Eulerian-Domain shape optimization for airbag deployment

Jean-Jacques Bernies, Michael Wick, Norbert-Attila Ionas,
Nicolas Bonnin, Dr. Doris Ruckdeschel

Affiliation of Authors

Abstract: *The Coupling-Eulerian-Lagrange (CEL) method used in Abaqus/Explicit allows fluids to interact with solid structure. Interdisciplinary studies in FEM, combined with CFD simulation, have a great performance in the automotive industry - for cockpit and airbag development, especially in the early stage of the airbag deployment. The CEL methods help to get a more realistic airbag unfolding, by injecting inflator gas into the airbag chambers. Compared with the common Abaqus Uniform-Pressure-Method (UPM), the CEL method delivers a more precise airbag deployment. This is necessary for an easier interpretation of load levels, outgoing from the inflator gas pressure on surrounding components, like the headliner or the airbag-cover. To describe the motion of the airbag membrane during the deployment, it is necessary to define an Eulerian-Domain in CEL, in which the inflator gas can flow and interact with the membrane. This domain should be big enough to keep the membrane inside its borders to avoid inflator gas leakage. In the first stage of project work, all domains are designed as big as possible to get the information in which direction the airbag will unfold and to know about the needed domain size. The resolution, depending on the Eulerian element target length, is a main criterion for folded airbags, because the contact calculation between inflator gas and airbag membrane should be precise enough to minimize numerical leakage. To get the best effort in starting a CEL project for airbag development, these two main aspects should be always considered. Using small element size, combined with a large dimensioned Eulerian-Domain, increase the computational time non-linear. This paper will propose a methodology to optimize the Eulerian-Domain size and to save computational time. The goal is to adapt the Eulerian-Domain dimension to the motion of the airbag and delete unnecessary Eulerian elements that are not in contact with the airbag membrane.*

Keywords: *UPM – Uniform Pressure Method; CEL – Coupling Eulerian Lagrange; ODB – Output Database*

1. Introduction

For airbag development, in occupant safety topics, the UPM method in Abaqus/Explicit is normally in use. The deployment of the airbag in the early stage does not require much precision in regard of unfolding behavior, because the dummy is in contact with the airbag when it is fully deployed. In some cases, for example in an “Out of position loadcase” or “I-panel airbag cover opening investigation”, realistic unfolding behavior for at least the first 20 [ms] after injection of the inflator gas is required. That is why a switch to the CEL method is needed, which costs more computational time and an advanced user experience in airbag development.

The common workflow to use the CEL method is to calculate one time completely in UPM, to get the maximum dimension of the unfolded airbag. After getting the maximum X-Y-Z dimension, the normal procedure is to convert the UPM airbag to CEL and to construct the Eulerian-Domain according to the maximum X-Y-Z dimension of the airbag shape. The Eulerian-Domain is modeled like a square box around the airbag and is in its dimension bigger than the full unfolded airbag, to ensure enough space between the Eulerian borders and the airbag membrane.

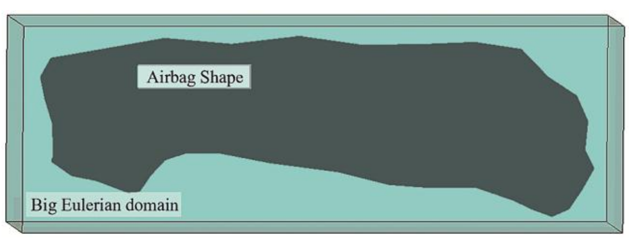

Figure 1. Big Eulerian-Domain surrounding a deployed curtain airbag.

Depending on the airbag type, the Eulerian-Domain square box is not corresponding to the shape of the unfolded airbag. The domain can be huge and the computational time will be dependent on the number of Eulerian elements, if the element target length remains the same. That is why in another simulation loop, the Eulerian-Domain will be reduced, step by step from areas where the airbag will not deploy into. This will help to save computational time but it is an iterative procedure, because the risk is high to cut too much of the domain. The outcome of this will be that the airbag membrane is getting out of the Eulerian borders and some leakage occurs or in some cases the simulation crashes.

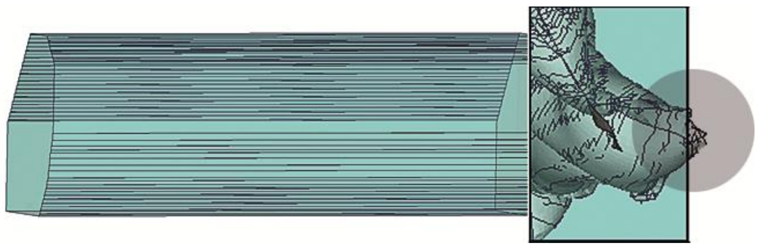

Figure 2. Trimmed Eulerian-Domain where the airbag goes out from the domain.

The results, displayed in the picture above, show a trimmed Eulerian-Domain, done manually with commercial Prepost-Software. As a matter of fact - cutting the domain manually, simulate the deployment every time and checking if the airbag remains inside the Eulerian-Domain – costs a lot of project time and makes it very clear that an automated process is needed to optimize the domain shape. Of course it depends on the project work itself and the goal to be reached, but basically the preparation to gain a well working Eulerian-Domain, that has a good relation to computational time, is an experience needed to get anew for every CEL airbag project.

2. Eulerian-Domain – shape optimization

The purpose of optimizing the Eulerian-Domain is to generate an automated process for shape optimization, so that the cutting process done manually is not needed anymore. The goal is to decrease pre-processing time and computational time with the help of a PYTHON script. The following description will show the automated operation of the so called “Cloud Script” and describe the main idea how the script is working.

The name “Cloud-Script” and the created optimized Eulerian “Cloud-Domain” found their names during the testing phase of the script with a curtain airbag. The optimized Cloud-Domain, surrounding the curtain airbag, had the look similar to a cloud.

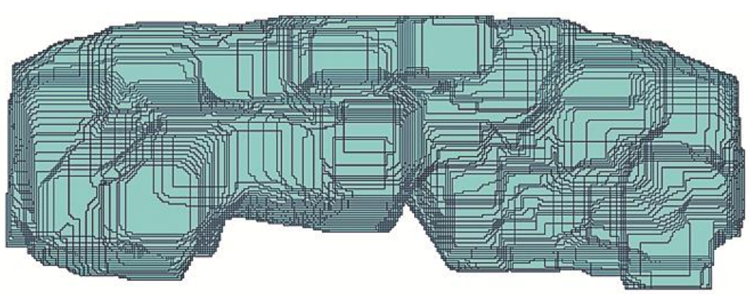

Figure 3. Eulerian Cloud-Domain for a curtain airbag.

To create a Cloud-Domain, the first step is to simulate with a big Eulerian-Domain or a cut one. The results of this simulation as ODB file, is required as an input parameter in the user interface of the script.

The main idea behind the Cloud-Script is to determine, which areas of the big Eulerian-Domain are not used to simulate the injected inflator gas inside the airbag and to delete the unused Eulerian elements. The main search criterion for the script, to detect these unused Eulerian elements in the ODB result file, is the percentage of gas volume fraction of each element ID in the domain.

Before the airbag deployment, the Eulerian-Domain is initially filled with 100% air, so the air volume fraction for each element will be “1”, regardless of the definition of air constituents in Abaqus/Explicit. During the injection of the inflator gas the air inside the Eulerian elements is pushed away from the inflator gas and will flow out of the domain boundaries. The volume fraction inside the airbag membrane will be a mixture of inflator gas and a reduced percentage of air.

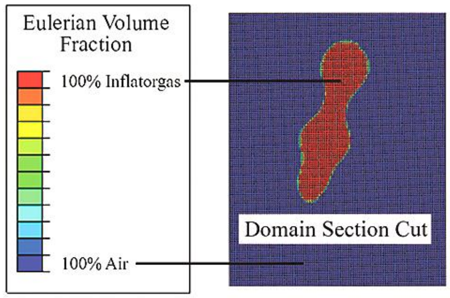

Figure 4. Section-Cut of a curtain airbag showing in red the inflator gas and in blue the areas filled with air.

The Cloud-Script scans for each increment in the ODB result file all Eulerian elements for a certain percentage of inflator gas. If this specified percentage is reached, the element ID will be marked as useful and saved in an element list. That means, the script is using the material instance of each Eulerian element for every increment, to point out the used area of the Eulerian-Domain for airbag deployment.

The script output is the original domain include with the same keywords, but without the unused Eulerian elements, which will never been filled with a certain percentage of inflator gas. After this process the element list, created from the Cloud-Script, will be used to redesign the previously designed big Eulerian-Domain. The raw shape of the created Cloud-Domain is shown in the following picture.

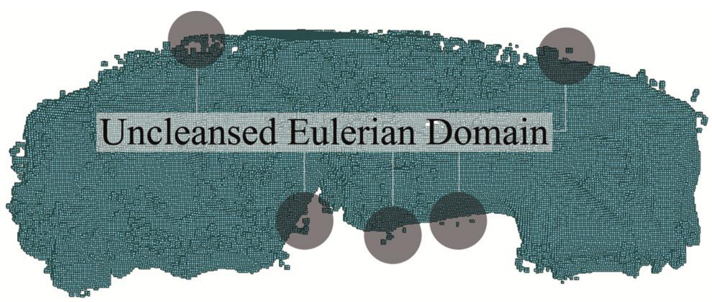

Figure 5. Uncleansed Eulerian Cloud-Domain with not pendant elements.

The raw shape of the Cloud-Domain is unclean in regard to surface texture and independent single Eulerian elements, because it represents exactly the material stream of the inflator gas through the airbag membrane. The impurity is also a result of inevitable numerical leakage through the membrane. It is important to adapt the domain shape to provide a better visual quality and the possibility to create a *SURFACE boundary for in- and outflow of the Eulerian material. Two additional steps are now needed to enhance the outer skin and to add a specified number of Eulerian elements to ensure that the airbag membrane stays inside the Cloud-Domain.

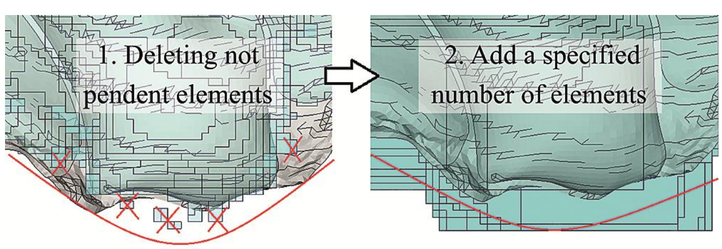

Figure 6. Smoothing process for an Eulerian Cloud-Domain.

A smoothing process is needed to decrease the number of independent single Eulerian elements, so that the Eulerian-Domain surface becomes flatter. The second step is to add a specified numbers of elements, homogenously around the skin. This enhanced process is used to ensure that there is enough space between the outer boundary and the fabric of the airbag to avoid leakage.

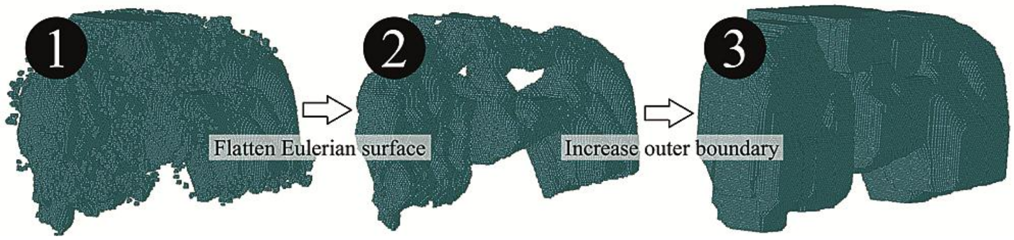

Figure 7. Enhancing process of the Cloud-Script.

The following flow chart is describing the general process of the Cloud-Script during execution in project work:

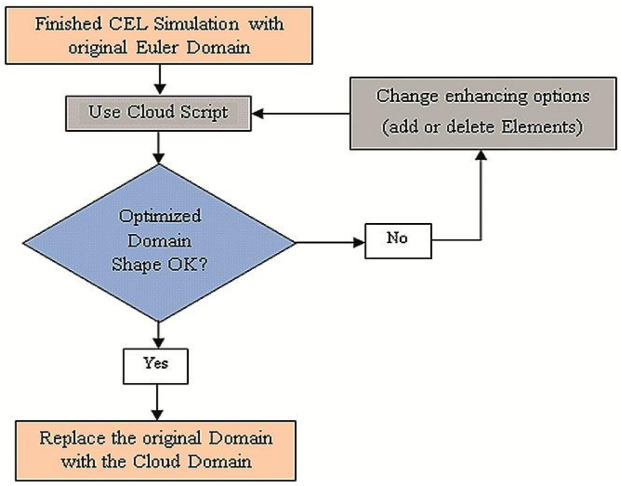

Figure 8. Flow chart for the general process of the Cloud-Script.

Of course the script provides the opportunity for manual intervention on the shape enhancing process during execution. User knowledge in airbag deployment behavior is needed as well as a feeling in which direction the airbag will inflate, when parameters are changed, to provide enough space to ensure the benefit of using the Cloud-Script. The question how big the Cloud-Domain can be is always a compromise between improvements in computational time and robustness, in this regard that the airbag membrane stays inside of the Domain.

3. Working with the Cloud-Domain

For testing purpose, three different airbag types were used with optimized Cloud-Domains. To know how much computational time in general can be saved, a passenger-, a curtain- and a knee-airbag are taken for the comparison between an uncut big Eulerian-Domain, a manual cut Eulerian-Domain and the Cloud-Domain generated by the Cloud-Script.

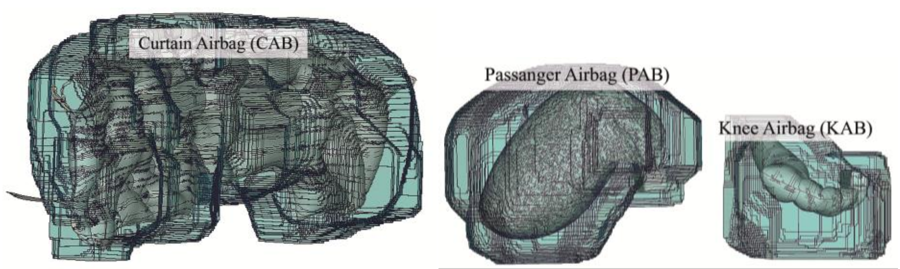

Figure 9. Eulerian Cloud-Domains for different types of airbags.

In LINUX, the script will require approximately 10 - 30 minutes to generate a Cloud-Domain, depending on the size of the outgoing Eulerian-Domain. As previously mentioned the goal of the Cloud-Script is to decrease computational time. In usage, the optimized Cloud-Domain helps to save about 20% - 40% computational time, compared to a non-cut Eulerian-Domain and 10% - 30% to a manual cut domain.

Airbag types	CPU time [h] / Sim. time [ms]	Number of Eulerian elements	% CPU time decrease
CAB Big-Domain	35 [h] / 50[ms]	4736000	Reference
CAB Cut-Domain	31 [h] / 50[ms]	3523140	12
CAB Cloud-Domain	21 [h] / 50[ms]	1742364	40
PAB Big-Domain	89 [h] / 40[ms]	4026280	Reference
PAB Cut-Domain	75 [h] / 40[ms]	3451256	16
PAB Cloud-Domain	64 [h] / 40[ms]	2294077	28
KAB Big-Domain	11 [h] / 50[ms]	1100176	Reference
KAB Cut-Domain	10 [h] / 50[ms]	940709	10
KAB Cloud-Domain	9 [h] / 50[ms]	742487	20

Table 1. Comparison between different types of domains for different types of airbags.

Table 1 presents the results of the three tested airbags. The simulations were started with Abaqus version 6.12-2B in single precision and with 32 CPUS. The table presents also the computational (CPU) time, the simulation duration, the number of Eulerian elements and the saved time in percentage for each domain referenced to the Big-Domain.

As shown in the table above, computational time is coupled to the number of Eulerian elements used in the domains. The results show, that the Eulerian Cloud-Domain has a higher efficiency for airbags with complex geometries. In this case the curtain-airbag is bigger and has a more complex shape than the knee-airbag. Generally using the Cloud-Script to generate the Cloud-Domain can save around 30% computational time, taking the passenger-airbag as a reference in standardized airbag size. For not so complex and smaller airbags, like the knee airbag, the Cloud Script can save 20% computational time, because the script can not delete so many elements from the original Eulerian-Domain. To show that the results of the deployment with the Cloud-Domain remain the same, the comparison between the pressures inside the airbags is plotted in the Figure below.

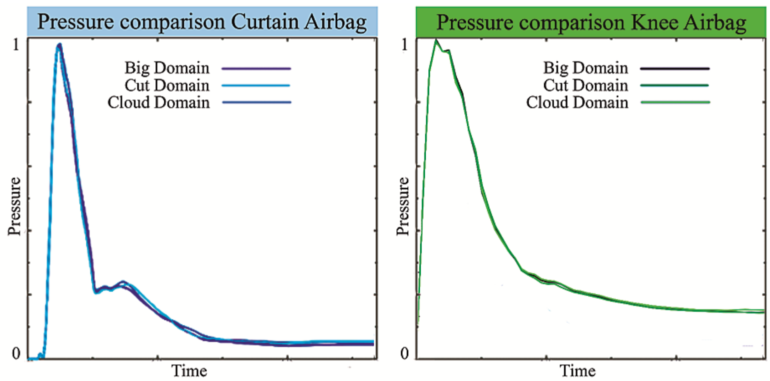

Figure 10. Pressure comparison for the above mentioned airbags.

When using the Cloud-Domain, the compromise is that the enhanced options for the Cloud-Domain are set in proportion to the effort of saving computational time and robust airbag unfolding. It would be beneficial to extend the size of the Cloud-Domain in the way, that there is enough space to provide changes in parameters for project work, but also to save enough computational time compared to a manual cut Eulerian-Domain. Considering this requirement it is better to save 20% computational time and to have a robust model, than saving 30% computational time but take the risk that the airbag membrane is exceeding the limits of the Eulerian border.

The following flow chart will explain a standardized CEL airbag development workflow and show at what point the Cloud-Script is in use:

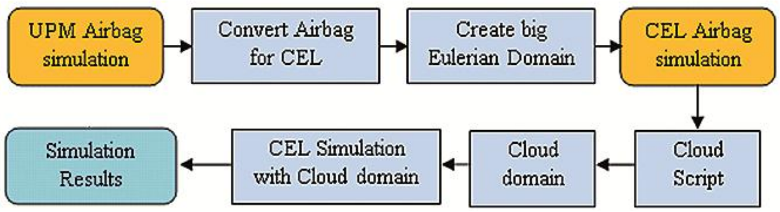

Figure 11. Flowchart of a CEL airbag simulation including the conversion step from UPM to CEL.

4. Conclusion and forecast

As shown in the previous chapter, the usage of the Cloud-Script can save a lot of computational time, depending on the airbag type and enhancing options you want to use for the Cloud-Domain. Of course the benefit is depending on the size of the Cloud-Domain and is always restricted by the requirements of the project.

The Cloud-Script helps to broaden the acceptance of the CEL method in project work. An optimized Eulerian-Domain is necessary to improve the perception of the CEL method, that it is no longer seen as a necessary time-consumer, especially in airbag development, but rather as an opportunity to improve the results in regard of realistic unfolding behavior.

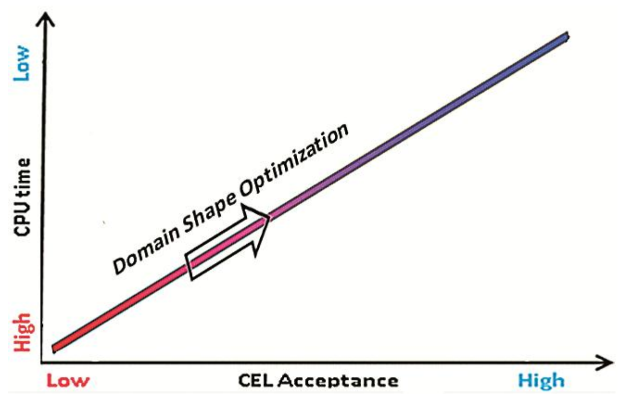

Figure 12. CPU time over CEL Acceptance by using domain shape optimization.

To get a better feeling of how efficient the Cloud-Script is working in regard to computational time-saving, PuZ Engineering GmbH is furthermore testing in addition to the already tested airbags, a side-airbag and a pedestrian-airbag. The assumption here is: the bigger and geometrically complicated the airbag is, the more project time and computational time can be reduced by using the Cloud-Script.

The aim of the actual work on the script is to get more possibilities in enhanced options for the domain shape optimization and to be able to handle the user interface of the Cloud-Script better.

PuZ Engineering GmbH is working on a method to check if the airbag membrane is touching or exceeding the borders of the Cloud-Domain after calculation. This method will be applicable for every CEL airbag simulation, independent of the Eulerian-Domain shape type and will help the user to shorten the post-processing time.

5. Reference

1. Abaqus Users Manual, Version 6.12, Dassault Systémes Simulia Corp., Providence, RI.