Identifying similarities and exceptions in deformations and mesh functions.

Comparing many simulation results automatically

J. Garcke[1], R. Iza-Teran[1], M. Pathare[1], P. Schwanitz[2], D. Steffes-lai[1]

1 Fraunhofer SCAI, Schloss Birlinghoven 1, 53757 Sankt Augustin

2 Dr. Ing. h.c. F. Porsche Aktiengesellschaft, Entwicklung, Porschestraße, D-71287 Weissach

Abstract

The growing complexity in the design space of finite element models and increasing requirements and regulations for crash safety together lead to large development trees in the CAE crashworthiness development process. That is, during the virtual product development numerous design changes are applied and analysed until the final model satisfies given design criteria. The biggest challenge is how to compare several simulation results to detect global events, examples are unexpected deformation behavior or unusual distribution of mesh functions such as plastic strain. We observe a lack of simple to use procedures that analyse and automatically categorize design measures together with their impact on simulation results. To address these challenges, we have developed a workflow to easily analyse the impact of design variations as an important step in the overall simulation data analysis. Our software, called SimExplore, enables an overview over many simulations pointing out their similarities and exceptions in deformations and mesh functions automatically.

The capabilities of SimExplore are demonstrated on data from a robustness analysis of a frontal vehicle crash, where the wall thicknesses of several selected components have been varied. Our automated workflow detects different deformation modes and identifies distinctly different behavior.

1. Introduction

Numerical simulations play an important role in the CAE product development process. Especially in the CAE crashworthiness analysis process, the growing complexity in the design space of finite element models (FEM) an increasing requirements and regulations for crash safety lead to large so-called development trees. These reflect the many undertaken design measures and corresponding simulation runs, which are performed until given design criteria

are satisfied. Following a path in the obtained development tree, the differences from one simulation to its predecessor, or root of the tree, consist of one or several design variations on the one hand, while on the other hand the simulation results can show numerous changes in the crash behaviour.
Decisions about the product are based on engineering judgement as well as economic constraints. The evaluation of specific designs requires fast evaluation and comparison of multiple variants of a model based on quantities derived from simulations such as the head injury criterion (HIC) index, intrusion values at points in the structure, energy absorption and so on. Multiple scalar values or curves have to be evaluated and compared for each variant.

Such complex tasks suggest the use of artificial intelligence (AI) methods to manage the evaluation complexity. Specifically, machine learning has gotten impressive results for image recognition as well as for learning tasks using data. Methods that can learn from existing data and then make predictions for related, but new situations are indeed very attractive for product development. An example is to make predictions for design parameter combinations that have not been simulated so far, i.e. estimate the effect of design changes on quantities of interest. To identify the main trends in a bundle of simulations, computed by varying several parameters, is another possible data analysis application. But, existent machine learning methods are not easily applicable to the analysis of CAE simulations. On the one hand because they typically require large amounts of training data, which are not available since simulations are expensive to compute, and on the other hand because the data is high-dimensional, where the dimensionality is given by the size of the finite element meshes. We have developed approaches that can overcome these limitations and are adapted to CAE simulations, which has the potential to significantly improve phases of the virtual product development.

With the software tool SimExplore we have implemented a new and easily applicable analysis approach, which allows the study of the impact of design variations as an important step in the overall simulation data analysis workflow. Our workflow applies novel machine learning approaches to provide an overview over many simulations, in particular automatically pointing out similarities and exceptions, respectively, in deformations and mesh functions, as well as their propagation over time. In detail, we investigate suitable concepts of similarity to arrange simulation results in an overview diagram reflecting the different deformation modes. A dimensionality reduction approach based on the Laplace-Beltrami operator is used, which can be thought of as a „Fourier“-decomposition for geometries. This novel approach enables a clustering of many simulations based on their crash-behavior using these geometric "Fourier"-modes.

With our proposed methodology, the CAE engineer is provided with a browsable overview of all the simulations, analysed in terms of clusters as well as outliers, due to deformations or other mesh functions. Here, the simulations within a cluster show similar deformation behavior, and the automated workflow identifies representative simulations for the different clusters of deformation modes. Moreover, outliers are detected, which show different, and most often unwanted, behaviour. All these deformation modes and outliers are identified, documented and visualized using a structured data representation, which allows a smooth integration in any simulation process data management tool for documentation and further analysis. Thus, our SimExplore approach marks a breakthrough towards an automatic global event detection for car crash simulations within the overall crashworthiness data analysis process.

The capabilities of our analysis workflow are demonstrated on an industrial use case. We investigate simulation data from a robustness analysis of a frontal vehicle crash load case where the wall thicknesses of several selected components have been varied. The simulation results are analysed to search for similarities as well as outliers in the deformation behaviour.

2. Workflow for Event Detection

The workflow for event detection is made of two main phases. First, a longer running off-line batch-process, which can run at the simulation cluster where the raw data is available, and a second interactive exploring phase, which can run on a client computer and where in particular the raw data does not need to be available. A schematic representation of the first phase is shown in Figure 1.

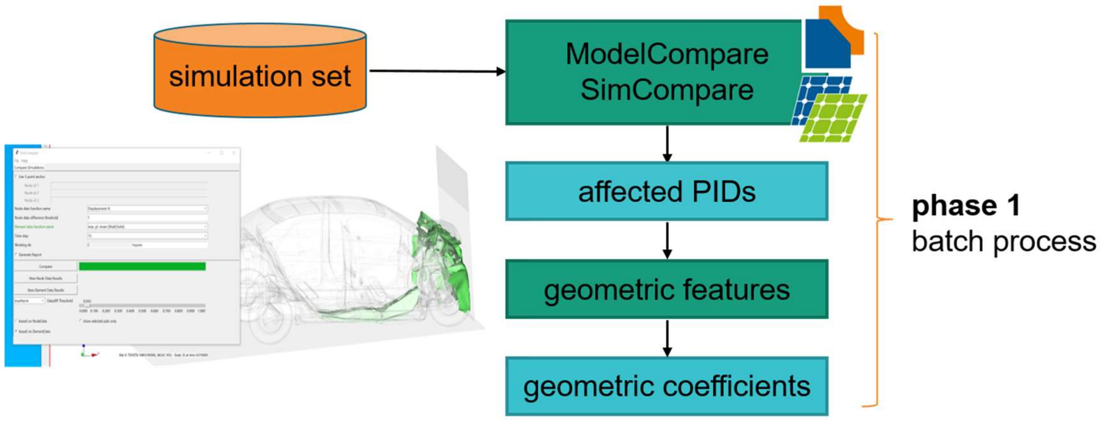

Figure 1 Schematic representation of phase 1 of the workflow

The results of the first phase are a list of most affected PIDs and the so called shape features necessary for the second phase. A schematic representation of this second explorative phase is shown in Figure 2.

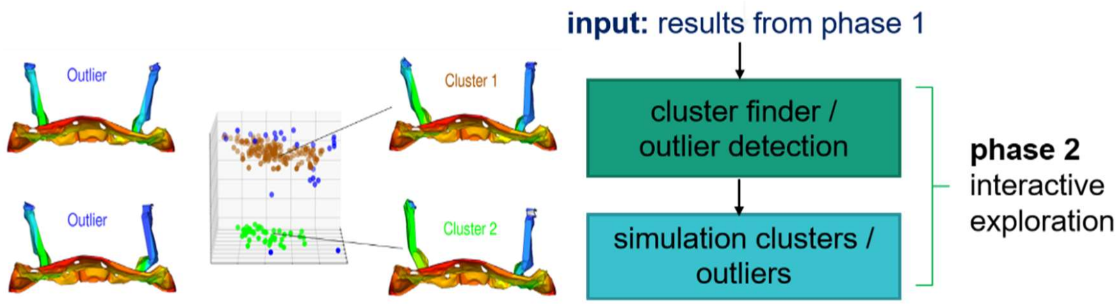

Figure 2 Schematic representation of phase 2 of the workflow.

In the following we describe some of the components of the workflow in more in detail.

Design Measures

The first step of the workflow consists of a comparison of the different input configurations for two numerical simulations using ModelCompare [2]. Thereby an identification of the design measures performed through model adaptation between the two simulations is achieved. This is based on the geometric parts and their discretization (mesh), and thus independent of the parts meta data, such as an identifier or a name. Examples of design measures (groups) are changes in geometry, multi-parts, that is several parts are merged into one part, changes in spot welds and rigid body elements, material ID and thickness changes. All the identified design measures are stored in a structured format, namely in JSON files. This allows an easy storage and integration, in particular for a later investigation together with the detected events.

Local Event Detection

The second step of the overall approach deals with the comparison of two simulation results based on the design measures applied by SimCompare [4], using the full output data, namely displacement and functions on meshes. The approach analyses functions on meshes rather than scalar data or sensor / curve data. More precisely, the impact of design changes can be analysed in terms of data functions either on the nodes (such as displacements or nodal mass) or elements (such as plastic strains, stresses, or failed elements) depending on the analysis objective. Thereby, detailed insight in local influences can be evaluated with respect to a certain design change. The comparison results, that is the local events found, are stored partwise in structured JSON files supporting further post processing. In particular, for each part

selected, the part-ID, part-name, metric value, as well as minimum and maximum deviation in the part is stored.
The determination of the most affected parts is based on two-sided comparisons using SimCompare, where each simulation is compared to a reference simulation, typically the root of the variant tree. Using all these comparison results, the most affected parts over all simulations are determined depending on a threshold and a suitable output quantity as selected by the user. This identifies those parts where large changes can be seen over all the simulations being investigated.

Global Event Detection

In the next step a series of geometry-driven features (so-called geometric Fourier spectra [3]) representing the displacements or mesh functions are computed for all time steps for selected parts. In particular, the selection can be based on the SimCompare-based local event detection, as just described.
Note that a common practice in CAE-based engineering design is to choose a specific set of PIDs (car parts) that are considered to be important by the user. Furthermore, the list of parts is typically organized in groups such as frontal beams, firewall, pillars, support frames, seats supports, fuel tank compartment and so on. This user-driven group or functionality-based selection can be combined with the local event detection. The parts selected for the global event detection in such a combined case either fulfil both criteria, i.e. those PIDs of the B-pillar that are identified as most affected, or at least one criteria, i.e. all PIDs of the B-pillar and all PIDs that are identified as most affected.

From the obtained geometric features for the selected parts an optimization algorithm will select three coefficients that shows the best overall separation of the simulation outcomes into clusters, which is done for each time step and for all parts. In addition, at each time step a clustering score is computed, which allows a selection of those time steps that show increased separation from the point of view of clustering, which we call peaks of the clustering score.

The analysis results are stored for further analysis and reproducibility. For example, we store information about the found clusters or outliers in JSON files, i.e. a simulation is labelled as part of a cluster or as an outlier per time step and mesh function. The data analysis information saved during the global event detection is relatively compact, so that it can be transferred to a client desktop for further data exploration or can be explored via a web-based visualization. Finally, for each of the found peaks, several images can be saved with different views of the simulation results for the PID(s) involved for later visual exploration, but note that these automatically generated images can take significant disk space and compute time.

Clustering and outlier identification finalizes the first phase of the SimExplore workflow.

Interactive Exploration

Using the information computed in the previous steps, an interactive part allows the user to seamlessly explore and analyze different parts or part combinations. The visual representation of the simulations and parts is based on a 3D representation using the geometric features. Each point in this representation represents a simulation and the overall visualization provides an intuitive overview of the similarities and exceptions in the simulations with respect to the chosen functions. Such an interactive visualization allows an easy-to-use exploration of the simulation results. In particular, clustering and outlier detection can be easily visualized by this representation. Currently for the interactive view a web based app is used, but an integration into other data visualization software is feasible via plugin-architectures.

3. Experimental Results

As an example to test the workflow we consider a DOE (design of experiment) of a frontal vehicle crash. A total of 38 random thickness changes for the parts as shown in Figure 3 have been used. The study has been done as part of a PhD study [1]. From there, a total of 300 frontal crash LS-Dyna simulations have been considered in the following.

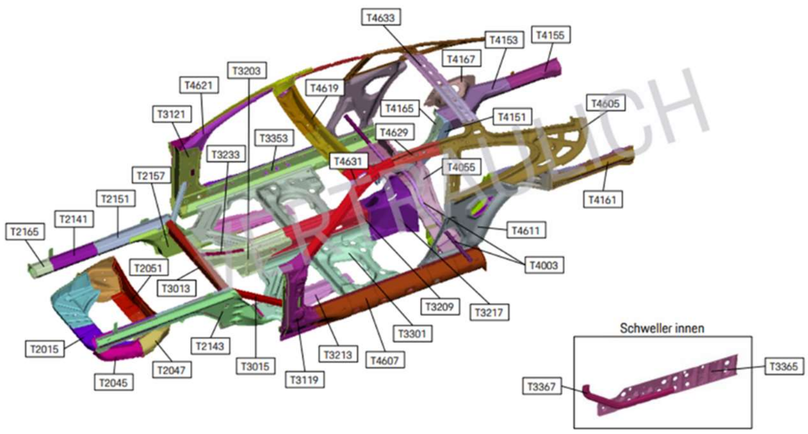

Figure 3 Structural members with thickness changes as considered in this study.

Results of the Local Event Detection

Simulation results are compared in pairs (299 simulations vs. the chosen reference) using SimCompare to detect the local differences in the mesh functions. An exemplary comparison with SimCompare between two simulations is shown in Figure 4.

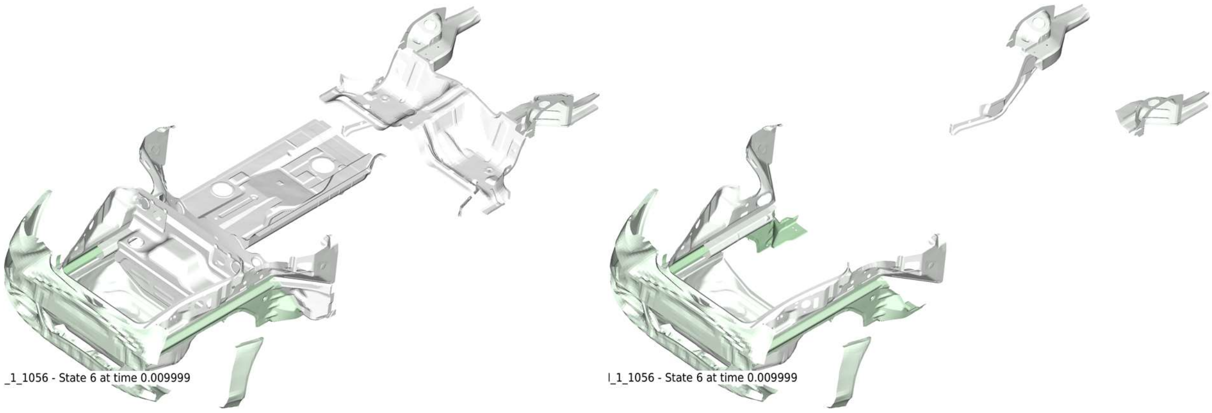

Figure 4 Example of a SimCompare result for two simulation, at time step 6 early in the crash. Shown are the parts with a difference in the plastic strain between the simulations above a given threshold, 33 parts are on the left, 23 parts on the right. The intensity of green indicates the magnitude of the difference. As one would expect from the design difference of the two simulations, the structural beams show the largest difference.

For further analysis of individual results, an investigation of the difference between two simulations can be performed. For the two identified parts with the largest difference in behavior for the two chosen simulations we show in Figure 5 an example of the difference in plastic strain on these parts.

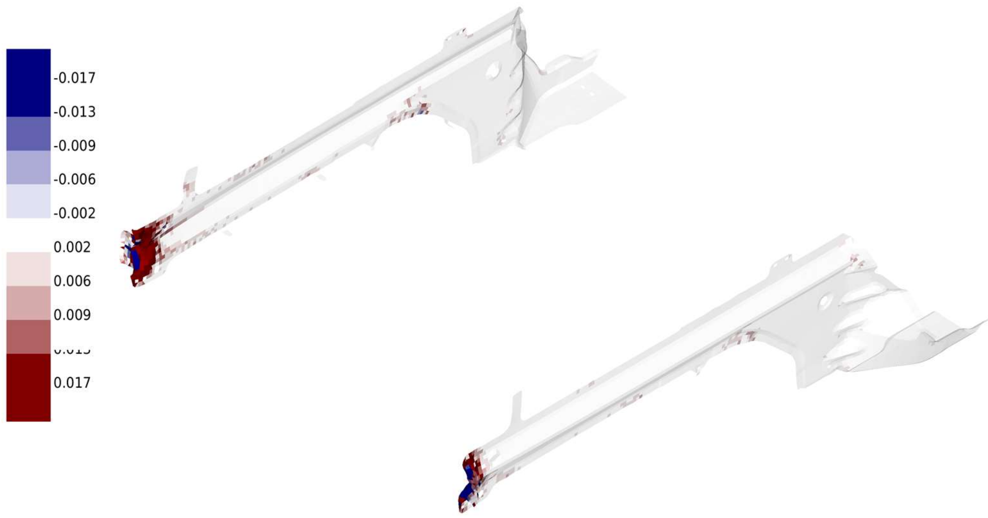

Figure 5 The difference in plastic strain between the structural beams at time step 6.

The most affected parts are then selected using a threshold for a user-chosen mesh function and suitable metric over all the pairwise comparison. An exemplary outcome of a filtering based on local event detection is shown in Figure 6.

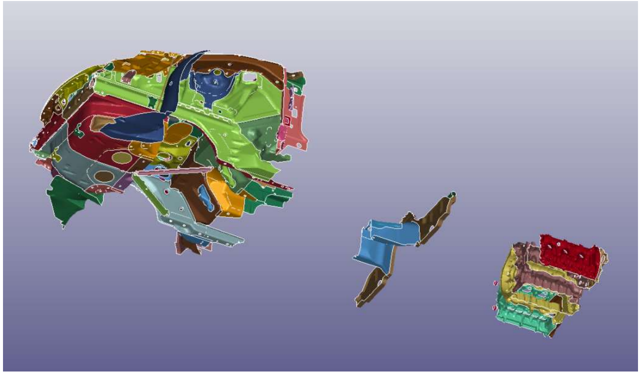

Figure 6 Visualization of the parts selected by the local event detection.

From these PIDs we select those from the structural components in the front of the car. In other words, for this study we use those parts that fulfil both criteria, most affected according to the above local event detection and the functionality-based criteria. The selected parts for the following global event detection are shown in Figure 7.

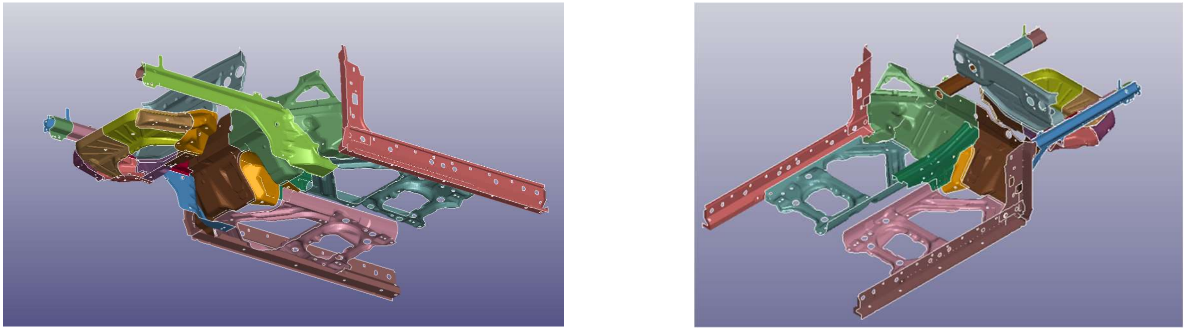

Figure 7 Parts (PIDs) of the car selected for the analysis of Global events.

Results of the Global Event Detection

Now, the geometrical features are computed for each time step for each PID (or group of PIDs) selected before. The cluster finder module returns scores for each time step (see Figure 8), from which the peaks can be identified, i.e. those clusters that are more separated.

In the following we use two chosen PIDs for an illustration of the results that can be obtained.

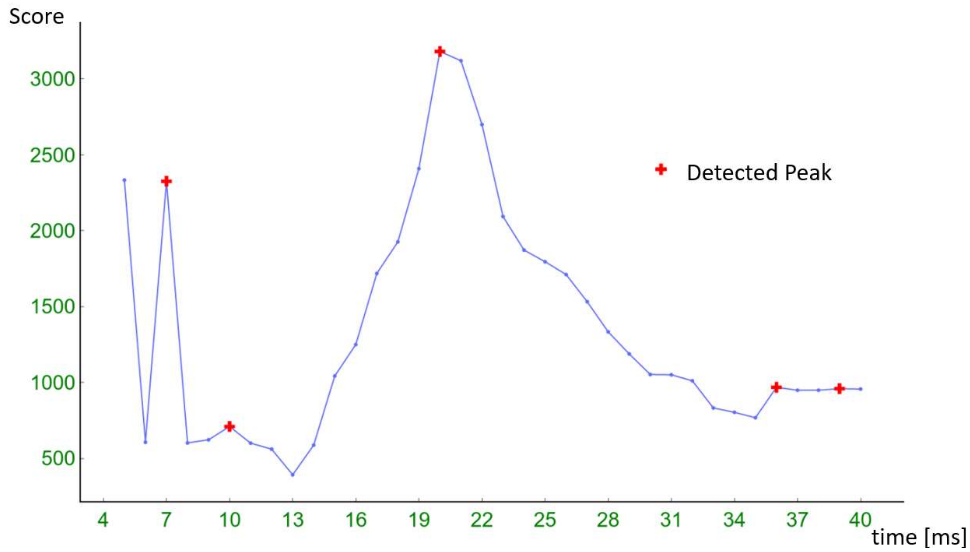

Figure 8 Clustering score for a selected PID. The x axis are the time steps considered, peaks are indicated by red crosses in the plot. For the peak at time t=7 a high peak indicates a separation in the clusters for the geometric Fourier coefficients (see Figure 10 for the corresponding 3D plot of the coefficients).

For a chosen time step (a peak) a 3D point cloud is determined, which is a 3D representation based on the geometric Fourier coefficients. Each point corresponds to one deformation of the PID considered. So we have in total 300 points corresponding to the 300 simulations of the DOE analysis. The color of the points represents the clustering obtained. In the case of Figure 9 using the firewall we have 2 clusters marked with orange and blue and in between we can see green points that corresponds to points not in any of the clusters, while additional green points are further away. By clicking on each point, the image of the corresponding simulation for a chosen view is shown in the web browser. For illustration, we superpose images along the simulation point cloud. From Figure 9, it can be seen that as we go from cluster 1 (orange) to cluster 2 (blue), the plastic strain shows different severity between the two clusters, but similar behavior per cluster. There are some parts interspersed between the clusters, while some parts are outliers further away. In other words, the point cloud as intended represents similarity between simulations in a global overview.

As another example we show the structural beams in Figure 10. Here it can be seen that as we go from cluster 1 (yellow) to cluster 2 (orange), the plastic strain shows a gradual change along the points.

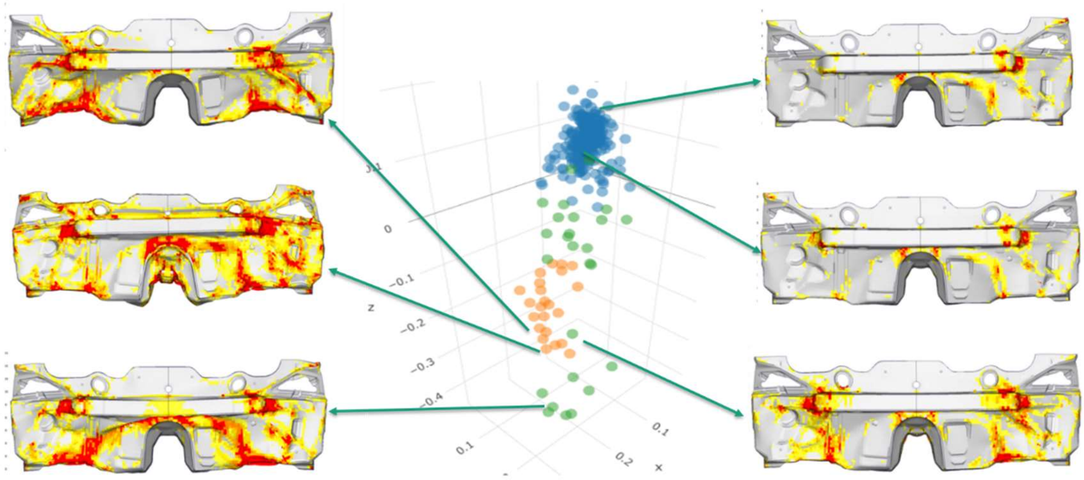

Figure 9 Example of an explorative analysis of the firewall, where the plastic strain at time step 27 is investigated. Note that the shown static two-dimensional picture of the three dimensional interactive visualization is limited. For example, the outlier on the bottom right appears close to the orange cluster, while it is actually further away, which one be seen by interactively rotating the three-dimensional visualization.

Sensitivity Analysis in Geometric Fourier Space

The DOE study [1] was focused on the sensitivity of thickness changes in view of the crash performance of the car. An investigation was performed on the basis of several post-processing quantities like intrusion values at some critical points in the structure and cross sectional forces of the main structural members. Statistical measures of sensitivity such as Sobol indexes based on the evaluated post-processing quantities showed that from the 38 thickness only a few are considered sensitive.

We now investigate if and how instead of taking the post-processing quantities, a preliminary analysis of the sensitivity can be done based on the 3D geometric Fourier representation. First results show that the same parts detected by [1] can also be determined using our approach. As an example, we present a visual analysis for one of the sensitive parts.

As seen in Figure 10, the points are not randomly colored, but they show a sequential change along the point cloud reflecting not only the sensitivity but also how the thickness variation affects the deformation. Furthermore, on the left one sees the changes in behavior along the clusters, while points not in the clusters are also detected by the event detection workflow. As a consistency check, we considered the change of thickness value of other non-sensitive parts and repeated the same evaluation. We could not identify the structured color change obtained for the sensitive parts, but rather observe a random distribution of the thickness colors.

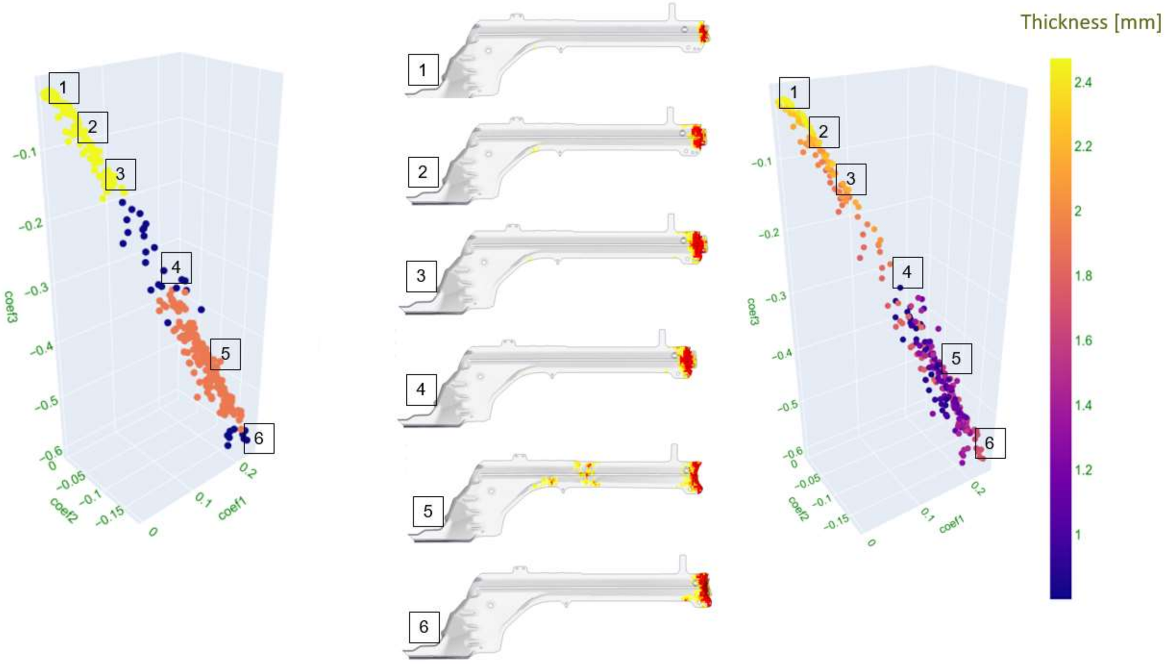

Figure 10 On the left is a point cloud of geometric Fourier coefficients for the selected part. Cluster 1 is in yellow, 2 in orange and points not assigned to a cluster are blue. For chosen simulations we show the corresponding plastic strain in the middle. On the right is the same point cloud, but now the coloring reflects the thickness value of one sensitive part.

Conclusions

This work described a workflow able to explore many simulations. One obtains in an automatic way clusters of similar deformations and outliers having distinctly different behaviour. The detection of events has been done locally with respect to pairs of simulations and globally with respect to all simulations. Low dimensional 3D point clouds with geometric Fourier coefficients capturing similarity were used for clustering and outlier detection. Finally, preliminary investigations on sensitivity analysis in geometric Fourier space show promising results that warrant further studies.

[1] J. Büttner, Effiziente Lösungsansätze zur Reduktion des numerischen Ressourcen-bedarfs für den operativen Einsatz der Multidisziplinären Optimierung von Fahrzeugstrukturen, Dissertation, Bergische Universität Wuppertal, 2022.

[2] J. Garcke, M. Pathare, and N. Prabakaran. ModelCompare. In Scientific Computing and Algorithms in Industrial Simulations: Projects and Products of Fraunhofer SCAI, pages 199–205. Springer International Publishing, Cham, 2017

[3] R. Iza-Teran and J. Garcke. A geometrical method for low-dimensional representations of simulations. SIAM/ASA Journal on Uncertainty Quantification, 7(2):472–496, 2019.

[4] D. Steffes-lai, M. Pathare, and J. Garcke. Towards a Framework for Automatic Event Detection for Car Crash Simulations. In NAFEMS World Congress 2021. 2021