

Model Reduction Techniques for On-board and Parametric Crash and Safety Simulations

Kambiz Kayvantash, CADLM SAS (www.cadlm.com)

kambiz.kayvantash@cadlm.com

1 Introduction

Model reduction techniques (or Reduced Order Modelling - ROM) are algebraic solutions for reducing the volume of a data set while preserving the most important parts of the information contained within the data, necessary for retrieving all or the most essential part of the data when needed. This is commonly done via decomposition or clustering or other efficient data compression techniques. Such techniques allow for creating on-board and real-time applications based on voluminous experimental or simulation results (ex. Finite element).

In this paper we shall present the major idea behind the reduction (or fusion) methods as well as providing three potential applications for crash and safety simulations. The results are obtained by ODYSSEE (Lunar) software [2] and compared with LS-DYNA FEM results.

***KEYWORDS:** Model Reduction, ROM, POD, CLUSTERING, SVD, On-board computing, Real-time modeling, Crash, Safety, Parametric design

2 Reduced Order Modeling

In 2015 Kayvantash [1] reported on an innovative solution and presented results based on model reduction techniques in order to exploit fully the results of time dependent DOE's such as in the case of Crash or ALE. An explanation of the algorithm (based on POD standing for Proper Orthogonal Decomposition) was given and a demonstration of the method was provided for a typical safety simulation model as well as an ALE application (ballistic impact including detonation and fluid/structure interaction). In 2017 Yasuki [5] presented a full sled test demonstration using tools developed by Kayvantash [2] and provided detailed comparative studies of the FEM and ROM solutions. Further, in 2018 Yasuki [6] presented a population based comparison of 100 sled tests using an FE dummy + sled simulation and compared it to ROM reconstructions of the same scenario. Additionally, after validation of the comparative studies, a Pareto Front profile of Nij for 10000 hypothetic individuals was established. In 2018 Ovazza and Kayvantash [4] developed new methods of decomposition based on clustering techniques and machine learning which improved yet further the computation speed and reduced the data storage required for implementing ROM on on-board devices. These works have launched an interest in applications of ROM for crash and safety simulations, parametric studies and optimizations. In this work we shall explain the principles and discover further the potential of such reduction methods.

Some reduction techniques are based on reducing the order of the operators representing the set of algebraic operations resulting from discretization of differential equations. These are commonly referred to as PGD (Proper Generalized Decomposition) and are only applicable to reducing the order of partial differential equations (PDF). These are considered as "intrusive" since they require extensive modifications or even rewriting of solver code. However, recent research has shown that using machine learning techniques other physical phenomena - for which a PDF (partial differential equation) is not known, or does not exist, or not fully or efficiently handled from a computation effort point of view - may also be explored and predicted using ROM methods. In particular, "non-intrusive" or a posteriori techniques of results post-processing are of interest. These methods (such as POD) require simply the establishment a new vector base (orthogonal or reference vectors) as opposition to the original parameter/temporal related and a projection of the known model or experimental results on that new base. This technique is similar to a PCA (Principal Component Analysis) projections.

Independent of the method selected, a ROM technique implies three steps: Decomposition, Reduction and Reconstruction. In the next paragraphs we shall describe shortly these steps and then proceed with three crash and safety related applications.

In what follows we shall use the following conventions:

X = Table of design parameters (or variables) defining experiments (we also call this a DOE)
Y = Results of the experiments associated with **X**
XN = A new table of parameters for which we seek results without actually performing experimenting or computing
YN = Results of the new experiments associated with **XN** (to be computed by ROM)
G, H, S = Decomposition matrices of which **G_r, H_r, S_r** are sub-matrices.
y = Spatial or latent metrics of system (sometime called coefficients)
t = time
m, n, p, r = indices (dimensions of matrix)

Note – The **N** in (**XN** and **YN**) stands for **NEW** since we do not include **XN**, which is known, in the decomposition process and we don't know the **YN** (since we want to predict it).

2.1 Decomposition

Assuming we have a set of data representing a time dependent phenomena such as **Y(y, t)**, where **y** represent a “spatial” but latent variables associated to the response.

We define the decomposition:

Y(y, t) = G(y). s . H(t) where **G** is a “spatial” or “latent” operator (a matrix) , **s** is a transfer matrix (diagonal or semi-diagonal) and **H** is a “temporal” or “modal” operator (a matrix).

Let's assume also that we know of another set of data **X** (representing a sampling table of some properties or criteria which have resulted in the variations of the responses. In short, **X** represents a Design of Experiments and **Y** the corresponding outcome at the **X** sampling points. We are interested in establishing a function (the ROM operator) which relates **Y** to **X** as in the case of a “supervised” learning algorithm.

Presented in matrix form, if the original data are of dimensions **X_{m,n}** and **Y_{m,p}** then the decomposed matrices have the dimensions **G_{m,m}, S_{m,p}** and **H_{p,p}**.

If the above decomposition exists, and we may explore this existence by some matrix algebra methods such as *SVD* (Singular Value Decomposition) or *Machine Learning* (such as clustering, etc.) then we can claim to have reduced (or projected) the original data set (on) to a new set of basis with special and very useful properties such as orthogonality.

In particular by applying the decomposition and the projection we have separated the variables **y** and **t** (similar to modal decomposition techniques in transient dynamic problems). We can claim to have obtained a decoupled version of the original set of data. Secondly, we can now perform separate operations on any of the two major components (matrices **G** or **H**) in order to further exploit the data, in association with many available interpolation techniques.

2.2 Reduction

In general matrix **s** contains either a set of singular values (if *SVD* based decomposition is performed) or some cluster coupling coefficients (if clustering is performed) allowing to relate the spatial and temporal components of **Y**. It is possible to consider only subsets of **G** and **H** based on some tolerance criteria (such as the ratio of diagonal terms of matrix **s**) captivating only the most important part of the two matrices. We may call these subsets or reduced matrices **G_r** and **H_r**.

In this case only the “reduced” matrices need be considered with important consequences for CPU and storage issues which are essential for on-board computing solutions. In matrix form, if the original data **X** and **Y** are of dimensions **X_{m,n}** and **Y_{m,p}** then the decomposed matrices may have the dimensions

$\mathbf{G}_{m,r}$, $\mathbf{S}_{r,r}$ and $\mathbf{H}_{r,exp}$ (where r represents the number of retained modes or clusters). We can observe that if a new set of parameters is experimented such as $\mathbf{XN}_{q,n}$, the reduced matrices still provide a result matrix $\mathbf{YN}_{m,p}$ (same order as $\mathbf{Y}_{m,p}$).

At any instance we can either reconstruct the original set of data either completely (using decomposed operators) without loss or partially (using reduced operators) with some loss of information. A partial reconstruction would mean that we only retain selected parts of \mathbf{G} and \mathbf{H} , a technique which can be assimilated to filtering. This allows a reduction of the volume of the “on-board” data as well as a filtering of noise or other high frequency sources of uncertainty in the results.

2.3 Reconstruction

In case of predictions which require the results \mathbf{YN} for a new parameter set \mathbf{XN} , we observe that we can replace \mathbf{G} or \mathbf{H} , by their modified updates \mathbf{G}' or \mathbf{H}' , taking into account of the change in the parameters, and obtain a new set of response corresponding to slightly modified new positions \mathbf{XN} compared to the original \mathbf{X} . In simple terms we can make predictions of effect of \mathbf{X} moving to \mathbf{XN} on \mathbf{Y} (moving to \mathbf{YN}) via considering its effect on \mathbf{G} , moving to \mathbf{G}' or \mathbf{H} moving to \mathbf{H}' only. If we need to compute \mathbf{YN} , we need to compute the effect of \mathbf{XN} on \mathbf{G} or \mathbf{H} . It turns out that the updates \mathbf{G}' or \mathbf{H}' may be simply obtained by an adequate interpolation technique such as radial basis functions, kriging, etc. A final back multiplication provides the results \mathbf{YN} .

Lastly, it is important to point out that it is possible to mix the experimental and numerical results and construct a unified data base ($\mathbf{Y}_{\text{experiments+simulation}}$) in which case we could call the data base a fusion of real and virtual information. This alone opens new horizons in modelling which until now was always suffering from “correlation” and “validation” issues. We can claim that we have a new tool which benefits from the best of both experimental and numerical technologies.

3 Applications

Three applications will be presented hereafter showing different potential of the proposed method and solution software. The starting point is, as in many other parametric design projects, the construction of the DOE samples and the time dependent response of the system for each case present in the DOE table (Figure 1.). Note that the response may be measured experimentally or computed via discretization techniques such as Finite Elements, etc.

For all three applications a Desktop version of the software ODYSSEE (Lunar) has been used running on a Laptop (DELL M4800, INTEL i7 4910 MQ, 2.9GHz, 2.9GHz, 16 GB RAM) was used.

3.1 Simply supported plate without or with rupture

A simply supported plate with variable thickness and moving support position is considered in order to compute the deflection at the loaded side of the plate (Figure 2.). A DOE of 15 cases was first assumed and analysed via FEM. The results were obtained for four new points by ROM methodology [Figure 1.] A finite element based solution (1000 elements) is compared with a real-time (instantaneous, ~1sec) solution. CADLM's ODYSSEE (Lunar) software [2] was used for the ROM modelling.

3.2 Simplified Bonnet head impact

A simply supported plate with a spherical impactor is considered as a simple model of a head impact on a bonnet as in the case of pedestrian impact. The plate is made of composite material allowing for failure corresponding to rupture of fibres (Figure 2.). It is intended to evaluate the capabilities of the ROM solution in case where rupture (or bifurcation occurs). This clearly shows that the method is not limited to simple, linear or smoothly non-linear cases. A finite element based solution (1000 elements) is compared with a real-time (instantaneous, ~1sec) solution. CADLM's ODYSSEE (Lunar) [2] [3] was used for the ROM modelling as well as conducting the DOE runs.

3.3 Sled with airbag and uncertain impact velocity

A typical sled test is considered. An FEM model is constructed using LS-DYNA dummy and the results of the FEM model are obtained for a DOE of 9 Optimal Latin Hyper Cube augmented by 6 additional "space filling" type samples. A DOE of 15 cases was first constructed and the results were computed via LS-DYNA. CADLM's ODYSSEE (Lunar) software was used [2]. The results of the pelvis acceleration and chest displacement are compared [Figure 5. & Figure 6.] Additional parametric studies were also conducted and animation screen shots are also obtained and compared.

4 Figures and Tables

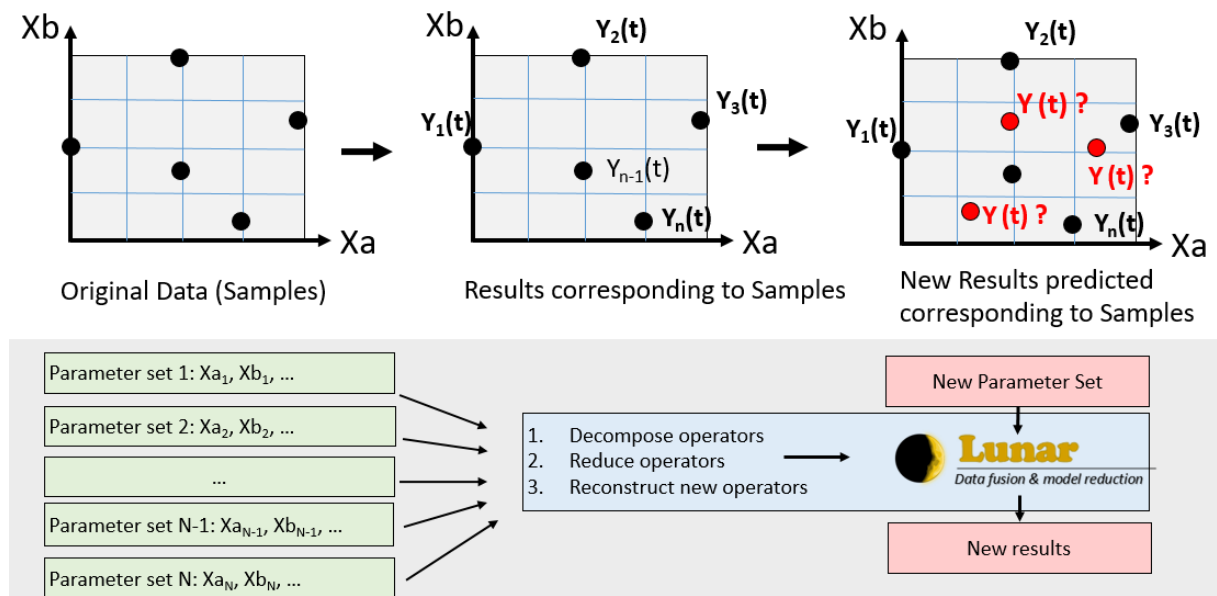


Fig.1: Procedure for ROM modeling (using ODYSSEE (Lunar) Software).

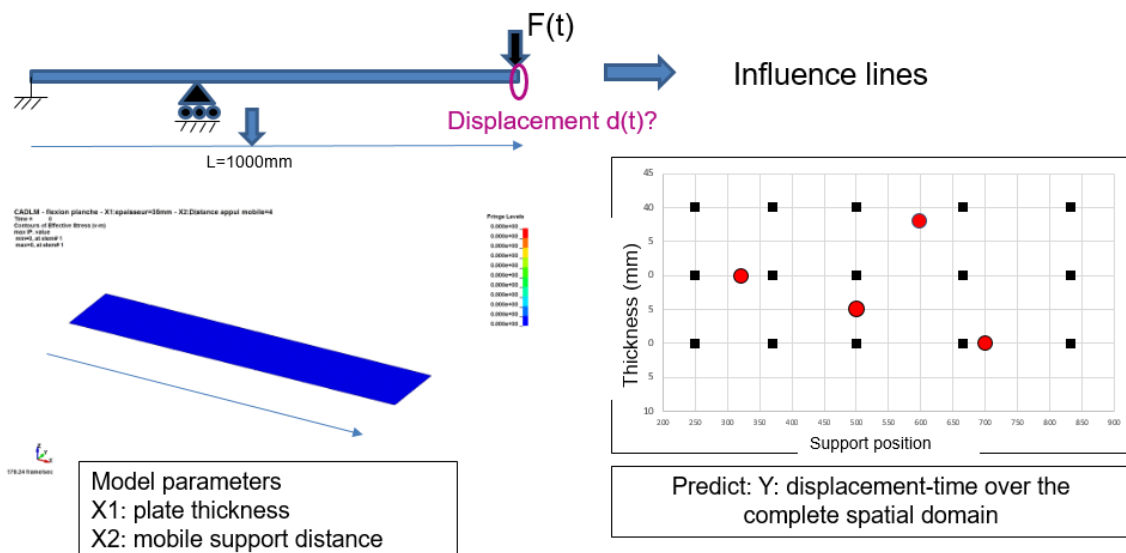


Fig.2: Computing influence lines for moving support point – Black square points represent the DOE samples and the red circles represent the prediction points.

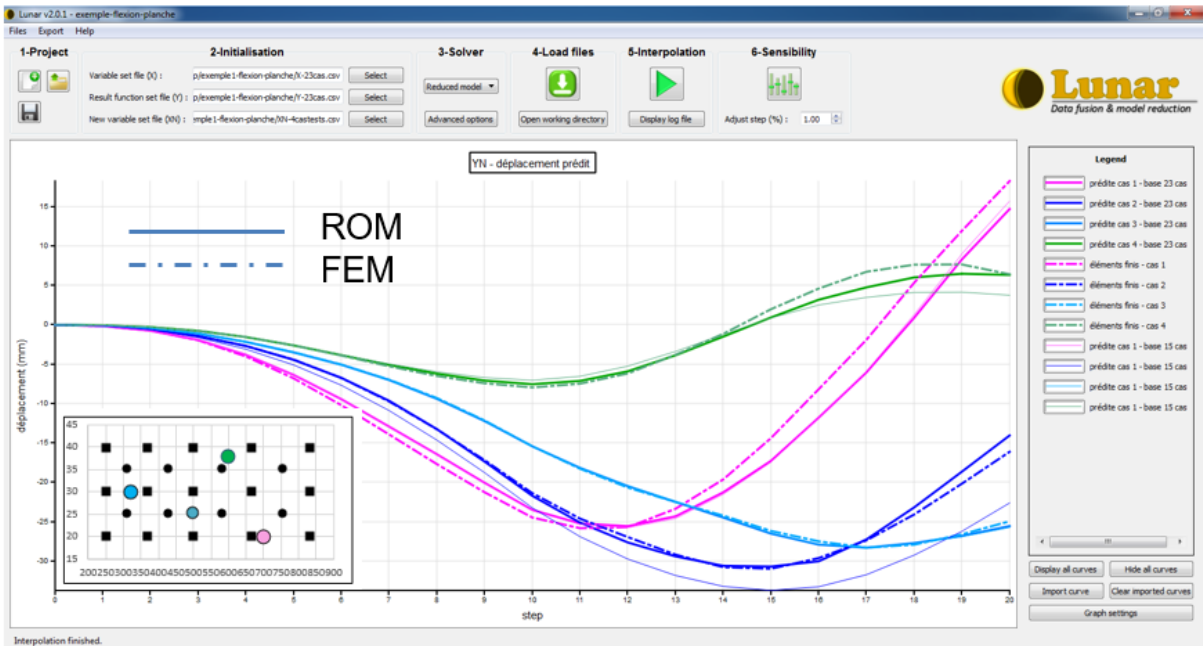


Fig.3: Computing influence lines for moving support point at four combinations of model parameters (colored circle points) – Results (Finite Elements versus ROM)

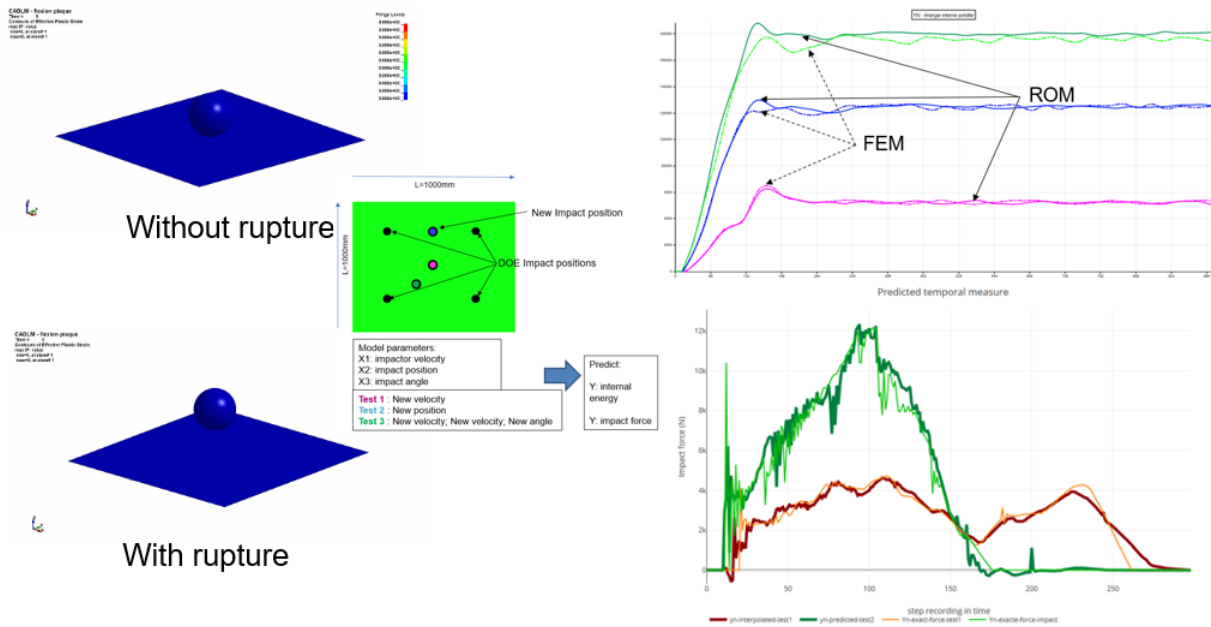


Fig.4: Plate-sphere impact (representing bonnet-head in pedestrian case) including possible rupture of composite material – Results (Finite Elements versus ROM)

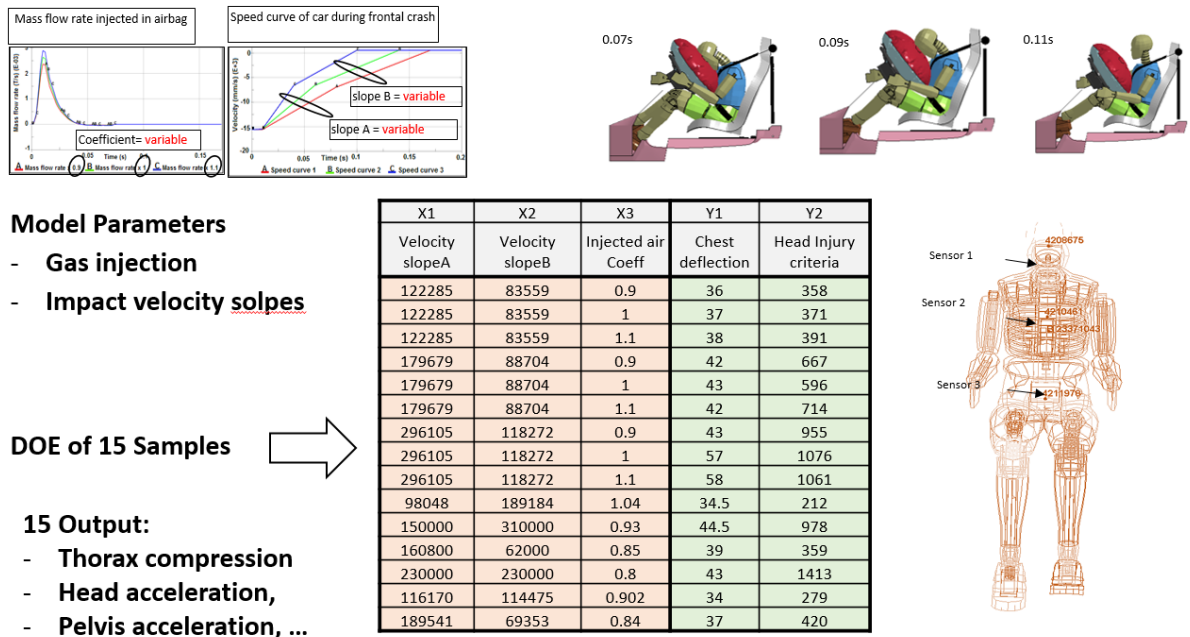


Fig.5: Sled test + airbag with variations on airbag mass injection and impact velocity - DOE with 3 parameters, 9 (OLH) + 6 “space filling” samples

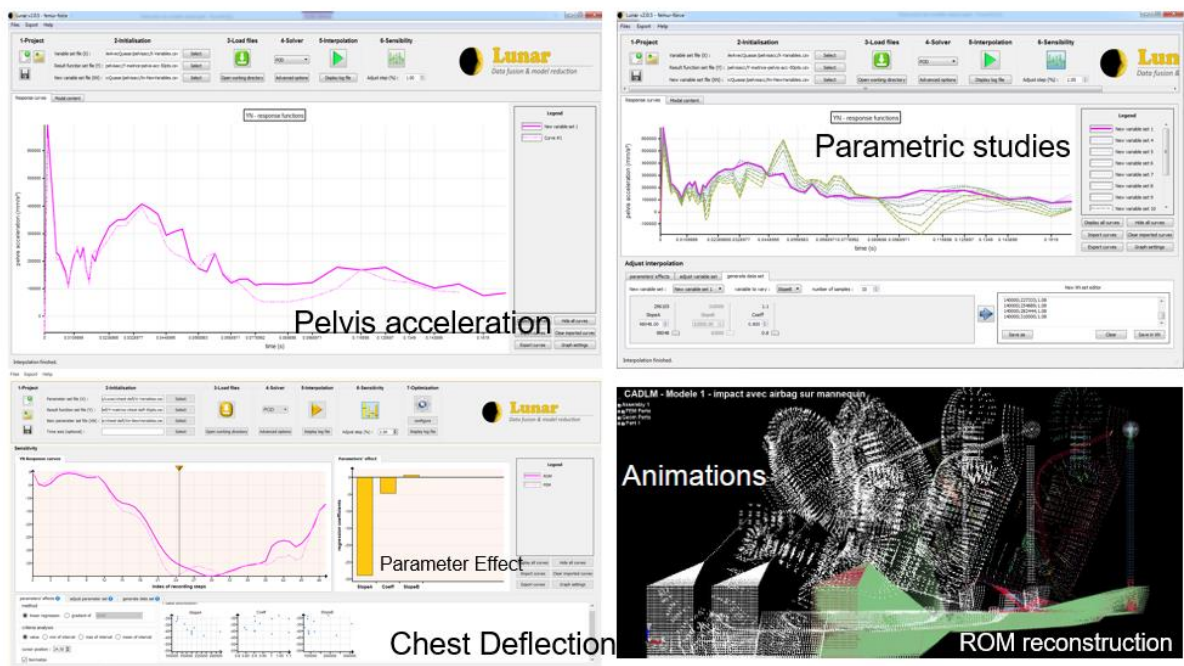


Fig.6: Results of sled test comparing FEM (LS-DYNA) and ROM (ODYSSEE/Lunar). (1)

5 Summary and conclusions

All three comparative studies undertaken during this work were used to compare results accuracy and computing time. All the results were satisfactory in terms of differences with the FEM counterparts (difference in the range of 1-10%). All the ROM CPU times were at most in the order of a few seconds.

Results obtained by ROM lie within 1-10% of the finite element counterparts. The computing time for ROM models is often in the order of seconds whereas the FE models range from minutes of computing (applications 1 & 2) to hours for sled test model. In practice any new set of parameters could be studied in a matter of seconds or “quasi real-time” with a gain of many orders of magnitude in terms of computation time.

Finally, comparative performance and portability studies were also investigated on an on-board version (Raspberry PI 3) of ODYSSEE/Quasar (The results will be published in a separate work). It has been shown that nearly all the presented applications may be conducted on a very small CPU while the current limitations concern the storage available on such on-board devices.

6 Literature

- [1] Kayvantash, K.: "European LS-DYNA Conference", Wurzburg, Germany, Proceeding, 2015
- [2] Kayvantash, K.: "ODYSSEE (Quasar/Lunar) software package for Machine learning, Model Fusion, Forecasting and ", CADLM, www.cadlm.com , Wissous, France, Release 9/2017
- [3] Le Duc, T., Kayvantash, K.: "Quasar Online software implementation for web-based Model Reduction and Machine learning", CADLM, www.cadlm.quasaronline , Wissous, France, Release 10/2018
- [4] Ovazza, H., Kayvantash, K.: "An onboard implementation of Quasar software for Raspberry PI 3 for Lidar/ Radar/ MobileEye data - Fusion and forecasting of Autonomous Vehicle Data recordings", Internal Report, CADLM, www.cadlm.com , Wissous, France, 8/2018
- [5] Yasuki, T.: "Application of reduced model to estimating Nij of HYBRID3 AF05 dummy in sled FE simulation, DYNA conference, Salzburg, Austria, 2017
- [6] Yasuki, T.: "Application of Reduced Model to Simulations of Occupant Protection and Crashworthiness at Toyota", Advanced CAE Division, Toyota Motor Corporation, 15th International LS-DYNA® Conference & Users Meeting, Detroit, USA, 2018