Application of Model Order Reduction Techniques in LS-DYNA

Peter Friedrich[1,2], Uwe Reuter[2], Marko Thiele[1], Daniel Weigert[3]

[1]SCALE GmbH
[2]TU Dresden
[3]AUDI AG

1 Introduction

Model order reduction (MOR) aims to approximate a large dynamical system, like the nonlinear ordinary differential equations system of a finite element (FE) model of dimension n with a system of much smaller dimension m ([1]). This is called the reduced model. Model order reduction is widely used in implicit FEM codes. Because of the increasing attention towards explicit FEM, MOR techniques have been implemented in some explicit FEM codes as well and are starting to getting used in crash simulations ([1], [2]). The MOR methods used in this work are static condensation and the dynamic reduction with component mode synthesis (CMS). This method adds features of specific eigenmodes to the static condensation. Therefor the number of eigenmodes is a very important parameter. These methods are described in more detail in [3] and [4].
Characteristic for crash simulations are large deformations and nonlinear behavior in some areas of the model. Since the MOR methods mentioned above are linear, applying those methods to a nonlinear part would result in large errors. However, most areas of a crash model will have almost linear characteristics, so they can be reduced with relatively small errors (see [1]).
FE simulation is an important part of today's automotive engineering process. Crash simulations are used to reach the design goals regarding security and functionality. Hundreds of iterations are needed, to optimize each individual part of the car. Every iteration is evaluated by one or more simulations. Based on the results, conclusions can be made and an improved version of the previous model can be developed. Multiple iteration processes are happening in parallel, focusing on different parts of the model. At specific points in time, all changes are merged into a new, up to date model, which is the base for new iterations. This leads to an increasing complexity in the FE model, and therefore longer computing times of the crash simulations. Computing time is an important factor in the performance of the iteration process. New iteration steps can only be made, as soon as the results of the previous simulations can be evaluated. Longer computing times result in potential waiting periods and an overall slower process. Different approaches can be used to reduce the computing time of individual simulations. During the design of the front end of the car, details at the back might not be relevant. But since all disciplines use the same model, all parts are modelled with the highest amount of complexity available. By applying model order reduction techniques, parts that are expected not be relevant for the outcome of a specific simulation, can be approximated by reduced models of those parts. While the error introduced by this procedure might be relatively large in the direct surroundings of the reduced part, this can be irrelevant for the output that is of interest for this simulation.
The aim of this work is to evaluate the use of MOR in situations similar to crash simulations. In addition to that, a special focus will be the automation of MOR to make it usable in a typical environment and reduce manual adjustments to a minimum.

2 Module strategy for crashworthiness simulation

Finite element models for industrial crash simulations are very complex and can consist of hundreds of thousands of elements. In addition to that, there is a variety of different load cases that have to be considered. Therefore, the models are split into multiple substructures, that are stored as individual keyword files. These parts can be changed on their own. As long as the interface between adjacent parts is kept unchanged, a change in one part does not force a corresponding change in any other substructure. This is crucial to allow large teams of engineers to work on different parts of the model in parallel. Depending on the load case and the focus of development, different parts of the whole model have to be included in the simulation and others can be left out to save resources. This is done in an assembly step before the actual simulation. As mentioned above, the interface between adjacent substructures is an important part of the modelling. In this paper, two variants of interface modelling are considered.

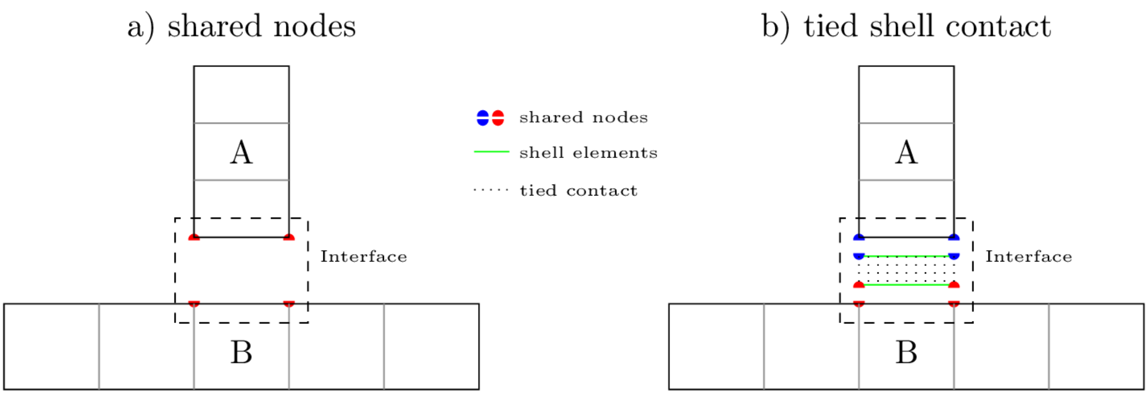


Fig.1: Different interface modelling techniques

2.1 Shared nodes

The first variant is the most straightforward technique to connect two parts. The elements that have direct contact to each other in the interface plane, share the same nodes. Figure 1a) shows a simplified example of two parts A and B, that consist of 8-node brick elements. The gap between the parts is only for a better visualization, the parts touch each other directly. The red nodes are shared between elements of the different parts and therefore connect the parts.

2.2 Tied shell contact

Again, we will look at the example of part A and B in Figure 1b). Instead of connecting the parts directly, additional shell elements are created in each part. These shell elements share nodes with their respective parts. The shell elements are then connected by the solver through the keyword **`*CONTACT_TIED_SHELL_EDGE_TO_SURFACE`**. While the advantages of this modelling convention might not be obvious in this very simple example, they become important as the models get more complex and especially the number of elements in the interface plane grows.

2.3 Comparison of the methods

2.3.1 Flexibility and reusability

To use shared nodes to connect substructures, these nodes have to have fixed node numbers. If the node numbers change, they have to change on both sides of the connection. In addition to that, one substructure can only be used one time at one specific location. For some scenarios, it can be useful to reposition individual parts depending on model attributes, i.e. left hand drive and right hand drive versions of the same car. Some substructures can even be used multiple times in the same model, like wheels for example. This flexibility is only possible with tied contact modelling.

2.3.2 Mesh compatibility

Since shared contact nodes, connect elements of different substructures directly, the meshing has to be considered as well. Nodes of both parts that will be connected, have to have the same coordinates. If there are no nodes that can be shared between substructure, a connection is not possible. Again, the tied contact modelling is much more flexible. This can be a big advantage, if a substructure is available in different mesh sizes or changes in detail over the development period.

2.3.3 Number of interface nodes

All nodes that are required to construct the interface that is needed for a substructure to be connected to others, will be called interface nodes. The number of interface nodes will be important for model order reduction. More interface nodes lead to worse performance of the reduced substructure. By choosing how many shell elements are used to connect the parts, the number of interface nodes can be controlled when using the tied shell contact. When using shared nodes, the number of interface nodes is determined by the number of nodes in the interface plane. This can lead to significantly more interface nodes depending on the meshing.

3 Model order reduction with LS-DYNA

The application of model order reduction in LS-DYNA can be split in three parts: the adaptation of the original model include, the reduction process itself and the import of the reduced model.

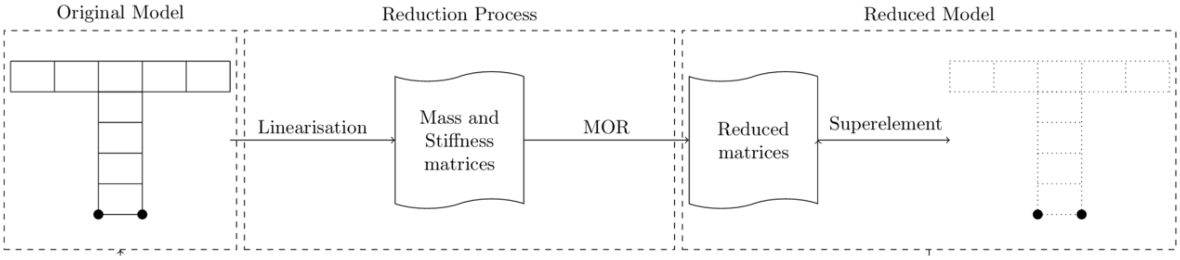


Fig.2: Major steps for model order reduction with LS-DYNA

A typical input deck for crashworthiness simulation with LS-DYNA consists of multiple keyword files that are joined in one master keyword file via ***INCLUDE**. Apart from general definitions like material cards, most of these keyword files contain independent parts of the whole model. The example in Figure 3 consists of the master include, 3 different submodels and the material definitions. We assume part "Mod2" is a good candidate for linear model order reduction, i.e. “Mod2” can be replaced by a reduced representation “RMod2”. The reduced component "RMod2" consists of two parts: The template for importing the reduced matrices and the matrices themselves. Besides exchanging "Mod2" with "RMod2" no change to the rest of the model or the master keyword file is required.

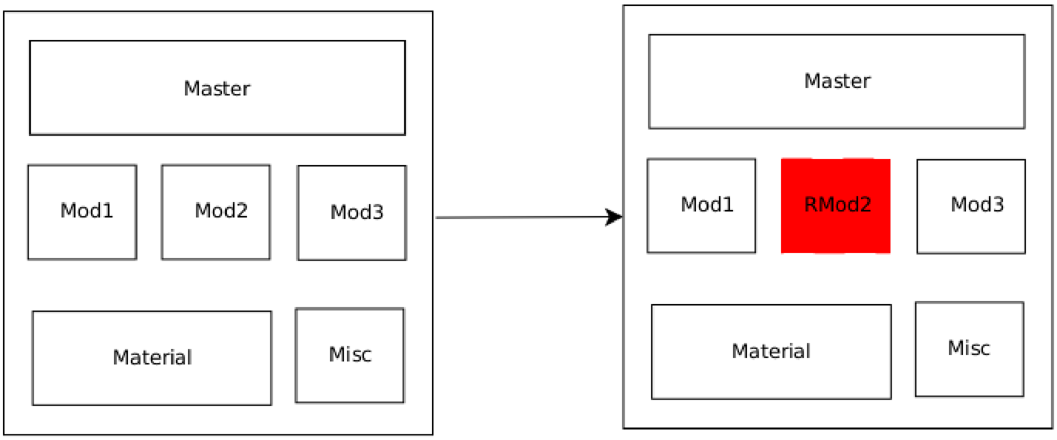


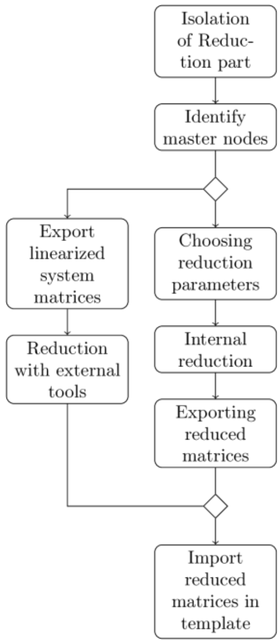
Fig.3: Exchange of submodel with approximation obtained by MOR

3.1 Adaptation of the original model include

To make model order reduction possible for a new module, the include has to be modified and a new reduced model template has to be created. This assumes, that the module contains only elements that can be used with the LS-DYNA matrix exporting and internal reduction features. In this work, shell elements were tested successfully. In the module include, the master nodes have to be defined. This is done by creating a ***SET_NODE** that contains all master nodes. Another node set containing all nodes of the module has to be defined, if the reduction will be done externally. The set ID should follow a universal convention across all modules.
The corresponding reduced model template include can be created by taking the original include and removing all keywords except those, that represent the interface to other submodules. So, if the interface of the specific module consists of 3 shell elements, those have to be present in the template. We call this the reduced model template, because with a single include, one can use different reduction techniques and accuracies.

3.2 Reduction process

After preparing the module include there are two possible ways to perform the actual reduction: using LS-DYNA internal algorithms or exporting the mass and stiffness matrices and using external toolkits for the reduction. A special reduction keyword file that includes the module include, controls the offline reduction step. This can be used for all modules.
The internal reduction is triggered by the ***IMPLICIT_PART_MODES** keyword. By changing the parameters of this keyword, LS-DYNA can perform static condensation or CMS with different amounts of eigenmodes. The reduced matrices are exported to text files using a format called DMIG. For external reduction, the mass and stiffness matrices of the whole model are exported. The reduction with an external tool should produce a DMIG file with the reduced matrices as well.



3.3 Importing the reduced matrices

The reduced mass and stiffness matrices are imported into the reduced model template with the ***ELEMENT_DIRECT_MATRIX_INPUT** keyword. The solver creates a superelement with the properties given by the matrix input. Any entries in the input file that don't correspond to existing node IDs in the template file, are interpreted as generalized coordinates used for dynamic reduction methods.

4 Integration into simulation data management

Simulation data management (SDM) software allows to manage large FE models consisting of multiple include files. Depending on the discipline and the simulation configuration, different models can be assembled. This includes different variants of specific parts, i.e. right-hand-drive or left-hand-drive, different levels of interior equipment and different crash scenarios with the corresponding barriers. In addition to that, parameters can be set for initial velocities or the positioning of the model, to fit different regulations. To make model order reduction usable on a larger scale, the integration into existing SDM tools is crucial. At the example of the LoCo SDM software from SCALE GmbH, first approaches were tested.
The first and most straight forward approach is, to generate the reduced matrices directly when the corresponding part is modified and store them alongside the original include. During the assembly process, the SDM system can either choose the original part or the reduced variant depending on the configuration of this specific simulation. As all offline steps are already taken, this is the fastest and easiest way at assembly time.
The second approach integrates the reduction calculation into the assembly process. If there are any reduced parts included in the simulation, the corresponding reduced matrices are generated during the assembly. This has the advantage, that variables defined in the reduced parts and, more importantly, the reduction method itself can be set by the SDM system. The resulting reduced matrices respect those changes. If the reduced matrices were precalculated, as described in approach 1, variables regarding for example mass or stiffness attributes or the number of eigenmodes (regarding dynamic reduction with CMS), cannot be changed at assembly time. On the other hand, executing the reduction for every simulation is possibly redundant and leads to a higher resource usage. The time for the reduction step adds to the overall duration of the simulation. Depending on the time needed for the reduction itself and the potential time saving by using model order reduction, this can be a significant downside.
The third approach tries to combine the advantages of the other two. To allow different parameter variants of the reduced part or different reduction methods, the SDM system can store multiple versions of the reduced matrices. If a new simulation is assembled, the database can be queried for a reduced part that might already exist for the current configuration. When this is successful the part can be used directly, otherwise the reduction process is started by the SDM system in the background. As soon as this is ready, the assembly process can continue. The reduced model is saved by the SDM system for future uses. In addition to that, the SDM system can start the reduction process proactively. So, if the original part is changed, the most frequently used reductions can be done at a convenient time, i.e. when there is not much load on the cluster.

5 Examples

The example model is shown in Figure 4. It consists of a clamped frame and one of two different substructures ("I" and "T"), that are attached at the top right corner of the frame. The system is excited

by a sphere impactor moving from left to right with an initial velocity. The FE model uses shell elements, that form a hollow beam with a quadratic cross-section. The displacements shown in the following plots are measured at a node in the top right corner of the frame.
The attached substructure is reduced with the LS-DYNA internal static condensation and dynamic reduction. For the dynamic reduction, multiple variants with different parameters are considered: CMS with 1 eigenmode, 5 eigenmodes and 100 eigenmodes. The displacements of the simulations with reduced substructures are compared to the reference simulation with the original substructure includes.

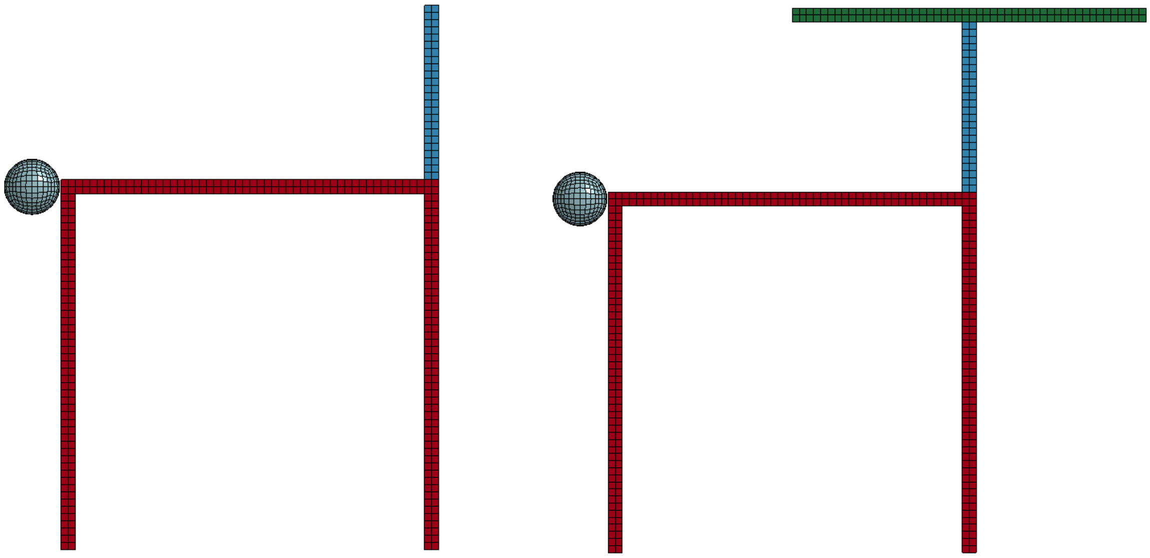


Fig.4: Example models: with "I" substructure on the left, with "T" substructure on the right

5.1 Approximation quality

Figure 5 shows the results for the "I" substructure. The first peak of the displacement curve is approximated well by all variants. Even though the static condensation shows the largest errors, the amplitude and shape are very similar to the reference curve. Only after about 0.5ms the static condensation starts to deviate significantly in terms of mode shape and frequency, which also results in big amplitudes in the error plot. The variants of dynamic reduction however, converge to the reference curve. Even with only one eigenmode considered, the mode shape and absolute values of the original simulation is approximated pretty well. With increasing eigenmodes the approximation is even better. The higher amplitudes in the error plot towards the end of the simulation time can be explained by the nature of explicit time integration, where errors accumulate over length of the simulation. All in all, it is noticeable, that for this very simple model, even one eigenmode in the dynamic reduction results in a very good approximation of the reference solution. In addition to that, the static condensation may be sufficient for some applications. Especially if only the first peak in displacements is of interest.

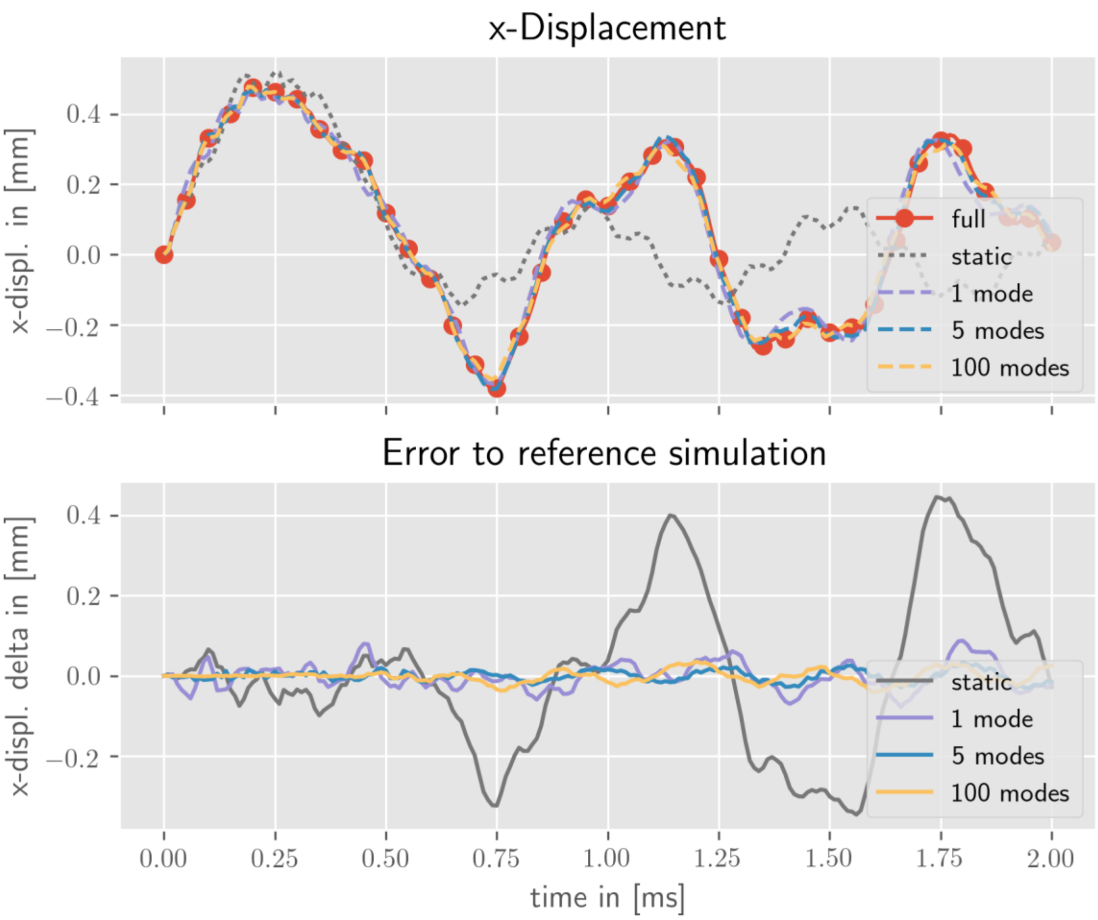


Fig.5: Comparison of displacement curves for different reduction variants to the original model ("I" substructure)

Figure 6 shows the results for the "T" substructure. The static condensation curve diverges from the reference significantly right from the beginning. Its first peak is reached at about 0.2ms and has a lower amplitude than the first peak of the reference at 0.24ms. In addition to that, the displacement curve for the static condensation shows a drastically different frequency and mode shape. This results in increasing deviations from the reference as clearly shown in the error plot. The dynamic reduction with 1 mode is a little better than the static condensation, but not a good approximation for the reference simulation. The 5 modes variant is already very close to the original, but CMS with 100 modes is still better, as can be seen in the error plot.

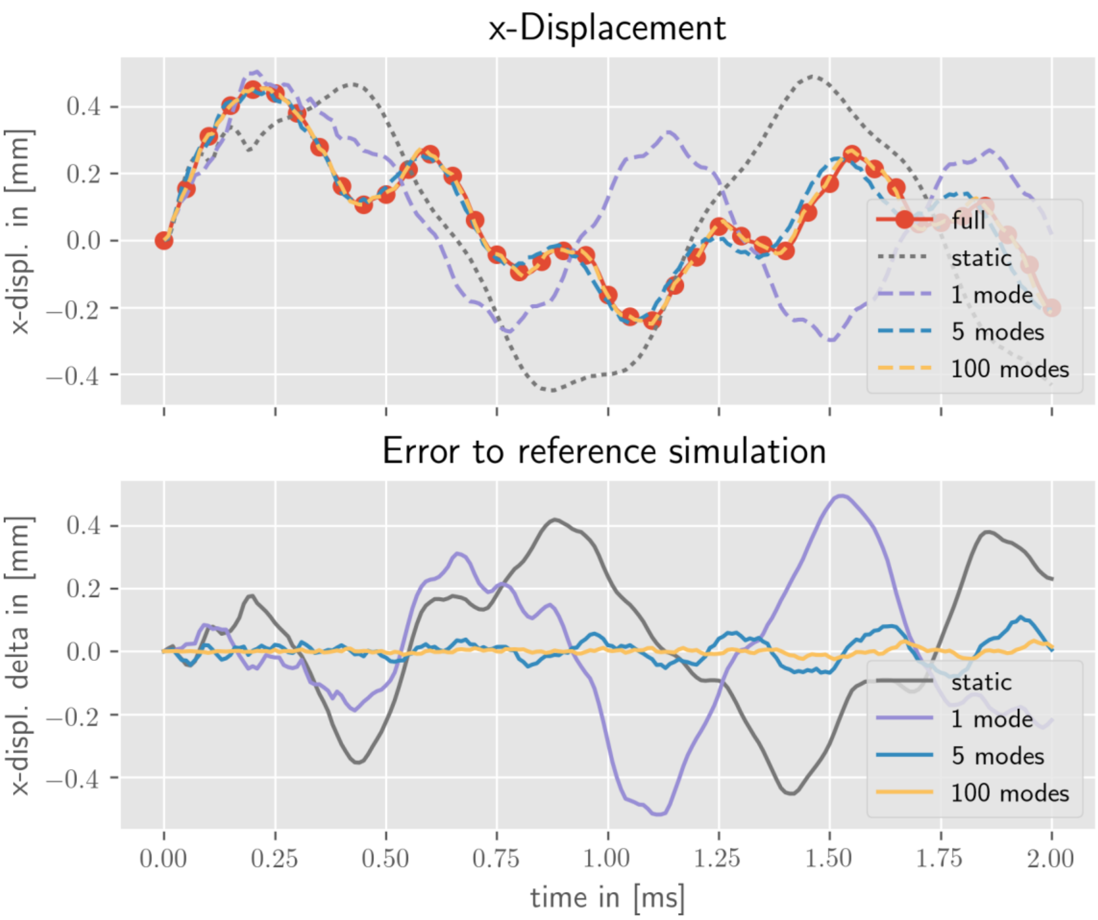


Fig.6: Comparison of displacement curves for different reduction variants to the original model ("T" substructure)

5.2 Comparing calculation times

To compare calculation times of the model with a reduced part to the original model, each simulation was run 10 times. The results will be visualised as boxplots. The lower and upper boundary of a box are the first and third quartile of all data points. The ends of the whiskers represent the lowest and highest datum still within 1.5 times the distance between those quartiles. Outliers are marked with circles and the line inside the box marks the median. Although the sample size is relatively small, some conclusions may be drawn from the obtained data.

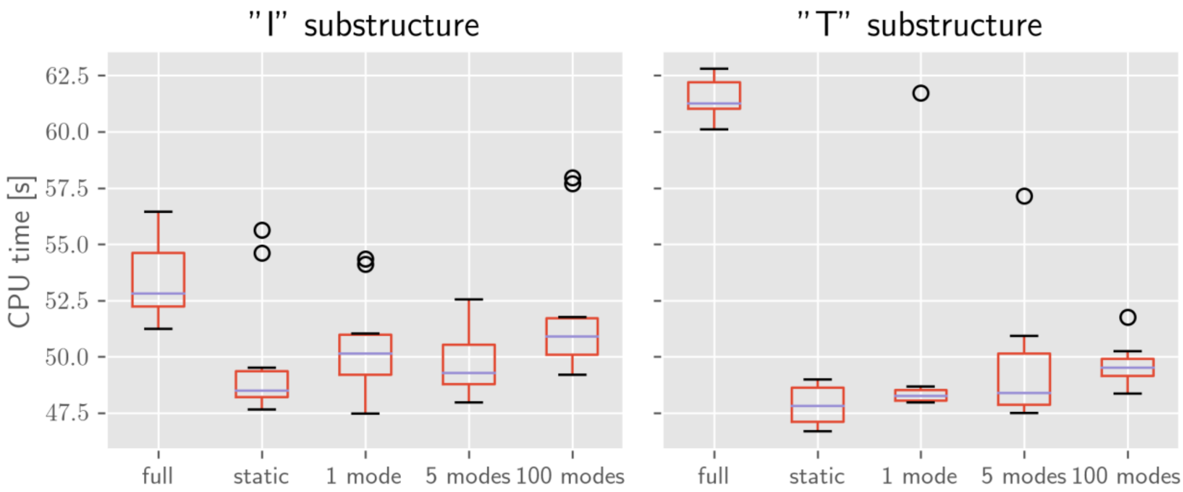


Fig.7: Comparison of simulation times for different reduction variants and model sizes

The Tables 1 and 2 compare the median CPU time and number of degrees-of-freedom (DOF) in more detail. Since the "T" substructure has more DOFs that can be reduced, the reduction in CPU time with respect to the original model is bigger than for the "I" substructure. The differences between static condensation and the different variants of CMS are relatively small, although it can be observed that more DOFs result in longer computation times for dynamic reduction. But the significantly better approximation by dynamic reduction with more eigenmodes, as discussed in section 5.1, justifies the slightly longer computation. Overall the speedup that can be achieved by model order reduction is significant.

Table 1: DOF and CPU time change with respect to original model ("I" substructure)

	DOF	CPU time [s]	DOF change [%]	CPU time change [%]
full	8484	52.81		
static	7326	48.48	-13.649222	-8.1992047
1 mode	7327	50.12	-13.637435	-5.0937322
5 modes	7331	49.28	-13.590288	-6.6843401
100 modes	7426	50.90	-12.470533	-3.6167393

Table 2: DOF and CPU time change with respect to original model ("T" substructure)

	DOF	CPU time [s]	DOF change [%]	CPU time change [%]
full	10890	61.26		
static	7326	47.80	-32.727273	-21.971923
1 mode	7327	48.25	-32.718090	-21.237349
5 modes	7331	48.39	-32.681359	-21.008815
100 modes	7426	49.50	-31.808999	-19.196866

6 Conclusions

The LS-DYNA internal reduction methods were successfully applied to models consisting of shell elements. A good approximation of the original model was achieved in a system with highly dynamic excitation, similar to crash simulations. Depending on the reduced part, even dynamic reduction with very few additional degrees-of-freedom resulted in a very good approximation. The use of external tools for the reduction step was successful using the DMIG interface to and from LS-DYNA. The process of adjusting new parts to be used in a reduced representation could be documented. By integrating the process into a SDM system, using different reduction methods and choosing different combinations of original parts and reduced parts was possible without manual adjustments. Possibly laying the groundwork for the application in more complex models.

7 Literature

[1] Fehr, J.: "Interface and Model Reduction for Efficient Explicit Simulations – a Case Study With Nonlinear Vehicle Crash Models", Mathematical and Computer Modelling of Dynamical Systems, vol. 22 number 4, 2016, 380-396

[2] Cobanoglu, A.: "Model Order Reduction Methods for Explicit FEM", Science in the Age of Experience, 2016

[3] Qu, Z.-Q.: "Model order reduction techniques: with applications in finite element analysis", Springer, 2004

[4] Paulke, J.: "Optimal Combination of Reduction Methods in Structural Mechanics and Selection of Suitable Intermediate Dimension", TU Dresden, 2014