Model Reduction Techniques for LS-DYNA ALE and Crash Applications

Kambiz Kayvantash[1], Amadou-Tidiane Thiam[1], David Ryckelynck[2], Samir Ben Chaabane[3], Josselyn Touzeau[3], Philippe Ravier[3]

[1]CADLM SARL (www.cadlm.com)

[2]Mines Paris Tech (http://matperso.mines-paristech.fr/Personnel/david.ryckelynck/&?lang=fr)

[3]SILKAN SA (www.silkan.com)

1 Model Reduction

Model Reduction Techniques (MRT) are algebraic approximation solutions allowing for fast (real-time) interpolations (reconstruction) or extrapolations (prediction), based on previously existing DOE-type results, obtained either from FE computations or directly from constructions of reduced FE solutions. In a sense reduced models are subsets or decomposed domains of the solutions allowing for reconstruction of all spatial or temporal domain response.

Contrary to FE where global interpolations are based on local "shape" (geometrical) functions, reduced models are based on basis functions which include not only geometrical but also material, boundary conditions and loading. This contributes greatly to fast solver solutions for on-board computing and may be used for time dependent approximations.
Further, unlike response surface methods where smoothed solutions on certain criteria are obtained, MRT's provide complete solutions (reconstructions) of the space-time response of the original differential equation.

Keywords – Model Reduction, POD, Non-linear explicit, Crash, Safety, ALE

2 Stat of the art

Many methods such as Kriging, Neural Networks, Radial Basis Functions, PCA, Kernel PCA, have been proposed for reconstruction of response surfaces or reduced models based on sampling or time-history recordings. These are potential solutions which may be employed for the spatial-temporal decomposition of the response domain in consideration. Unfortunately they lose their computational efficiency for the treatment of temporal dimension with high discretization, especially for non-linear explicit time integration techniques, with small time-steps. More generally and with higher computational efficiency, two families of solutions are often considered. The first is called PGD (Proper Generalized Decomposition) [1],[2] which is an intrusive method (NOTE: by intrusive or *a priori* methods we mean that one needs to modify the solver code and structure in order to obtain a reduced set of equations to resolve), based on a decomposition of the initial set of equations via a separation of variables procedure, involving assumed functions for some dimensions of the problem. As a crude example beam and shell type elements may be considered as PGD solutions for 3D solid elements as could also be considered SPH solutions for fluid dynamics simulations. The second, non-intrusive, is the POD (Proper Orthogonal Decomposition)[3][4]. While the PGD is based on a decomposition of the original differential equations, for obvious reasons (mainly not having access to the solver source code) it is not adapted for general purpose applications. We shall therefore consider only an implementation of POD in this paper which is non-intrusive (or *a posteriori*) and needs no access or modification of the source code. However, it may be shown that the POD is by no means less efficient when considering the total number of computations needed for the reconstruction of the model. For a comprehensive description of the related theory we refer the reader to the works of [5][6][7][8].

Apart from a few matrix algebra manipulations, the algorithms are straight forward and may be applied with more or less efficiency to LS-DYNA or any other explicit finite element models. We shall briefly present two applications for crash (Lagrangian) and for missile impact simulation requiring (ALE) solutions. Other applications such as safety or multi-physics are also typical candidates since they require a complete and simultaneous simulation of all model components. Notice that these applications are often very time consuming (such as crash barriers or airbag ALE or CFD simulations) irrespective of the objectives of simulation. Additionally the interest in the modelling of such computationally expensive components (missile, barrier, airbag) is secondary and the real objective (criteria) of the simulations are often structural (damaged parts) or human biomechanical response. Additionally while the inclusion of these models in a full model may be considered affordable, their applications for either direct or surrogate based optimization or parametric studies becomes computationally expensive and often prohibitive.

2.1 The POD method

In the following we shall present the basic steps of the POD method, which is a decomposition technique, allowing for real-time (on-board) solutions used in optimization or simulator technologies. The two solutions presented are generic and may be readily applied for other multi-physics applications such as coupled fluid/thermal-structure interactions. The engineering consequence of this decomposition is an encapsulation of the reduced model as two independent, uncoupled responses in space and in time of the original function which may be used in a full model as easily as a material law, a contact force or a loading-time function, etc.

The first step is to provide a FE solution of a complete domain including all structures (physical domains) with full output available at the boundaries of the reduced model (i.e. contact zones, ALE interface, etc.). We shall refer to these output time-histories as $\mathbf{Q}(\mathbf{x},t)$. The basic idea behind POD is a simple decomposition strategy, allowing for the spatial and temporal domains to be handled in a decoupled manner and enabling the spatial-temporal response to be reconstructed via a multiplication of two uncoupled fields. Indeed we are looking for a decomposition of the type:

$$\mathbf{Q}(\mathbf{x},t) = \mathbf{q}(\mathbf{x}) \, . \, \mathbf{V}(t)$$

Once this decomposition is achieved, for any new position vector $\mathbf{x}_{new}$, $\mathbf{q}(\mathbf{x}_{new})$ may be computed (via RDA – Redundancy Analysis - Kriging or RBF, etc.) and subsequently multiplied by the $\mathbf{V}(t)$ matrix in order to obtain the new $\mathbf{Q}(\mathbf{x}_{new},t)$. The decomposition may be achieved via a singular value decomposition (SVD) of the matrix $\mathbf{Q}_{m,n}$. The algorithm is presented in (Fig.1:) where *m* represents repetitions of the response vector and *n* represents the recorded time intervals following an adequate sampling (such as OLH – Optimal Latin Hypercube) of the space parameters $\mathbf{x}$, or a PCA based method of the same samples (see Fig.2:), called the “POD snapshots”, using the autocorrelation matrix $\mathbf{R}=\mathbf{Q}^T\mathbf{Q}$ which is symmetric and positive definite or the matrix of information $\mathbf{K}=\mathbf{Q}\mathbf{Q}^T$ notice that $[\mathbf{K}_{m,m}] >> [\mathbf{R}_{n,n}]$ with ($m>n$) which makes it computationally less efficient (note that both $\mathbf{R}$ and $\mathbf{K}$ share the same the eigenvectors).

3 Applications

We shall present a simple crash type application allowing for the reconstruction of complete time-histories of a typical crash scene using only a few runs obtained via an Optimal Latin Hypercube design procedure. The “POD snapshots” method is used. The reconstructed responses may then be used for further predictions, allowing for parametric studies or real-time simulations to be performed with little computational effort. A second ALE type application where the “POD SVD” method is used and will be presented during the oral presentation (omitted here for page limit considerations) and may be obtained by contacting the first author (see Fig.9:, Fig.11:, Fig.11:).

3.1 "Formula Student" nose analysis

In the following example the robustness of a “formula student” type vehicle in order to improve its impact bearing capacity was studied [9] (Fig.3:). The main crash criteria were considered to be related to, among other criteria, the acceleration of the head back of the vehicle (sensor acceleration points). This was considered to be linked to the response of the nose, registered at its back side (normal force). A simple DOE type study was considered insufficient since the whole time response was needed for evaluations and not only peak values which could be obtained from some response surface method. The idea with the reduced model is to reconstruct the complete time-history of the output (for parametric studies and optimization) based on a POD (snapshots are used here) methodology.

The time history outputs were observed at 14 time increments over 2ms, (not to be confused with time steps). This number is relatively low and has only been chosen arbitrarily for demonstration purposes. In practice, more time increments – up to a reasonable factor of the total time steps – may be considered (Fig.4:). Our objective is to reconstruct the whole time-history from these selected 14 time observations.

3.1.1 Sampling and POD reconstruction

Subsequently, an Optimal Latin Hypercube sampling (OLH) with 100 runs (6 variables/parameters, 14 responses) was conducted allowing for variations of ±5% on design variables and environmental parameters. The nominal values were:

Design Variables:

Young modulus	Elastic limit	Thickness of selected elements
70GPa	300MPa	1.6mm

Environmental Parameters:

Initial speed	Chassis beam thickness	Chassis shell thickness
20m/s	3.2mm	3.5mm

An output file (corresponding to the $\mathbf{Q}(\mathbf{x},t)_{(100,20)}$ matrix) was recorded including **x** (6 input) and **y**(14 output) . Among the 100 runs, 80 were used for the reconstruction and the remaining 20 were used for verifications and predictions ($[\mathbf{x}]_{80,6}$ & $[\mathbf{y}]_{80,14}$). The Snapshots method was used to compute the reduced basis. Analysis of the eigenvectors demonstrates that only 6 first eigenvectors provide ~100% of the response necessary for reconstruction (Fig.5:). Since the norm of the eigenvectors (their "size") are associated with the value of the eigenvalues, if we conserve $[\lambda]_{5,5}$ then $[\phi]_{80,14}$ is nearly equivalent to (approximated by) $[\phi]_{80,5}$.

We observe that the POD implementation reconstructs well the first 10ms while it approximates closely the rest, up to 20ms (Fig.6:). Note that what is an approximation here is not the POD itself but the RDA (regression type) model – and the low number of time steps, i.e. 14, adopted here for demonstration purposes - of the $\mathbf{q}(\mathbf{x})$ matrix and could be much improved via a Kriging or a RBF type model. One can plot an error estimate, showing that the reconstructed response is in good agreement with the initial FE simulation response (Fig.6:).

3.1.2 POD Prediction

It is now interesting to compare the POD predictive capabilities using the 20 remaining runs which were not used for the basis computation (we only use $[\phi]_{80,5}$). We shall use a simple linear regression (first order reconstruction – Fig.:8) in order to predict the remaining 20 outputs. Fig.7: presents the outcome showing very good correlation for an arbitrarily chosen run (out of 20) with an error magnitude of around 13%. The two algorithms are implemented in ODYSSEE package [10] and available for academic or industrial use.

4 Figures and Tables

Summary (SVD)

- 1) EF → $\mathbf{Q}(\mathbf{x}, t)$ $(t_0<t<t_N)$
- 2) DOE → $\mathbf{Q}_{(m, n)}$; $\mathbf{x}_{(m, p)}$
- 3) SVD → $\mathbf{Q} = \mathbf{U}_{(m, m)\,D}\, \mathbf{S}\, \mathbf{V}_{(m, n)\,T}$ (Note : QQ^T gives U and Q^TQ gives V)
- 4) RDA → $\mathbf{x} \mid \mathbf{U}_D\mathbf{S}$ $(O(\mathbf{x})=\mathbf{q}(\mathbf{x}))$
- 5) $O(\mathbf{x}_{new})$ → $\mathbf{q}_{new}$
- 6) $\mathbf{q}_{new} . \mathbf{V}_T = \mathbf{Q}(\mathbf{x}_{new}, t_N)$

Fig.1: Model Reduction based on Singular Value Decomposition

Summary (snapshots)

- 1) EF → $\mathbf{Q}(\mathbf{x}, t)$ $(t_0<t<t_n)$
- 2) DOE → $\mathbf{Q}\,(m, n)$; $\mathbf{x}(m, p)$
- 3) Offset + Normalization $\mathbf{Q}$ → $\underline{\mathbf{Q}}$ (temporal)
- 4) «PCA» → $\mathbf{K} = \underline{\mathbf{Q}}\,\underline{\mathbf{Q}}^T$; $\mathbf{KU}= \mathbf{SU}$
  - ($\mathbf{U}$ eigenvector of information matrix (spatial) $\mathbf{K}$; $\mathbf{S}$ = diag(λ)))
- 5) RDA → $\mathbf{x}|\mathbf{SU}$ $(O(\mathbf{x})=\mathbf{q}(\mathbf{x}))$
  - [Regression: $\underline{\mathbf{q}} = \mathbf{X}^*[\mathbf{X}^T \mathbf{X}]^{*-1}\mathbf{X}^{T*}\mathbf{q}$
- 6) $\mathbf{V} = \underline{\mathbf{q}}\, \mathbf{U} / \mathrm{sqrt}\,(\lambda\, r)$
- 7) $\mathbf{Q}(\mathbf{x}, t) = [\mathbf{V}r\, \mathbf{V}r^{\,T}]\, \underline{\mathbf{q}}$

Fig.2: Model Reduction based on Snapshots method

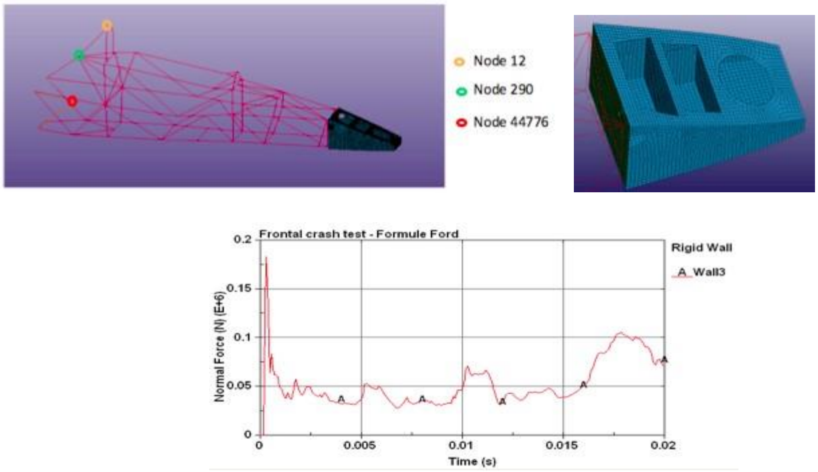

Fig.3: LS-DYNA simulations showing that the nose absorbs nearly 67% of the choc energy within 18 ms

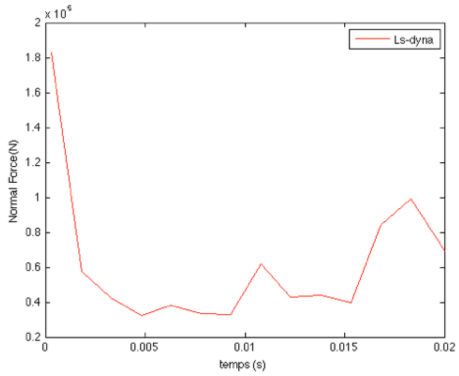

Fig.4: Typical time history recordings of rigid wall force (behind the nose) recorded at 14 time instances

Step	Eigenvalue	Value	Cumulated eigenvalue	Cumulated value in %
1	$\lambda 1$	17.8	$\lambda 1$	38.3
2	$\lambda 2$	12.2	$\lambda 2+\lambda 1$	64.5
3	$\lambda 3$	5.9	$\lambda 3+\lambda 2+\lambda 1$	77.3
4	$\lambda 4$	4.6	$\lambda 4+\lambda 3+\lambda 2+\lambda 1$	87.2
5	$\lambda 5$	3.5	$\lambda 5+\lambda 4+\lambda 3+\lambda 2+\lambda 1$	94.9
6	$\lambda 6$	2.4	$\lambda 6+\lambda 5+\lambda 4+\lambda 3+\lambda 2+\lambda 1$	100.0000
....	...	0.0000	....	100.0000
14	$\lambda 14$	0.0000	$\lambda 14+^{),13}(\lambda)$	100.0000

*Fig.5: Eigenvalues of the basis. Only 5 out of 14 need be retained for spatial basis (**q**(**x**)) reconstruction*

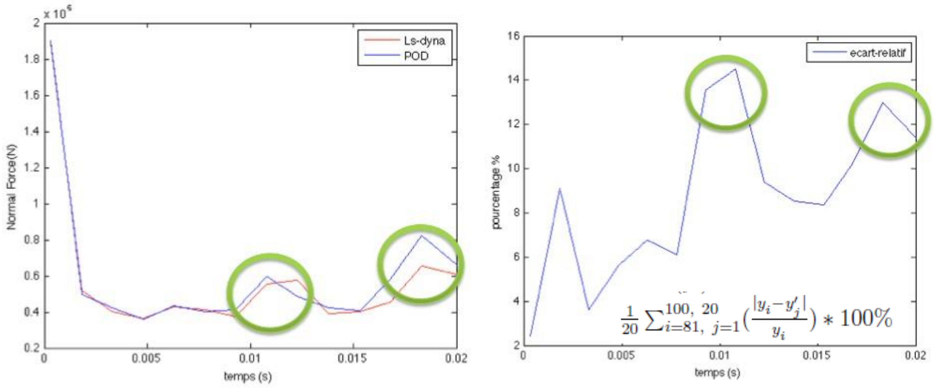

*Fig.6: Typical time history of rigid wall force **reconstructed** at 14 time instances (arbitrary selection of one among 80 runs used for basis construction)*

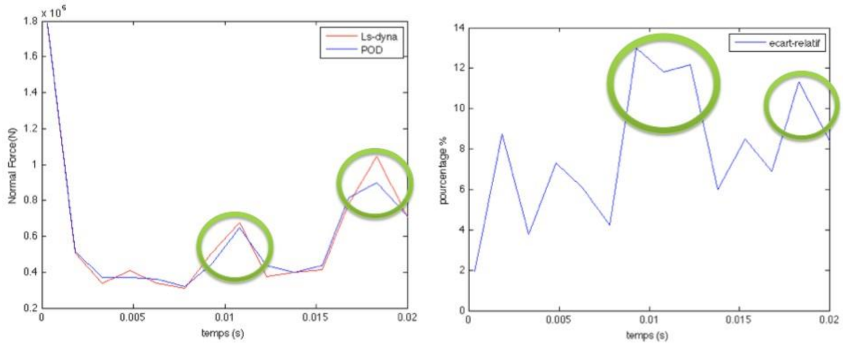

*Fig.7: Typical time history of rigid wall force **predicted** at 14 time instances (arbitrary selection of one among 20 unused runs)*

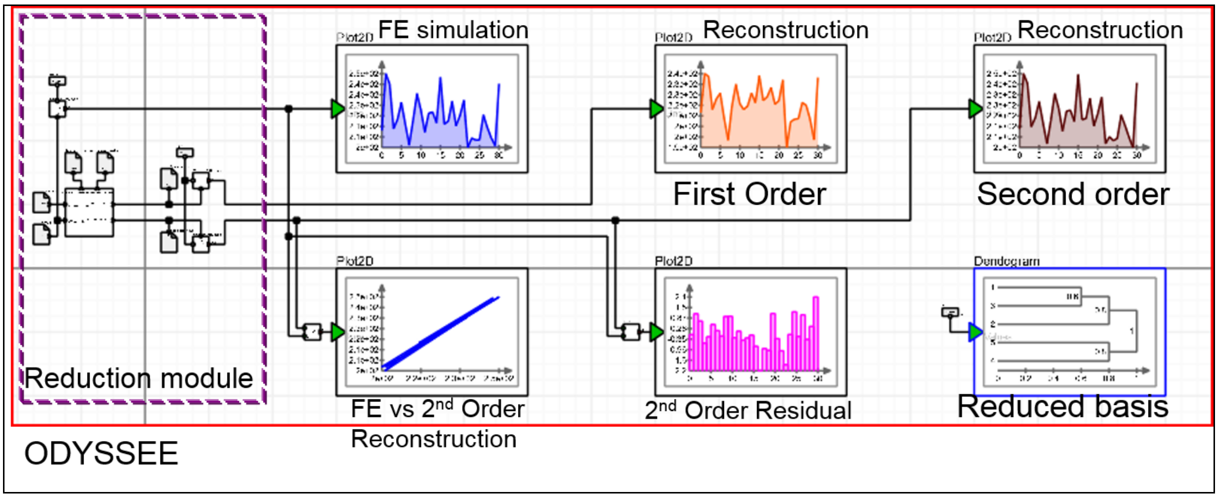

Fig.8: ODYSSEE Reduced Modelling Application for a 1st and 2nd order reconstruction of the response

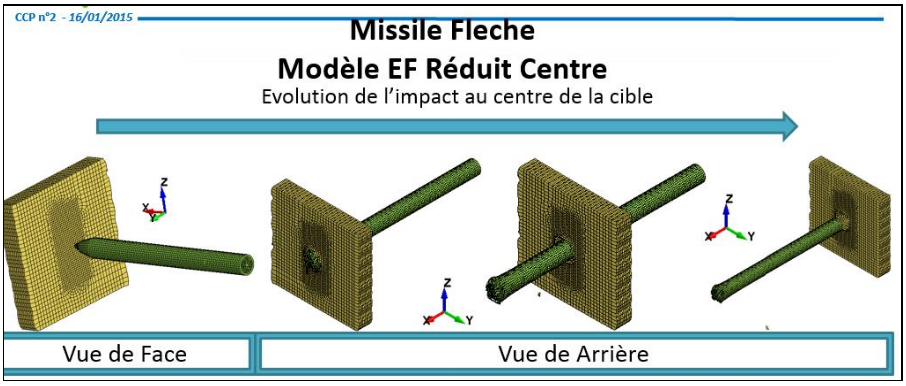

Fig.9: Missile impact zone with 10 Million elements

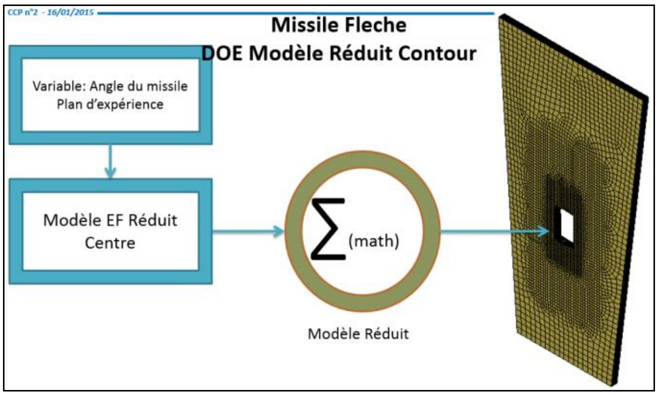

Fig.10: Missile impact zone replaced by its equivalent reduced model parametric studies reduced model to be used for parametric studies involving impact angle, velocity, mass, etc.

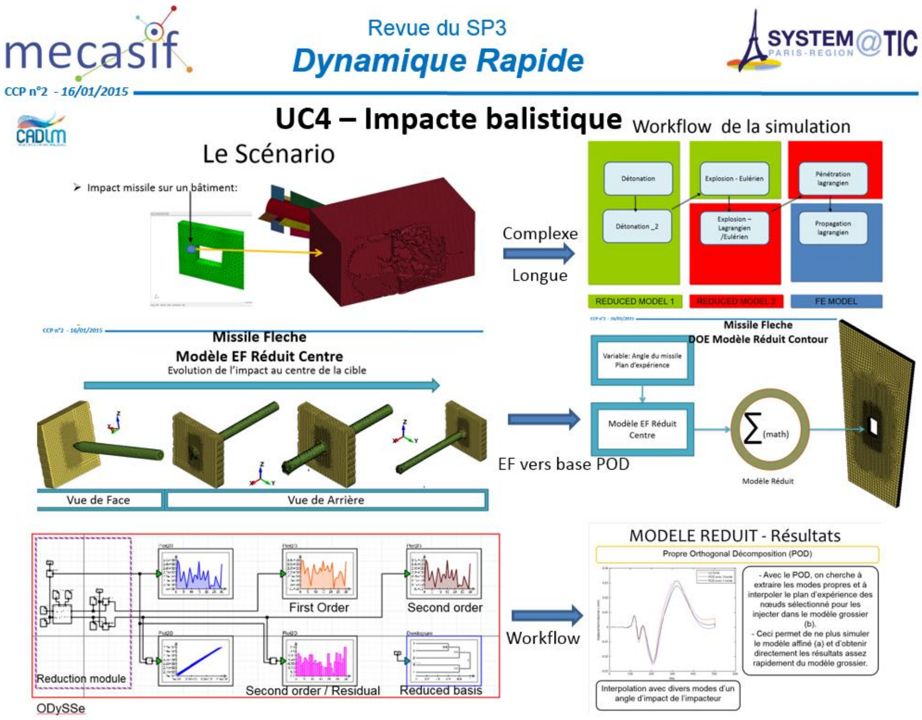

Fig.11: Application of POD method to ALE LS-DYNA for real-time (flight simulator) missile impact simulation conducted by CADLM for MECASIF project

5 Summary

We have presented a simple POD based methodology (snapshots and SVD) applicable to reduced modelling of explicit non-linear solvers for crash and ALE type simulations allowing for real-time, on-board computing of complex problems.

"The authors have been granted in part by the MECASIF project (2013-2017) funded by the French 'Fonds Unique Interministériel'-19. We wish to thank the financial support as well as the project partners for fruitful discussions and contributions.

6 Literature

[1] LADEVÈZE P., (1985c). Sur une famille d'algorithmes en mécanique des structures. Compte rendu de l'académie des Sciences, 300(2):41–44.

[2] BEYLKIN, G. et MOHLENKAMP, M. (2005). Algorithms for numerical analysis in high dimensions. SIAM, Journal on Scientific Computing, 26(6):2133–2159.

[3] KOSAMBI D.D., « Statistics in function space ». , J. Indian Math. Soc 7.1 (1943), p. 76–88.

[4] CHATTERJEE A. « An introduction to the proper orthogonal decomposition ». , Current Science 78.7 (2000), p. 808–817.

[5] RYCKELYNCK D., An a priori model reduction method for thermo-mechanical problems, Réduction a priori de modèles thermomécaniques, Comptes Rendus Mécanique, Volume 330, Issue 7, Pages 499-505, (2002).

[6] ASTRID P., Reduction of process simulation models: a proper orthogonal decomposition approach, Ph D thesis ISBN 90-386- 1653-8, Technische Universiteit Eindhoven (2004)

[7] AMSALLEM D., C. FARHAT C. http://web.stanford.edu/~amsallem/CA-CME345-Ch3.pdf

[8] RYCKELYNCK D., D. M. BENZIANE, Multi-level a priori hyper reduction of mechanical models involving internal variables, Computer Methods in Applied Mechanics and Engineering, Volume 199, Issues 17-20, pp. 1134-1142, (2010)

[9] ROSSELLO A., KAYVANTASH K., ROY, R. Masters Theses, Cranfield University, School of Applied science, (2012)

[10] KAYVANTASH K., ODYSSEE – A Decision Support System with Intelligence Kernels, CADLM, (www.cadlm.com) (2015)