \Sigma_1 \, . \, V^t \\ D_2 &= U_2 \, . \, \Sigma_2 \, . \, V^t \\ \vdots \quad & \quad \vdots \\ D_N &= U_N \, . \, \Sigma_N \, . \, V^t \end{aligned} \tag{4}$$

where

$U_i \in R^{m_i \times n}$: *normalized left basis vectors*
$\Sigma_i = diag(\sigma_{i,k}) \in R^{n \times n}$: *diagonal with $\sigma_{i,k} > 0$*
$V \in R^{n \times n}$: *normalized right basis*

We remark that U and V are **not orthogonal**, i.e. $U_i.U_i^t \neq I_{m_i}, i = 1, \ldots, N$ and $V.V^t \neq I_n$ but their columns are still independent and normalized. The interesting point here is that V is the same for all matrices D_i. The following steps show how to calculate U_i, Σ_i and V [27]. We define the S matrix with $S_{ij} = \frac{1}{2}(A_i A_j^{-1} + A_j A_i^{-1})$, $i \neq j$ where $A_i = D_i^t D_i$.

$$S = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j>i}^{N} (A_i A_j^{-1} + A_i A_j^{-1}) \tag{5}$$

$$S = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j>i}^{N} S_{ij}$$

S is proven has n independent eigenvectors so it can be decomposed :

$$SV = V\Lambda$$

Where $V = (v_1, \ldots, v_n)$, $\parallel v_k \parallel = 1$ and $\Lambda = diag(\lambda_k), k = 1, \ldots, n, \lambda_k \geq 1$. V and Λ are eigenvectors and eigenvalues respectively. Then we compute matrices B_i by solving N linear systems :

$$\begin{aligned} VB^t &= D_i^t \\ B_i &= (b_{i,1}, \ldots, b_{i,n}), \qquad i = 1, \ldots N \end{aligned}$$

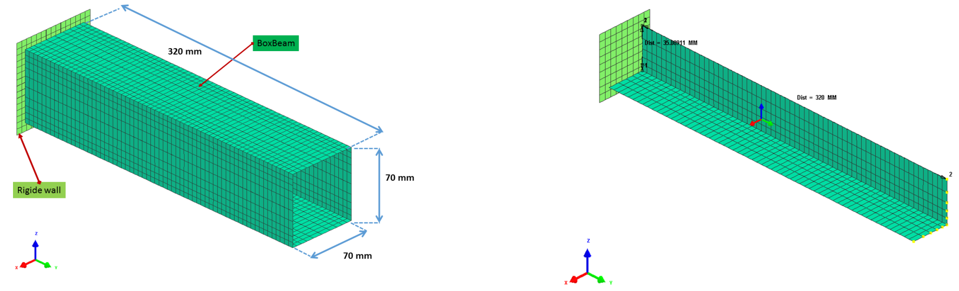
Finally,

$$\boxed{\begin{array}{rcl} \sigma_{i,k} &=& \| \; b_{i,k} \; \| \\ \Sigma_i &=& diag(\sigma_{i,k}) \\ B_i &=& U_i \Sigma_i \end{array}}$$

3 Description of crash use cases

A car is a complex object composing of millions of parts. Before testing a reduction method for a whole car, it is desirable to know how it works for a unit part and then a set of part. As a results, we made two use cases with the increasing complexity – BoxBeam (figure 1 (a)) and TBeam (figure 2).

1. **BoxBeam** : the crushing of a thin-walled beam against a rigid wall at the left section and hit by a rigid body, in the perpendicular direction to the wall, at the right section (the rigid body is not presented in the figure 1). The BoxBeam has square section. Because of the symmetry, we can consider from now on its quarter, see figure 1 (b). This represents the crushing of a side-member during the crash.
2. **TBeam** : This presents some significant parts in the motor compartment. The purple part (a rigid body) hits perpendicularly the dark green part with an initial velocity. In crash, the motor doesn't deform so it is modeled as a rigid body (cylinder int the figure 2).



(a)
Complete BoxBeam

(b)
A quarter

FIGURE 1: BoxBeam

4 Matrice representation of crash output

After a crash simulation, there are two kinds of outputs : one allows to visualize the temporal evolution of the deformation of the whole structure (like a film of the deformed structure) and other allows to draw plots of interested physical quantities in function of time (ex. displacements of a node, efforts in some sections, . . .). We tend to use all available outputs for the reduction.

In order to use the properties of SVD family, we expressed all output in the matrix form. Take a crash simulation, there are 3 kinds of outputs : local, partial and global if we consider a car part.

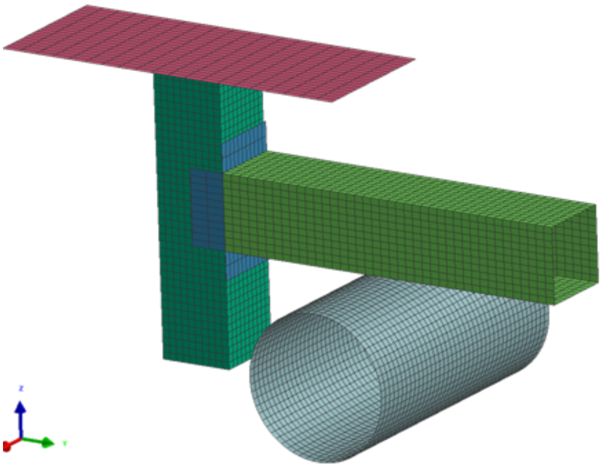


FIGURE 2: TBeam

- **Local** : They are elementary and nodal quantities, such as the displacements of a node or the moment of a element.
- **Parital** : Ex. cross section efforts – the force transmitted from a part to another
- **Global** : Ex. Deformation energy of a part

For each part, we are going to write the output data in a matrix of which each column is the temporal evolution of a physical quantity (local, partial or global) in each direction if this physical quantity is a vector (ex. displacements have 3 components) or tensor (ex. tensor of stress has 6 components). The outputs are taken each 5 time steps (BoxBeam) and 10 time steps (TBeam) of the crash solver Pam-Crash. They are "fine" and "reliable" for the reduction.

The BoxBeam has about $m = 1500$ output time steps and $n = 700$ nodes so the matrix of the X-coordinate of all nodes is A :

$$A = \begin{vmatrix} a_{t_1,N_1} & a_{t_1,N_2} & \dots a_{t_1,N_n} \\ a_{t_2,N_1} & a_{t_2,N_2} & \dots a_{t_1,N_n} \\ \vdots & \vdots & \vdots \\ a_{t_m,N_1} & a_{t_2,N_2} & \dots a_{t_1,N_n} \end{vmatrix}$$

where $t_1, \dots t_m$: increasing time steps and $N_1, \dots, N_n$: identity of nodes from 1 to n. So A_{t_i,N_j} is the X-coordinate at instant t_i of node N_j.

5 SVD family for reduction

In this section, we test the classical SVD to reduce a simulation and then try to interpolate between two simulations when we modify parameters. We would like to rebuild the matrix of all coordinates (concatenation horizontally of X-coordinate, Y-coordinate and Z-coordinate matrices).

5.1 Singular Value Decomposition application

5.1.1 BoxBeam

Reduction of a unit simulation We call the matrix of all coordinates A_{bb}. Its decomposition following the (1) :

$$A_{bb} = U_{bb}.\Sigma_{bb}.V_{bb}^t$$

The number of column of A_{bb} is 3 times the number of nodes, i.e. 2100. So

$$A_{bb} = A_{15000 \times 2100}$$

The graph 3 shows that the logarithmic values of the first singular values $(\sigma_i, i = 1, \ldots\ldots, 20)$ drop rapidly. We don't display others because of their very small values. The graph allows us to expect a possible reduction with crash simulation.

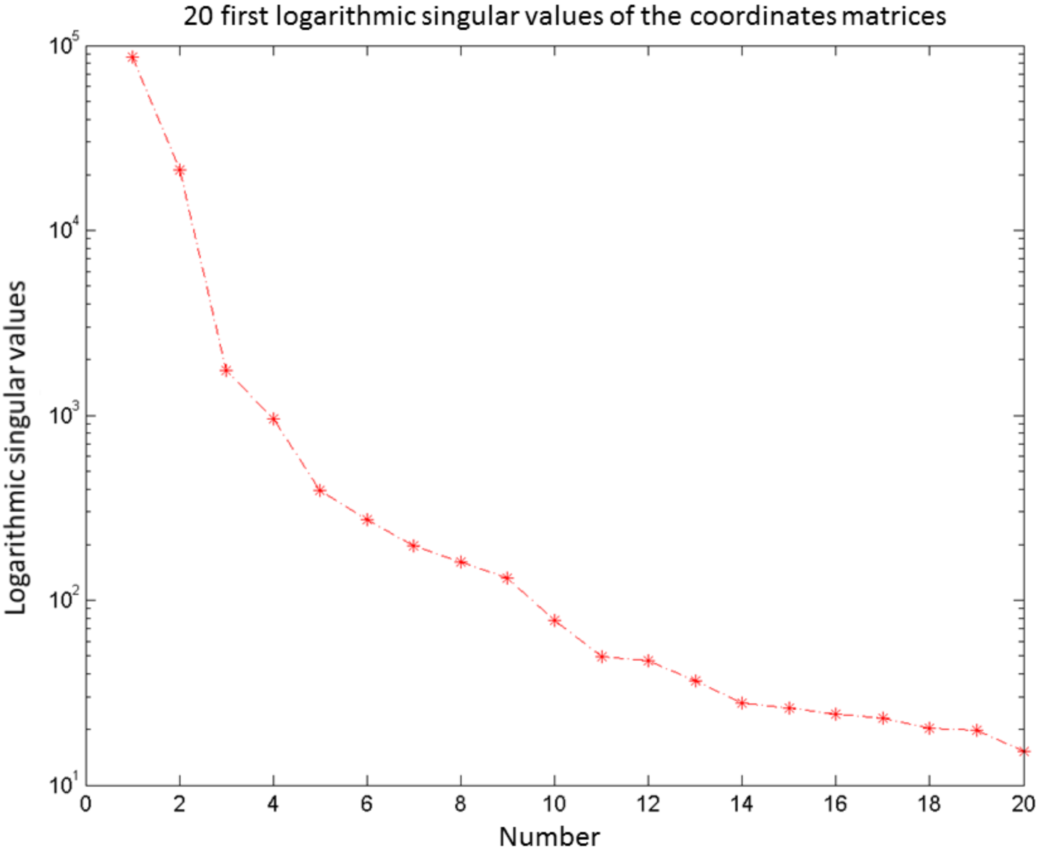


FIGURE 3: Evolution of 20 first logarithmic singular values

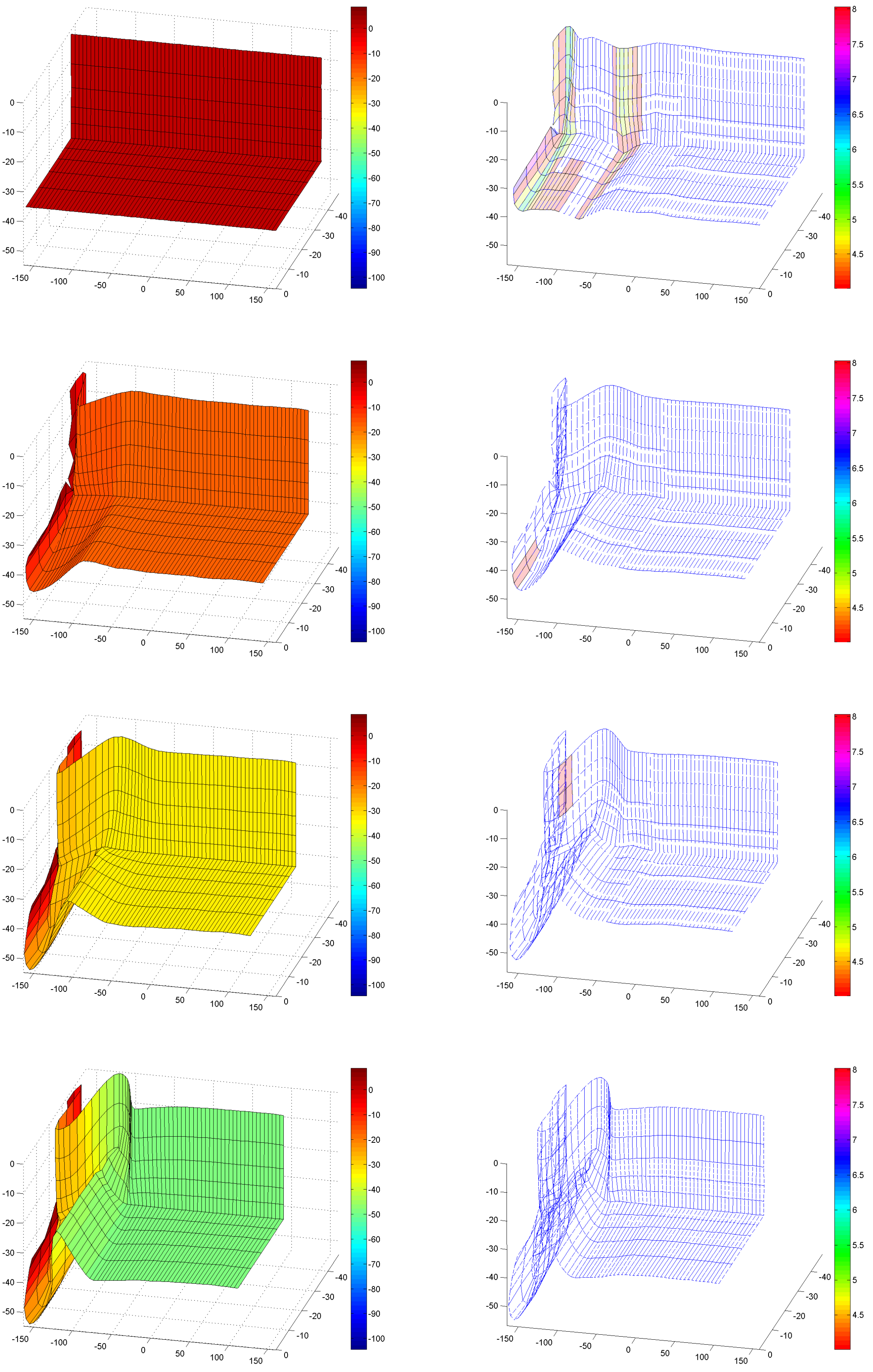
Then we try taking only 3 singular values to approximate A_{bb}, following the formular (2).

$$A_{bb} = \sigma_{(bb),1}\, u_{(bb),1}\, v^t_{(bb),1} + \sigma_{(bb),2}\, u_{(bb),2}\, v^t_{(bb),2} + \sigma_{(bb),3}\, u_{(bb),3}\, v^t_{(bb),3} \tag{6}$$

The rebuilt deformation of BoxBeam is showed in the figure 4. The left figures (with colors in the whole surface) are different deformation states (from the beginning to the end of the crash film). The right figures are the same thing but with an approximation of coordinates with the (6) ; the location of a colored element is where euclidean error is greater than 50 $\%$ of the maximum euclidean error (50 $\%$ quantile) (euclidean error is the distance point by point between approximated mesh and initial mesh). We can conclude :

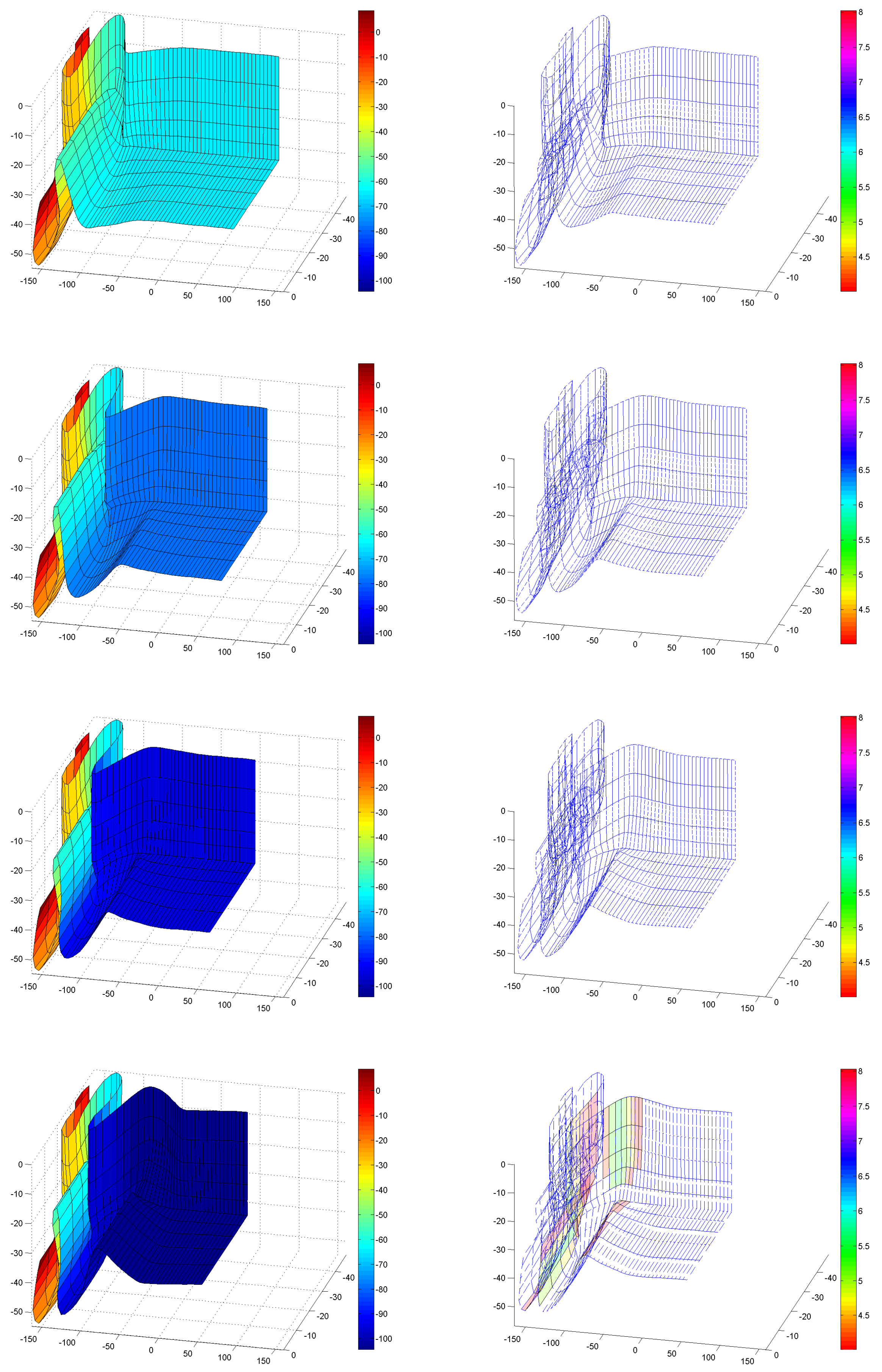
— Despite strongly nonlinear behaviors of the crash simulation, the SVD can rebuild quite well a crash deformation by only 3 axes.

- The errors are located in the folds so they will not spread from a part to another. This is interesting in a system simulation because the total error will not amplify.
- The greatest errors are at the beginning of the simulation which is purely numerical.
- In term of data size, the SVD allows us to keep only 3 columns of U and 3 columns of V so fewer than 300 times the number of columns of the initial matrix A_{bb} (2100).



(a) Without SVD (initial)

(b) With SVD



(a) Without SVD (initial)

(b) With SVD

FIGURE 4: Initial deformation v.s. deformation with 3 axes SVD for coordinates

Interpolation We try interpolating the simulation of thickness 1.2 mm between two different thickness 0.8 mm and 1.6 mm, other parameters stay unchanged. These are real part thickness in the car industry. The figure 6 compare the real simulation of 1.2 mm with the linear interpolation of the reduced model by 3 axes SVD of 0.8 mm and 1.6 mm. We remark :

- The interpolation represents quite well the real simulation despite large errors (50 % quantile) in the folds.
- If we increase the number of kept axes SVD, the interpolation errors reduce but stagnate at a certain axis number, see the figure 5.1.1. This means that the linear interpolation is not really good due to the fact than linear interpolation of two SVD is not a SVD. It is desirable to test with other methods.

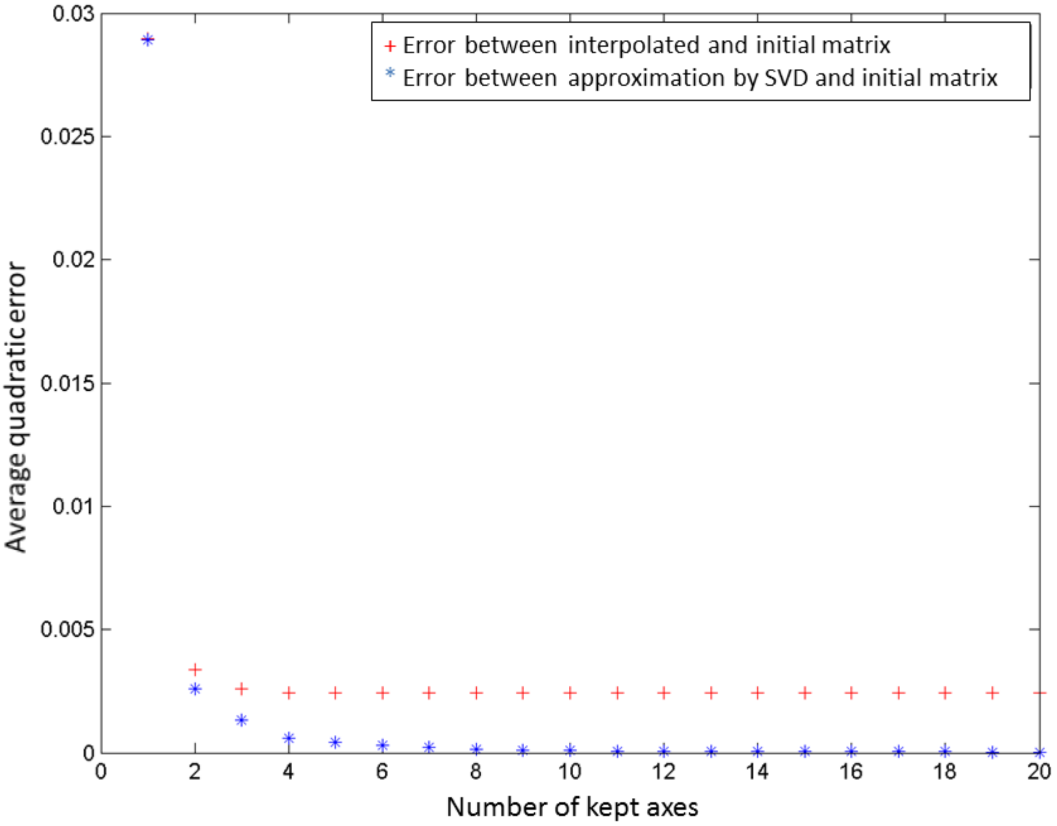
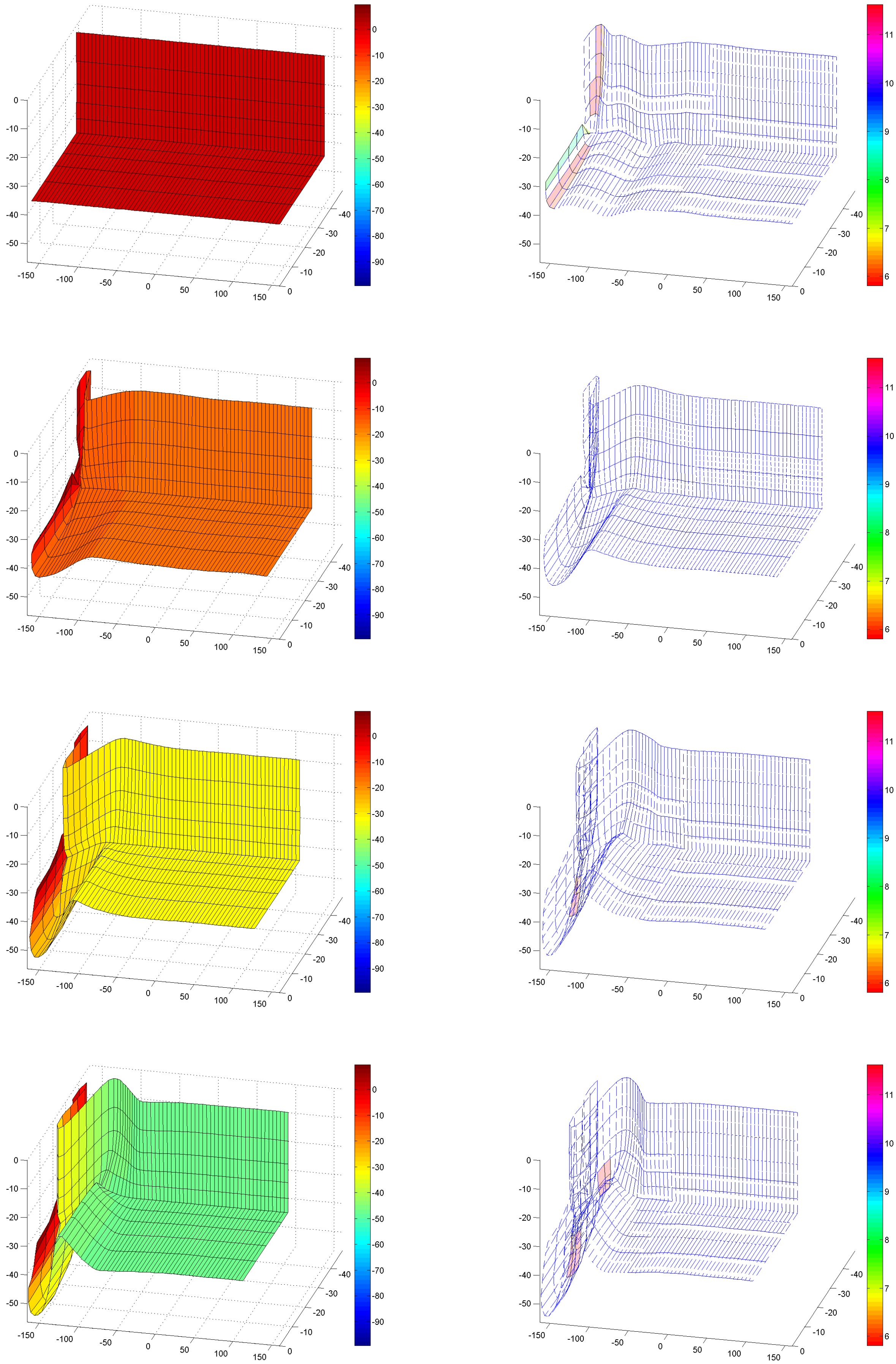
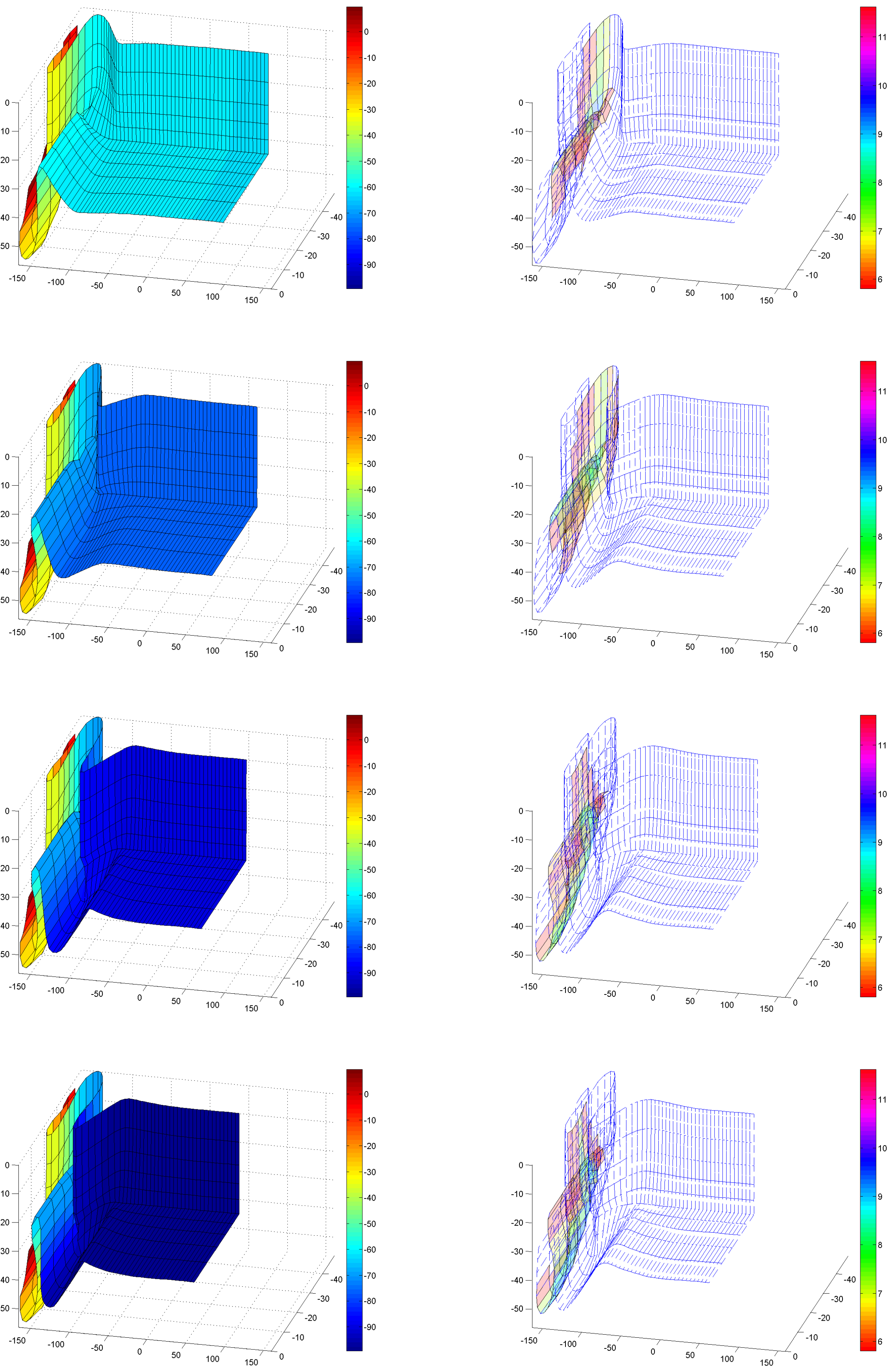


FIGURE 5: Evolution of average quadratic error in function of kept axes



(a) 1.2 mm without SVD (initial)

(b) 1.2 mm with 3 axes SVD then linear interpolation



(a) 1.2 mm without SVD (initial)

(b) 1.2 mm with 3 axes SVD then linear interpolation

FIGURE 6: Initial deformation of 1.2 mm v.s. interpolation of 0.8 mm and 1.6 mm with SVD

5.1.2 TBeam

Reduce a unit simulation We do the same things for the TBeam with the matrix A_{tb}

$$A_{tb} = U_{tb}.\Sigma_{tb}.V_{tb}^t$$

Different deformation states can be seen at the figure 7

The colors of the right figures are the errors between initial TBeam and that rebuilt with 3 axes SVD. The approximation is not really well in light colored area (the right end of the horizontal part) but it can be cut back because these areas have rigid body movements.

For the sake of time, we do not present here the interpolation for TBeam.

5.2 Higher-Order Generalized Singular Value Decomposition application

In the subsection 2.3, the HO GSVD gives the same V, then the same basis for all matrix. If we consider $m_i = N_{t_i}$ – number of time steps of the simulation i and $n = N_x$ – number of nodes of the FE model, we can have the same spacial basis for all simulations. The interpolation is then very easy. That is exactly what we expected in a parametric domain : interpolation all unknown simulations from done simulations.

Unfortunately, there are many restrictions that we can not apply HO GSVD for a coordinate matrix :

— X (Y , Z) - coordinate matrix has not always full columns rank

— The calculation of S is difficult, see (5) because it needs the inversion of matrices.

We conclude that we should not use HO GSVD as a reduction method with local outputs. We expected that may work with partial and global outputs, see definition in 4. In this article,we will present the HO GSVD with a matrix of deformation energies, see 2. We call

— E^1 : Kinetic energy of impactor (purple part)

— E^2 : Deformation energy of dark green part

— E^3 : Deformation energy of green part

— E^4 : Deformation energy of spot-weld

We realized 11 simulation while modifying the velocity of impactor and the thickness of two thin-walled parts. Each matrix D_i can be expressed :

$$D_i = \begin{vmatrix} E^1_{t_1,i} & E^2_{t_1,i} & E^3_{t_1,i} & E^4_{t_1,i} \\ \vdots & \vdots & \vdots & \vdots \\ E^1_{t_m,i} & E^2_{t_m,i} & E^3_{t_m,i} & E^4_{t_m,i} \end{vmatrix}$$

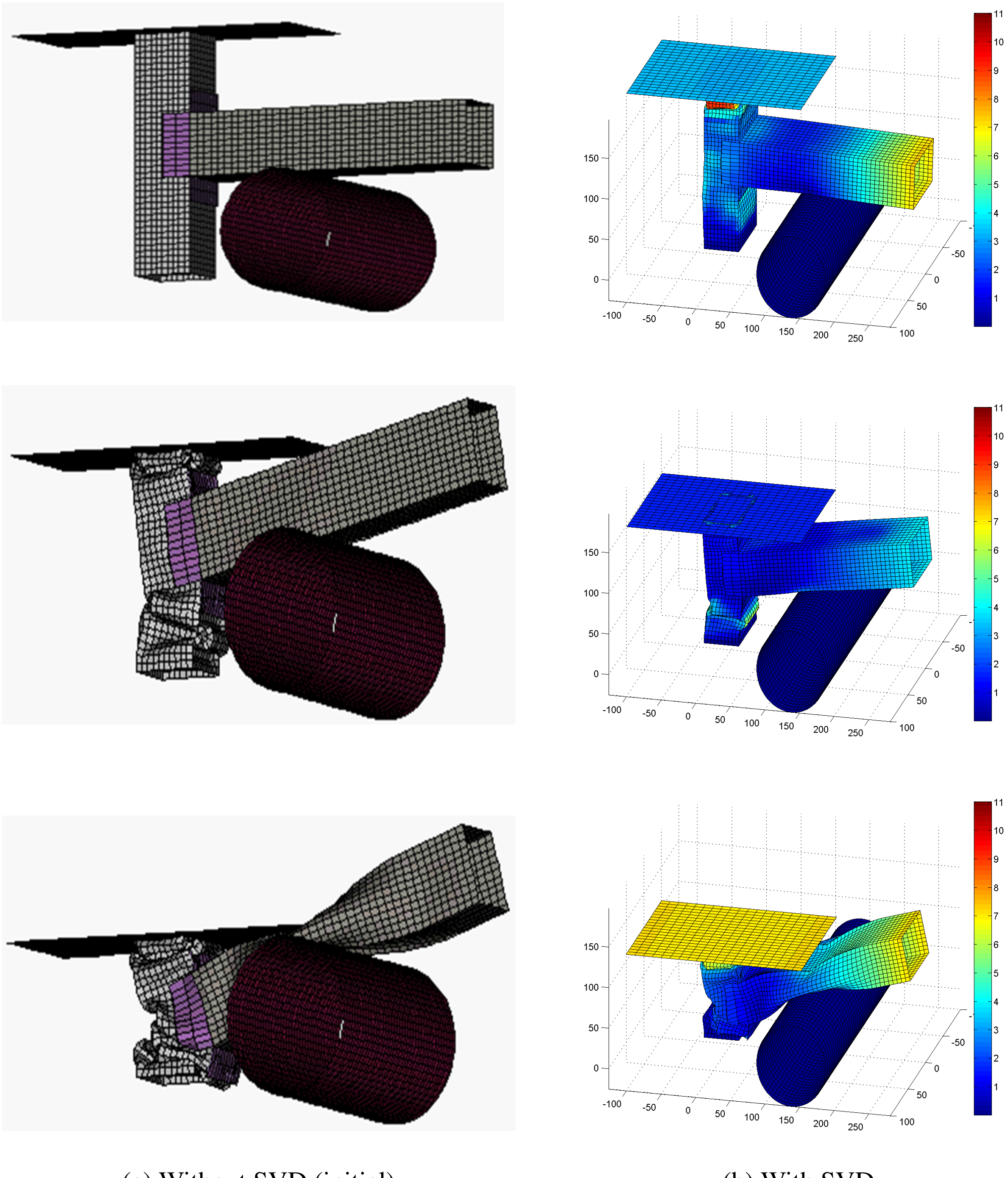
The singular values of 11 matrices are showed in the figure 8.

It is proven in [27] that if a eigenvalue λ_k of S in formula (5) is equal to 1, $\sigma_{i,k} = \sigma_{j,k}$ for all i and j. In analogy with the GSVD, the ”common HO GSVD subspace” of the $N > 2$ matrices to be the subspace spanned by the right basis vector v_k corresponding to the eigenvalues of S that satisfy $\lambda_k = 1$. As the result, in the figure 8 the simulation 119 is very ”different” from others and the couple (125,126) is quite ”similar” which is better for interpolation.

6 CONCLUSIONS

— The classical SVD allows to reduce a unit simulation. To interpolate, we need to find proper methods than linear interpolation.

— With partial and global outputs, the HO GSVD may show the difference and the similarity among simulations. It is very useful before the interpolation.



(a) Without SVD (initial)

(b) With SVD

FIGURE 7: Initial deformation v.s. deformation with 3 axes SVD for coordinates

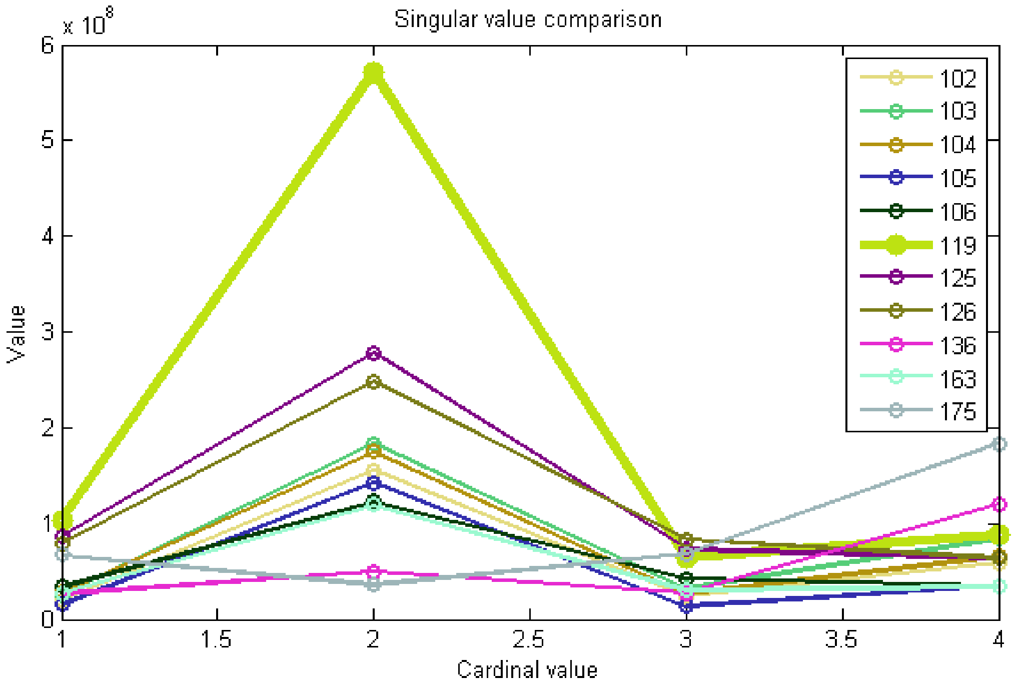


FIGURE 8: Singular values of energy matrices

— Concerning the next work, the PSVD and QSVD may be tested as methods to estimate a physical quantity from another quantity, i.e. if we know how to interpolate the coordinate of an unknown simulation, we can calculate the velocity of this interpolated simulation.

RÉFÉRENCES

[1] Kamal M. M., *Analysis and Simulation of Vehicle to Barrier Impact*. SAE Paper No. 700414, 1970.

[2] Hahm, S-J, Won, Y-H, and Kim, D-S, *Frontal Crash Feasibility Study Using MADYMO 3D frame Model*. SAE International Congress and Exposition, Detroit, Paper No. 1999-01-0072, 1999.

[3] Paul Du Bois, Clifford C. Chou, Bahig B. Fileta, Tawfik B. Khalil, Albert I. King, Hikmat F. Mahmood, Harold J. Mertz, Jac Wismans, *Vehicle crashworthiness and occupant protection*. American Iron and Steel Institute, 2000 Town Center, Southfield, Michigan 48075.

[4] Ladevèze P., *Mécanique non linéaire des structures , nouvelle approche et méthodes de calcul non incrémentales*. HERMES, ouvrage 280 p. 15.5x23.5 cm Broché.

[5] Christophe Heyberger, Pierre-Alain Boucard, David Néron *Multiparametric analysis within the proper generalized decomposition framework*. Comput Mech (2012) 49 :277–289. DOI 10.1007/s00466-011-0646-x.

[6] Amine Ammar, Francisco Chinesta, Elías Cueto and Manuel Doblaré, *Proper generalized decomposition of time-multiscale models*. Int. J. Numer. Meth. Engng 2012 ; 90 :569–596.

[7] Anthony Nouy, *A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations*. Computer Methods in Applied Mechanics and Engineering (2010), DOI : 10.1016/j.cma.2010.01.009.

[8] Antoine DUMON *Re´duction dimensionnelle de type PGD pour la résolution des écoulements incompressibles*. PhD 2011.

[9] Alfio Quarteroni, Gianluigi Rozza, Andrea Manzoni, *Certified reduced basis approximation for parametrized partial differential equations and applications*. Journal of Mathematics in Industry (2011) 1 :3. DOI 10.1186/2190-5983-1-3.

[10] N.C. Nguyen, G. Rozza, D.B.P. Huynh, and A.T. Patera, *Reduced basis approximation and a posteriori error estimation for parametrized parabolic PDEs ; Application to real-time Bayesian parameter estimation*. Massachusetts Institute of Technology, Department of Mechanical Engineering, Room 3-264, 77 Massachusetts Avenue, Cambridge MA, 02139-4307, USA.

[11] Gianluigi Rozza, *Reduced Basis Approximation and Error Bounds for Potential Flows in Parametrized Geometries*. Commun. Comput. Phys. doi : 10.4208/cicp.100310.260710a

[12] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, Anthony T. Patera, *An "empirical interpolation" method : application to efficient reduced-basis discretization of partial differential equations*. Numerical Analysis, C. R. Acad. Sci. Paris, Ser. I 339 (2004) 667–672.

[13] A. Chatterjee, *An introduction to the proper orthogonal decomposition*, Current Science, vol. 78 (7), pp. 808-817, April 2000.

[14] Seong Woo Hong, *A method to assess dynamic responses in a crash simulation*. A Dissertation submitted to The Faculty of The School of Engineering and Applied Science of The George Washington University in partial satisfaction of the requirements for the degree of Doctor of Philosophy

[15] MURUHAN RATHINAM and LINDA R. PETZOLD. SIAM J. NUMER. ANAL.Vol. 41, No. 5, pp. 1893–1925 *A new look at proper orthogonal decomposition*

[16] D. Ryckelynck, *A priori hypereduction method : an adaptive approach*. Int. J. of Computational Physics, 202 :346–366, 2005.

[17] A.C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM Book Series : Advances in Design and Control

[18] M. Loève, *Probability Theory*. Van Nostrand, 1955.

[19] K. Karhunen, *Zur spektral theorie stochasticher prozesse*. Ann. Acad. Sci. Fennicae, Ser. A1, 34, 1946.

[20] I.T. Joliffe, *Principal Component Analysis*. Springer-Verlag, 1986.

[21] L. Cordier et M. Bergmann *Réduction de dynamique par Décomposition Orthogonale aux Valeurs Propres*. Ecole de printemps OCET Optimisation et Contrôle des Ecoulements et des Transferts, 12 – 17 Mars 2006, Aussois

[22] G.H. Golub et C.F. Van Loan, *Matrix computations*. The Johns Hopkins University Press, Baltimore, 1990.

[23] Joos Vandewalle and Bart de Moor, *A variety of applications of singular value decomposition in identification of signal processing*. SVD and Signal Processing : Algorithms, Applications and Architectures. E.F.Depretterc, North Holland, Elseviers Science Publishers B.V., 1988, pp. 43-91

[24] Charles F. VAN LOAN, *Generalizing the singular value decomposition*. SIAM J. NUMBER. ANAL., vol. 13 ; No 1.March 1976.

[25] Bart L.R. De Moor, Gene H. Golub, *Generalized Singular Value Decompositions : A Proposal for a Standardized Nomenclature*. Numerical Analysis Project, Computer Science Department, Stanford University Stanford, California 94305.

[26] B. de Moor, *Generalizations of the OSVD Structure, Properties and Applications*. In SVD and Signal Processing, no. 181232, pp. 83-98, 1991.

[27] Sri Priya Ponnapalli, Michael A. Saunders, Charles F. Van Loan, Orly Alter, *A Higher-Order Generalized Singular Value Decomposition for Comparison of Global mRNA Expression from Multiple Organisms*. PLoS ONE, vol. 6, no. 12, December 2011.