Model Order Reduction Methods for Explicit FEM

Model Order Reduction Methods for Explicit FEM

Ali Cagatay Cobanoglu[a*], Simon Mößner[a,b], Majid Hojjat[b] and Fabian Duddeck[a]

[a]Technische Universität München, Arcisstr. 21, 80333 Munich, Germany;

[b]BMW Group, Knorrstraße 147, 80788 Munich, Germany

***Abstract**: Abaqus/Explicit is a well-established and widely used FEM solver for crash and pedestrian safety simulations. However, due to the large number of degrees of freedom, simulation time is still a limiting factor especially in context of structural optimization. In typical crash simulation, a large portion of the model undergoes only elastic deformation. Hence, model order reduction (MOR) methods can bring a significant decrease in the computational time. While Abaqus/Standard already offers several reduction methods, MOR is currently not applicable for Abaqus/Explicit. The purpose of this work is to enable MOR for explicit finite element models by use of superelements. Reduced mass and stiffness matrices are generated by Abaqus/Standard and transferred to the explicit solver through a VUEL subroutine. The method is applied successfully to low speed vehicle crash simulations. The achieved results show a significant gain in computational time and underline the potential of MOR for Abaqus/Explicit.*

***Keywords**: explicit FEM, model order reduction, Guyan reduction, superelement, crashworthiness, low-speed crash.*

1. Introduction

Explicit FEM is an effective and efficient approach for solving complex three-dimensional dynamic problems. Especially, it is appropriate for the simulation of impact loads with multiple contacts thanks to its small time step size. In advanced commercial explicit FEM software, such as Abaqus/Explicit, the computational time is dependent on both the number of elements and the element size normally. When the critical element size is kept constant in the model, the computational time scales almost linearly with the number of degrees of freedom, DOF. This motivates the development of methods to simplify the numerical models decreasing simulation time by, for example, reducing the number of DOF.

Model Order Reduction (MOR) has been widely used in implicit FEM codes for decades. There have been various MOR methods proposed, which obtain an improvement in the computation time by a slight loss in the accuracy of the results. Due to the increasing attention towards explicit FEM in the last decade, a number of developed MOR techniques have been implemented to some explicit FEM codes as well, some of which significantly decrease the computational cost (Faucher, 2003; Flídrová, 2010; Maker, 2002).

Impact simulation improving vehicular crashworthiness is a prominent application field for explicit FEM. Being a highly nonlinear phenomenon, optimization of car crash is currently mainly done by use of zero order algorithms, which require multiple executions of the crash simulation, e.g. (Duddeck, 2008). Typical FEM crash models have up to several million nodes and DOF; hence, an enormous amount of data has to be handled during the simulation. In low-speed crash simulations, only a small portion of the car models undergoes nonlinear plastic deformation. The rest of the car body structure, which is away from the impactor, deforms only in the linear elastic range. That part of the car model can be reduced to a limited number of DOF by use of MOR, without losing accuracy. This can significantly reduce the computational cost roughly with a comparable ratio as obtained between the number of DOF in the full and the reduced model. Currently, MOR is not available for Abaqus/Explicit. On the contrary, Abaqus/Standard is capable of creating the reduced models with different types of MOR methods. Here, the reduced part of the model is called *superelement*. The mechanical properties of superelements created by Abaqus/Standard, like mass, stiffness and damping matrices, can be used elsewhere and independently of Abaqus/Standard. This helps to implement the superelements in Abaqus/Explicit without the need to involve Abaqus/Standard after the reduction procedure. In this work, a VUEL user subroutine (Abaqus, 2016) is developed, which enables the use of superelements in Abaqus/Explicit. In the following section, the steps of this implementation are discussed in detail.

2. Superelement in Abaqus/Explicit

Using Abaqus/Standard for MOR, physical coordinates of a full-order model are replaced by physical or modal coordinates in the reduced-order model. Here, we focus on reduction of the model to the DOF of a small number of nodes, referred to as the interface nodes.

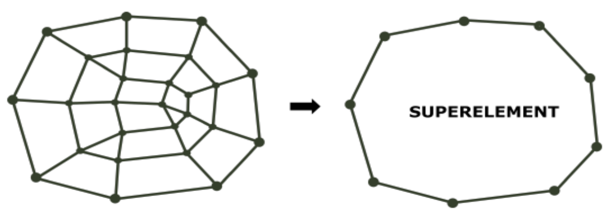

Figure 1. Superelement

In Figure 1, the full-order model (left) has *m* nodes and with each *n* degrees of freedom, hence in total $m \times n$ *DOF*. In Equation (1), the equation of motion of the full model is represented as:

$$\boldsymbol{M\ddot{u}} + \boldsymbol{C\dot{u}} + \boldsymbol{Ku} = \boldsymbol{f^{ext}} \quad \textbf{(1)}$$

M, *C*, and *K* are the mass, damping and stiffness matrices, respectively. *u* is the displacement vector and f^{ext} are the external forces. The reduced-order model (right) has *k* nodes, and therefore $k \times n$ degrees of freedom. The structural matrices and the external load vector of the reduced model are:

$$\boldsymbol{M_{red}} = \boldsymbol{T\ M\ T^T}, \quad \textbf{(2)}$$

$$\boldsymbol{C}_{red} = \boldsymbol{T}\,\boldsymbol{C}\,\boldsymbol{T}^{\boldsymbol{T}}\,, \tag{3}$$

$$\boldsymbol{K}_{red} = \boldsymbol{T}\,\boldsymbol{K}\,\boldsymbol{T}^{\boldsymbol{T}}, \tag{4}$$

$$\boldsymbol{f}_{red}^{ext} = \boldsymbol{T}\,\boldsymbol{f}^{ext}\,. \tag{5}$$

with $\boldsymbol{T}$ being the transformation matrix defined through the reduction algorithm. The role of $\boldsymbol{T}$ is to reduce the full-model matrices into reduced ones and to recover the solution in the full model with

$$\boldsymbol{u} = \boldsymbol{T}\,\boldsymbol{z} \tag{6}$$

After the reduction, the equation of motion is:

$$\boldsymbol{M}_{red}\ddot{\boldsymbol{z}} + \boldsymbol{C}_{red}\dot{\boldsymbol{z}} + \boldsymbol{K}_{red}\boldsymbol{z} = \boldsymbol{f}_{red}^{ext} \tag{7}$$

Figure 2 illustrates the steps to generate a superelement for Abaqus/Explicit. Firstly, the portion of the model, which undergoes only linear elastic deformation, is identified to be reduced as superelement. This part of the model is then modeled in Abaqus/Standard and the corresponding reduced matrices are exported. The obtained data is given to VUEL subroutine as superelement. This superelement is defined as a normal user element inside the input deck of the main analysis. In a final step, the created reduced-order model is run in Abaqus/Explicit: The implemented VUEL subroutine is responsible for all element calculations of the superelement during the analysis.

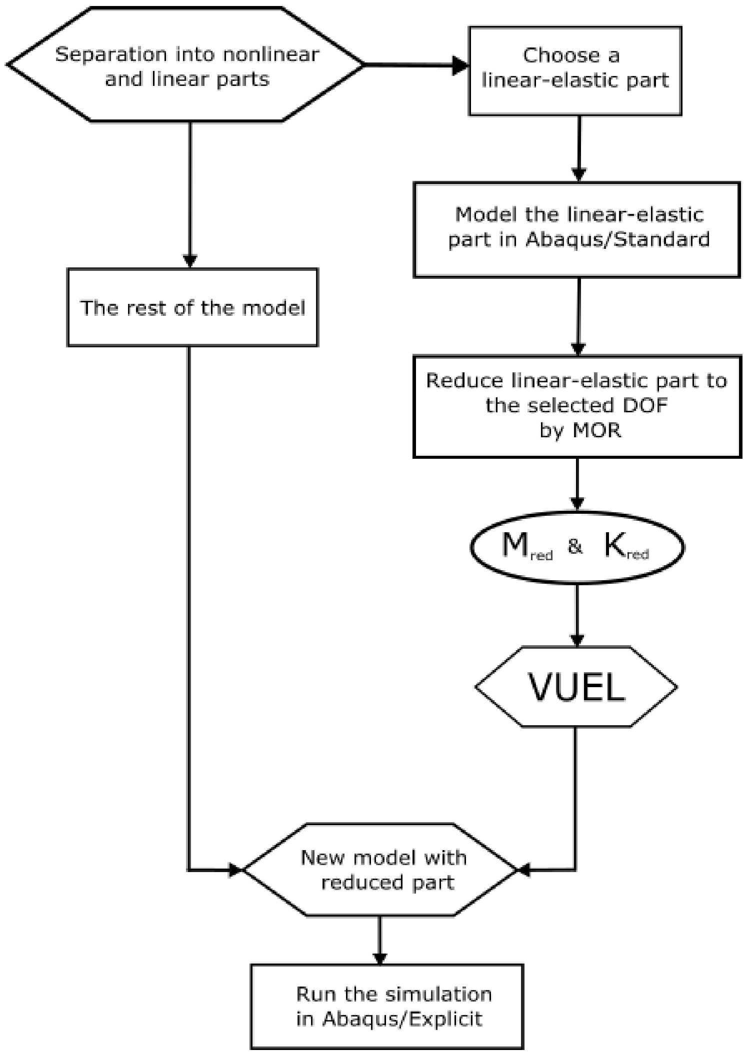

Figure 2. Flow chart of the superelement implementation to Abaqus/Explicit (here for the undamped case)

3. Numerical Application

The method and the implementation were tested by small 2D examples with an ideal matching of displacement, velocity and acceleration and then finally validated by a larger industrial test case. The former examples are not presented here for the sake of briefness. Hence, the potential of the method is demonstrated in this section via a body-in-white car model for an RCAR crash load case (Research Council for Automobile Repairs, 1999).

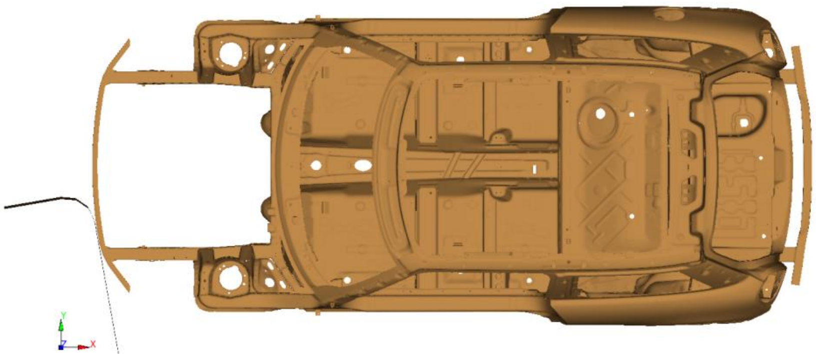

Figure 3. Full-order model of body-in-white

In Figure 3, a full body-in-white model and the RCAR barrier placed in front of the car are shown. Figure 4 shows on the right the part of the model which is assumed to deform nonlinearly and plastically. The rest of the structure is considered to show linear elastic behavior throughout the total crash simulation and can be reduced therefore by MOR. This separation is taken here as an example to illustrate the method. An ideal approach should distinguish linear and nonlinear in a more detailed and automated manner considering the number of interface nodes.

The linear-elastic part is modeled in Abaqus/Standard and reduced with the Guyan method (Guyan, 1965). Degrees of freedom on the cut between linear and nonlinear areas, as shown in Figure 4, are selected as retained DOF. In Table 1, the numbers of nodes and DOF in the full- and reduced-order models are documented. Under the ideal assumption that the computation time scales down linearly with the decreased number of DOF by MOR, the reduced model should run ideally three times faster than the full model (4.5/1.5=3). Of course, this is an upper limit estimate as the superelement operations and communications require a considerable amount of time.

Table 1. No. of nodes and DOF in full- and reduced-order model

No. of nodes in full-order model	~ 750,000
No. of nodes in reduced-order model	~ 240,000
No. of DOF in full-order model	~ 4.5 million
No. of DOF in reduced-order model	~ 1.5 million

3.1 Results

The deformation of the full- and reduced-order model against the crash barrier is compared in Figure 4. The general deformation patterns match very well. However, more plastic deformation is observed in the crash box of the reduced model as seen in the detailed view of Figure 4. Differences can be explained by the utilization of the simple Guyan reduction (Guyan, 1965) and the selection of the interface nodes. Using Abaqus/Explicit version 6.13-5 with 1 CPU and double=both precision, the reduced model decreased the computational cost by approximately factor 2.1.

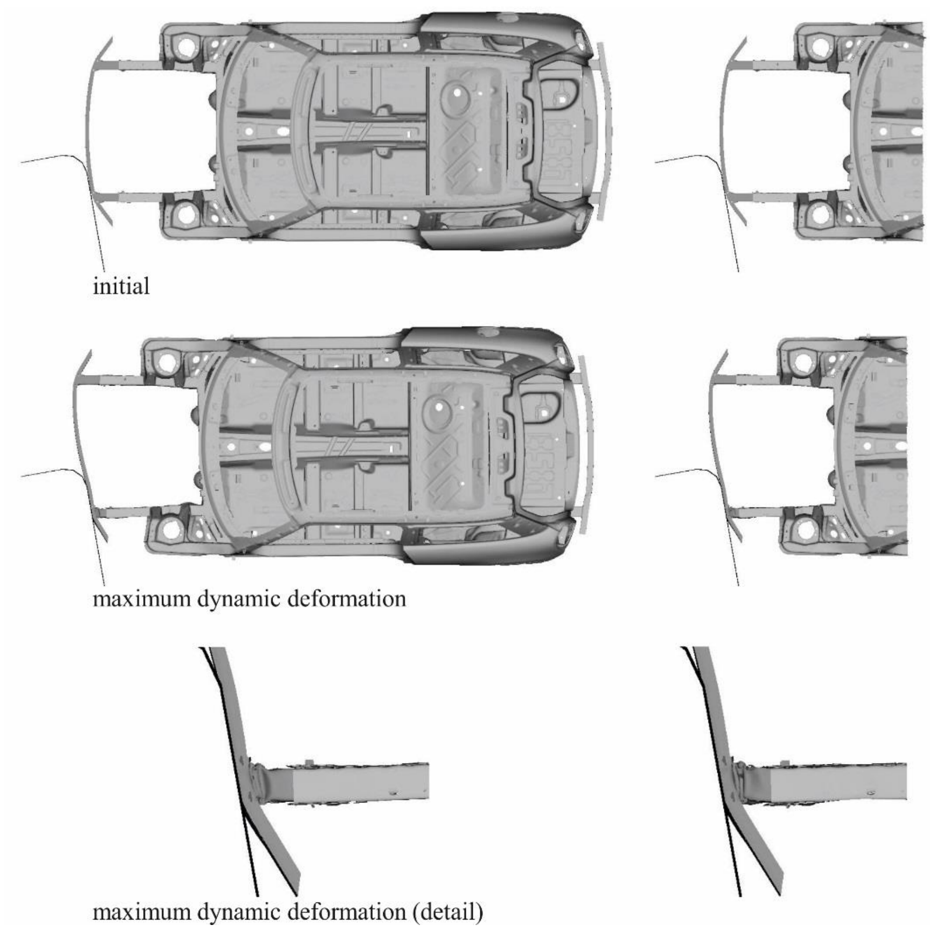

Figure 4. Comparison of full- and reduced-order model with detailed view of the crash system; left: full model; right: reduced model

4. Conclusion

An implementation of MOR in Abaqus/Explicit was realized here to reduce the computational cost of low-speed crash simulations. By taking superelement data outputs from Abaqus/Standard and integrating it through a VUEL user-subroutine in Abaqus/Explicit, a significant decrease in the computation time was obtained. These results show that superelements can improve the simulation efficiency with very small loss in the accuracy and have great potential for low-speed-crash simulation and other comparable impact situations simulated with Abaqus/Explicit.

5. References

1. Abaqus Users Subroutines Reference Guide, Version 6.14-1, Dassault Systèmes Simulia Corp., Providence, RI.
2. Duddeck, F., "Multidisciplinary Optimization of Car Bodies"Structural and Multidisciplinary Optimization, 2008, 35.4: 375-389.
3. Faucher, V., and Combescure, A., "A Time and Space Mortar Method for Coupling Linear Modal Subdomains and Non-linear Subdomains in Explicit Structural Dynamics". Computer Methods in Applied Mechanics and Engineering, 2003, 192.5: 509-533.
4. Flídrová, K., Lenoir, D., Vasseur, N., and Jézéquel, L., "Modelization by Superelements with Contact Management in Explicit Car Crash Simulations"1st Joint Int. Conf. on Multibody System Dynamics IMSD 2010, Lappeenranta, Finland ,2010.
5. Guyan, R. J., "Reduction of Stiffness and Mass Matrices" AIAA journal, 1965, 3.2: 380-380.
6. Maker, B. N., and Benson, D. J., "Modal Methods for Transient Dynamics Analysis in LS-Dyna" 7th Int. LS-Dyna Conf., Detroit, USA, 2002.
7. Research Council for Automobile Repairs, "The Procedure for Conduction a Low Speed Crash 15 km/h Offset Insurance Crash Test to Determine the Damageability and Reparability Features of Motor Vehicles", 1999.