NordDesign 2018
August 14 – 17, 2018
Linköping, Sweden

Methodology for plausibility checking of structural mechanics simulations using Deep Learning on existing simulation data

Spruegel T.C.[1], Rothfelder R.[1], Bickel S.[1] Grauf A.[1], Sauer C.[1], Schleich B.[1], Wartzack S.[1]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Konstruktionstechnik
spruegel@mfk.fau.de

Abstract (300-500 words)

In modern product development, the use of sophisticated simulation tools for assessing the effects of design changes on the intended product behavior is essential. However, setting up valid simulations requires expert knowledge, acquired skills, and sufficient expertise. Design engineers, who perform finite element analysis (FEA) infrequently, must be assisted and their FEA results need to be checked for plausibility. An automatic plausibility check for finite element (FE) simulations in linear structural mechanics can identify non-plausible simulations and warn the user to interpret the results cautiously or ask for expert help. In this context, currently available tools can only compare very similar simulations.

However, as the amount of available simulation data in the industry increases more and more, a data-driven simulation check is an obvious next step. Nevertheless, the question arises how simulation data of very different parts and simulations can be transferred to a single software tool, how this tool can learn the relevant rules behind plausible simulations, and how it can be applied to new simulations. In this context, it is especially important to train a metamodel that is able to generalize the rules so that it can later on be applied to unknown simulations.

This paper presents an approach to transfer different FE meshes, corresponding FE results and boundary conditions to an individual matrix of fixed size for very different structural mechanic FE simulation. The novel approach uses spherical detector surfaces to project three-dimensional information on its surface. It allows generating the so-called "DNA of an FE simulation"; classification algorithms i.e. Support Vector Machines or Deep Learning Neural Networks such as Convolutional Neural Networks (CNN) can then classify this information. The whole methodology reduces the dimension of a 3D finite element simulation to a 2D matrix of numeric values. The matrix contains all the relevant information for the classification in "plausible" or "non-plausible". An implausible simulation contains errors, which would be quickly identified by an experienced simulation engineer, whereas a plausible simulation does not contain such errors. As less experienced simulation users in design departments are not trained to find such errors in their simulation setup, they cannot detect them and take adequate countermeasures.

In the paper, every single step of the novel methodology for plausibility checking of structural mechanics simulations will be illustrated and explained in detail for simplified parts and corresponding simulations.

Keywords: Plausibility Check, Convolutional Neural Network, Deep Learning, Data-driven approach, design automation, visualization

1 Introduction

In current product development not only the amount of data is continuously increasing, but also the density of data (Sauer, Küstner, Schleich & Wartzack, 2017). In some companies of the automotive supply industry, several thousand FE simulations are done each day for different components. If simulations with non-ideal geometries (Schleich, Anwer, Mathieu & Wartzack, 2015) are performed, the amount of data increases even more. The simulated parts are of course not completely different from each other, but the simulated parts differ in geometry and boundary conditions and the simulations are performed by different engineers. It must be kept in mind that not only experienced simulation engineers with multiple years of experience perform these simulations, but also less experienced users (Kestel, Schneyer & Wartzack, 2016). Consequently, there is a need to have automatic plausibility checks to find obviously wrong simulation setups automatically. Machine Learning algorithms, such as Convolutional Neural Networks, are very well suited for this purpose. For similar simulations, algorithms can be used to compare the results from new simulations with the results from previous simulations, as long as the simulations are somehow similar. Therefore, the research question arises: How can heterogeneous FE simulations be used as an input for Machine Learning algorithms? In this contribution a methodology is presented how linear structural mechanic FE simulations can be transformed to a matrix of fixed size (containing the relevant simulation information to classify simulations as “plausible” or “non-plausible”). This matrix can be named as the “DNA of an FE simulation”. In section 2 the current state of the art is presented, followed by the description of the methodology in section 3. Finally a conclusion and outlook is presented in section 4.

2 State of the art

In modern product development the influence of data is getting more and more important and the amount of data is constantly evolving. Especially in the field of simulation the amount of available data from previous simulations is growing very fast. At first the terms Data Mining, Data Driven Design, Artificial Neural Networks, Deep Learning and plausibility are presented briefly.

2.1 Data Mining

Generally the notion Data Mining, also referred to as knowledge discovery in databases (KDD) (Quin & Tang, 1995; Chen, Han & Yu, 1996), includes a broad field of techniques and methods used for extracting (mining) previously unknown and potentially useful information, such as knowledge rules, constraints, regularities, from large amounts of data (Quin & Tang, 1995; Chen et al., 1996; Runkler, 2010). This field of research consists of, amongst others, methods for Classification, Regression-Analysis, Correlation-Analysis and Clustering (Runkler, 2010). The discovered knowledge can be applied to information management, query processing, decision making, process control, and many other applications (Chen et al., 1996). There are a number of different approaches contemplating Data Mining. According to the CRISP-DM (CRISP-Data Mining) Methodology, Data Mining can be divided into five main parts: Data Understanding, Data Preparation, Modelling, Evaluation and Deployment (Chapman, 1999).

Furthermore, Zhou (2002) regards Data Mining from three different perspectives (and the corresponding emphases): databases (efficiency), machine learning (effectiveness) and statistics (validity) in order to realize adequate mining methods.

2.2 Data-driven (Engineering) Design

Data-driven Design in the context of engineering is utilized for designing systems or processes, as fault detection and isolation, fault-tolerant control, optimization, process monitoring (Ding, 2012; Yin, Ding, Xie & Luo, 2014) and adaptive residual generator design (Ding, Yin, Zhang, Ding & Naik, 2009) by using information extracted from a large amount of data. Compared with the widespread model-based process monitoring design framework, the data-driven design framework is still in its early development phase. The most challenging topic, contemplating data-driven design, is dealing with incomplete data (Yin et al., 2014).

2.3 Artificial Neural Networks

Artificial Neuronal Networks (ANN), or just Neural Networks (NN), are a specific form of metamodel. They are inspired by biological neural systems, as bioelectrical networks in the brain of mammals, formed by neurons and their synapses, which are capable to perform intricate and complex computations without using explicit quantitative operations. In an ANN, nodes, also called neurons, are connected together to create a network. Contemplating their generation and function, ANN can be assigned to the field of machine learning (Quin & Tang, 1995; Kruse, Borgelt, Braune, Klawonn, Moewes & Steinbrecher, 2015). ANN are able to perform powerful pattern classification and recognition (Zhang, Patuwo & Hu, 1997) and are utilized in a variety of different research fields in business, industry and science for very different tasks. Such as forecasting (Zhang et al., 1997), cancer management (Dayhoff & DeLeo, 2001) and real-time process control for small utilities (Zhang, Shariff, Smith, Cudrak & Stanley, 2007).

2.4 Deep Learning

The term Deep Learning describes techniques to create computational models such as Artificial Neural Networks that consist of multiple operating layers to handle and process data using various levels of abstraction. The models are capable to discover complex structures within large data sets utilizing backpropagation algorithm (Bengio, 2009; LeCun, Bengio & Hinton, 2015; Deng, 2012). Deep Learning is located in the field of unsupervised learning, where no teacher is used. To distinguish Deep Learning Networks from conventional Neural Networks the concept of Credit assignment paths is used (Schmidhuber, 2014). For classification with Convolutional Neural Networks (CNN) supervised learning algorithms are used for training. Unsupervised learning algorithms for instance are used for clustering.

2.5 Plausibility of simulations and simulation results

Plausible FE simulations are apparently, likely valid (Spruegel, Schröppel & Wartzack, 2017). Consequently, they contain no obvious errors – which would be recognized easily by experienced simulation engineers. Plausibility checks are used in various fields (publications often in german):

- Plausibility checks to analyze electrical breakdown mechanisms in syntactic foams (Tröger, 2009)
- Plausibility checks for the measurement of uncertainties in soil analysis (Nestler, 2007).
- Plausibility checks for vehicle sensor data (Tischler, 2013).

- Plausibility checks in dynamic simulations for a cylindrical roller bearing model in wind turbines based on the known behaviors of bearings (Qian, 2013).
- Plausibility checks of integrated behavior models for consistency and critical signs (Ermel, Gall, Lambers & Taentzer, 2011).
- Automated pre-plausibility checks of input data for Intra-Logistic-Simulations in digital factories, entered by users (Müller-Sommer & Straßburger, 2010).

However, it lacks a systematic approach to the plausibility check of FE simulations using machine-learning methods. As a response to this, a method for the transfer of FE data to such data-mining techniques is presented. A plausible simulation does not contain obvious errors, that a simulation engineer would find quickly (i. e. mix-up of units, missing boundary conditions, above the ordinary forces or moments, too coarse meshes, etc.).

3 Transfer of FE data to Artificial Neural Networks

Artificial Neural Networks (ANN) are used in a vast variety of applications. Both regression and classification are common tasks. The simulation results from similar simulations can be transferred to neural networks very easily (i.e. stresses, deformations or safety factors at specific nodes). But currently there is no methodology available to transfer heterogeneous FE simulation data to neural networks. The presented methodology in this paper can transfer all relevant FE data (i. e. mesh, boundary conditions and results) to machine learning algorithms - such as ANN. The simulations can be totally different in case of geometry, boundary conditions or mesh. Preliminary work to this publication can be found in the contributions of Spruegel, Schröppel & Wartzack, (2017) and Spruegel, Kestel & Wartzack, (2016); in this publication the whole methodology is presented – including the visualization of sub-steps.

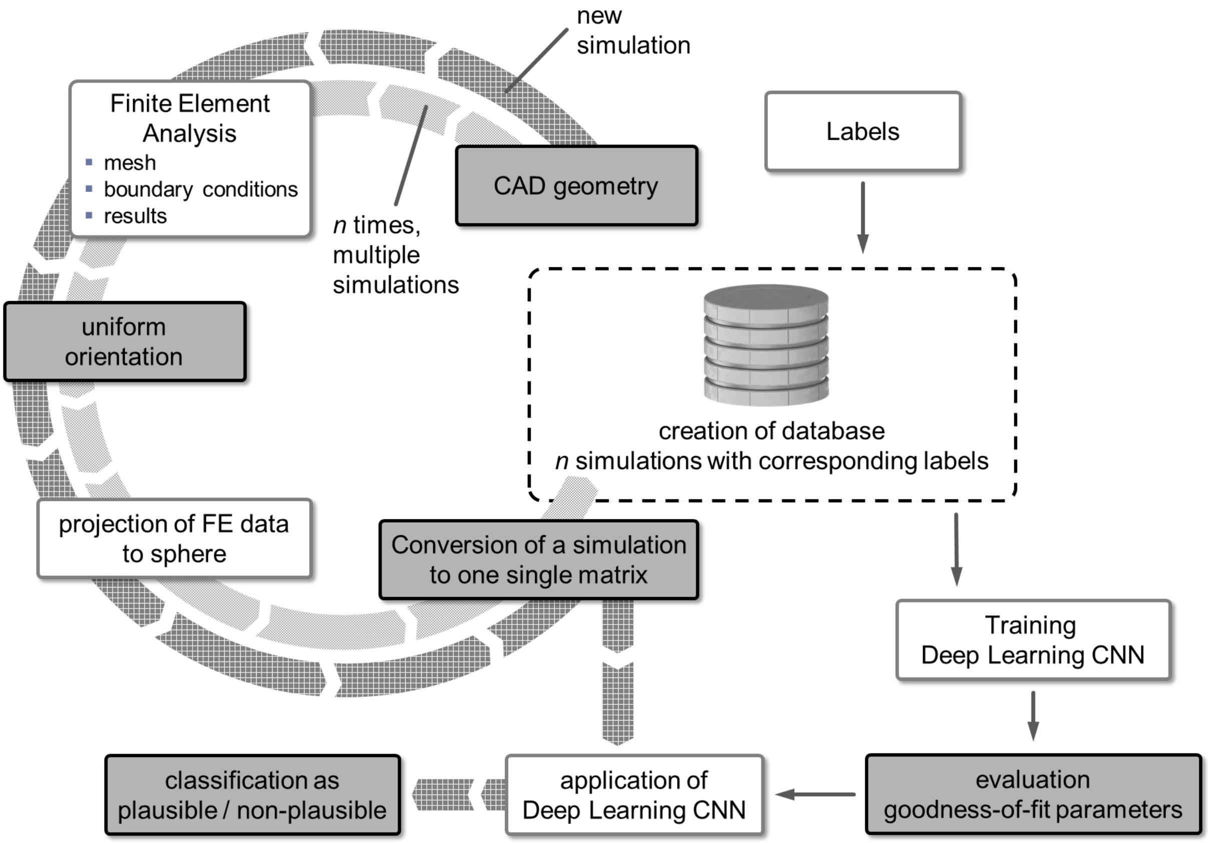

new simulation
Finite Element Analysis
mesh
boundary conditions
results
n times, multiple simulations
CAD geometry
Labels
uniform orientation
creation of database
n simulations with corresponding labels
projection of FE data to sphere
Conversion of a simulation to one single matrix
Training Deep Learning CNN
classification as plausible / non-plausible
application of Deep Learning CNN
evaluation goodness-of-fit parameters

Figure 1. Graphic representation of the methodology for plausibility checks of heterogeneous simulation data

The methodology for plausibility checks for finite element simulations is illustrated in Figure 1. Within the two circles the substeps “CAD geometry”, “Finite Element Analysis”, “uniform orientation”, “projection of FE data to sphere” and “conversion of a simulation to one single matrix” show the solution to the research question: How can heterogeneous FE simulations be used as an input for ANN. Each of these steps will be presented and visualized for a simple block-like part in section 3.1 to 3.5.

3.1 Sub-steps “CAD geometry” and “Finite Element Analysis”

The presented methodology focuses on linear structural mechanic FE simulations of individual parts, consequently assemblies or nonlinear behavior are not considered yet. The start is always the CAD model which is simulated using commercial FE software. After the pre-processing and the processing of the FEA, information about the mesh, the boundary conditions and the simulation results is available. Figure 2 shows the modeled block, including the defined boundary conditions and the dimensions.

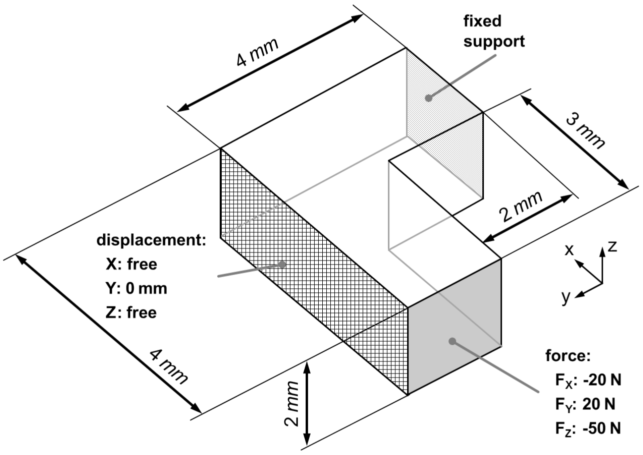


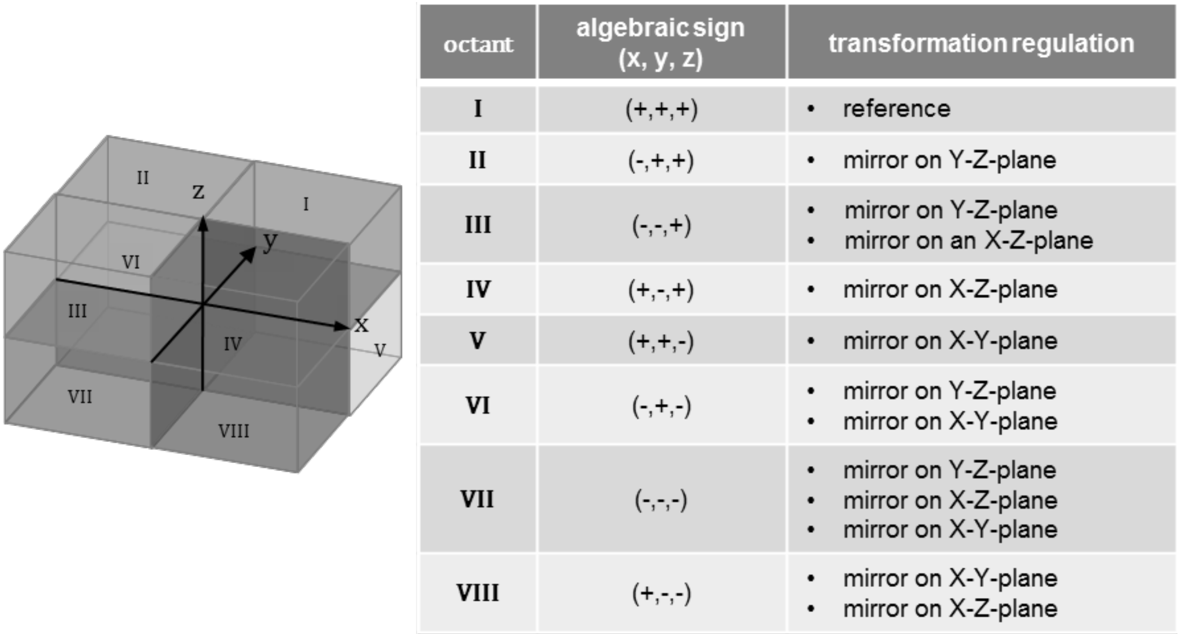
Figure 2. Demonstrator: Simple L-shaped block with FE simulation boundary conditions and dimensions

3.2 Sub-step “uniform orientation”

Similar parts in different FE simulations are not oriented in the same way, but the simulations need to be considered similarly. Therefore, a uniform orientation of the geometry must be extracted from the given point cloud of the FE mesh.

A property of the point cloud is used, which is unique for each initial orientation, since the property is limited to the relative position of the data points to each other. An orientation vector is stretched between the weighted center of a point cloud (the arithmetic mean of all data points in each coordinate direction) and the unweighted center (the median of maximum and minimum value per coordinate direction). The weighted center is often referred to as the center of gravity of the point cloud, because it forms the integral mean of all nodes of the cloud, analogous to the mass center of gravity of a component. The median, also known as the mean value, therefore corresponds to the value in the middle of all nodes and is independent of the node distribution within the two extreme values required for calculation. The point cloud is centered in its center of gravity and transformed into one of the main orientations by means of a principal component analysis. This step is necessary in advance to align the point cloud along one of its main axes. On the one hand, this has the advantage that it always ends visually in a clean alignment and on the other hand, the total number of possible orientations per octant is strongly limited by the PCA; thus, all possible orientations can be covered by the following transformation rules.

Afterwards, for the centered and aligned point cloud, both the weighted center (which must lie exactly in the coordinate origin after centering) and the unweighted center can be calculated. In contrast to the center of gravity, the mean value varies for each starting position and is an indication of the orientation of the point cloud. As a rule, it is not identical to the centre of gravity. By means of the two values just calculated, a statement can be made about which of the eight octants this orientation vector shows. The last step deals with setting up specific transformation rules for each of the octants. For this purpose, one of the octants must be selected as a fixed point: Each of the orientation vectors of the point clouds shall show in these specified octants after the transformation (see Figure 3).



octant	algebraic sign (x, y, z)	transformation regulation
I	(+,+,+)	• reference
II	(-,+,+)	• mirror on Y-Z-plane
III	(-,-,+)	• mirror on Y-Z-plane • mirror on an X-Z-plane
IV	(+,-,+)	• mirror on X-Z-plane
V	(+,+,-)	• mirror on X-Y-plane
VI	(-,+,-)	• mirror on Y-Z-plane • mirror on X-Y-plane
VII	(-,-,-)	• mirror on Y-Z-plane • mirror on X-Z-plane • mirror on X-Y-plane
VIII	(+,-,-)	• mirror on X-Y-plane • mirror on X-Z-plane

Figure 3. Transformation regulation for uniform orientation methodology

3.3 Sub-step "projection of FE data to sphere"

In commercial FE software solid CAD parts are meshed with tetrahedral or hexahedral elements. In Figure 4 the L-shaped block with hexahedral FE mesh is shown. The spherical detector surface with 36x36 pixels (equal detector area per pixel) is shown outside of the block.

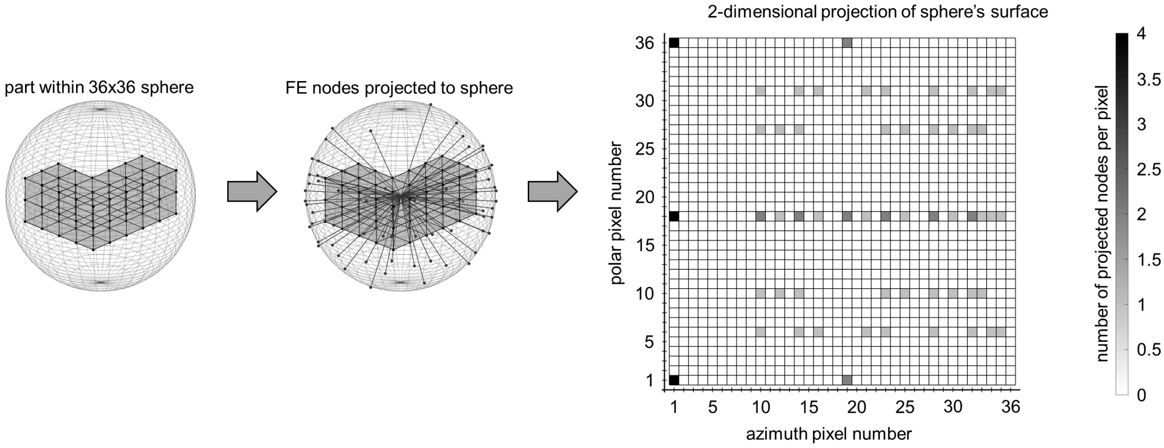


Figure 4. Node Projection of a simple block to the spherical detector surface (36x36 pixels)

The center of gravity of the block is the origin of the detector surface. Therefore, each node of the mesh can be projected onto the surface originating from the center of gravity. Each node is projected to a specific pixel, but multiple nodes can be projected to one pixel. Like a geography map, the surface of the detector sphere can be unfolded to a 2-dimensional matrix. This detector matrix is shown on the right in Figure 4 with the corresponding number of projected nodes to the 36x36 = 1,296 pixels. The projection of the nodes to a specific pixel only depends on the azimuth and polar angle of the spherical detector surface in spherical coordinates. Therefore, the pixels are labeled as polar pixels and azimuth pixels numbered from 1 to 36. Every geometry including parts with undercuts or holes can be projected onto the surface. The projection process is not reversible.

3.4 Sub-step “conversion of a simulation to one single matrix”

After the projection of the nodes to the sphere, for each node one specific pixel is assigned. In FE simulations all relevant information is node-bound, this means that it is known whether a node has fixed constraints due to defined supports or Forces are applied to specific nodes of the FE mesh. Also, all the results (i. e. stresses and deformations) are node-bound. Consequently it is possible to build different matrices for all the mentioned information. For example all the equivalent stresses from the nodes projected to one pixel can be accumulated.

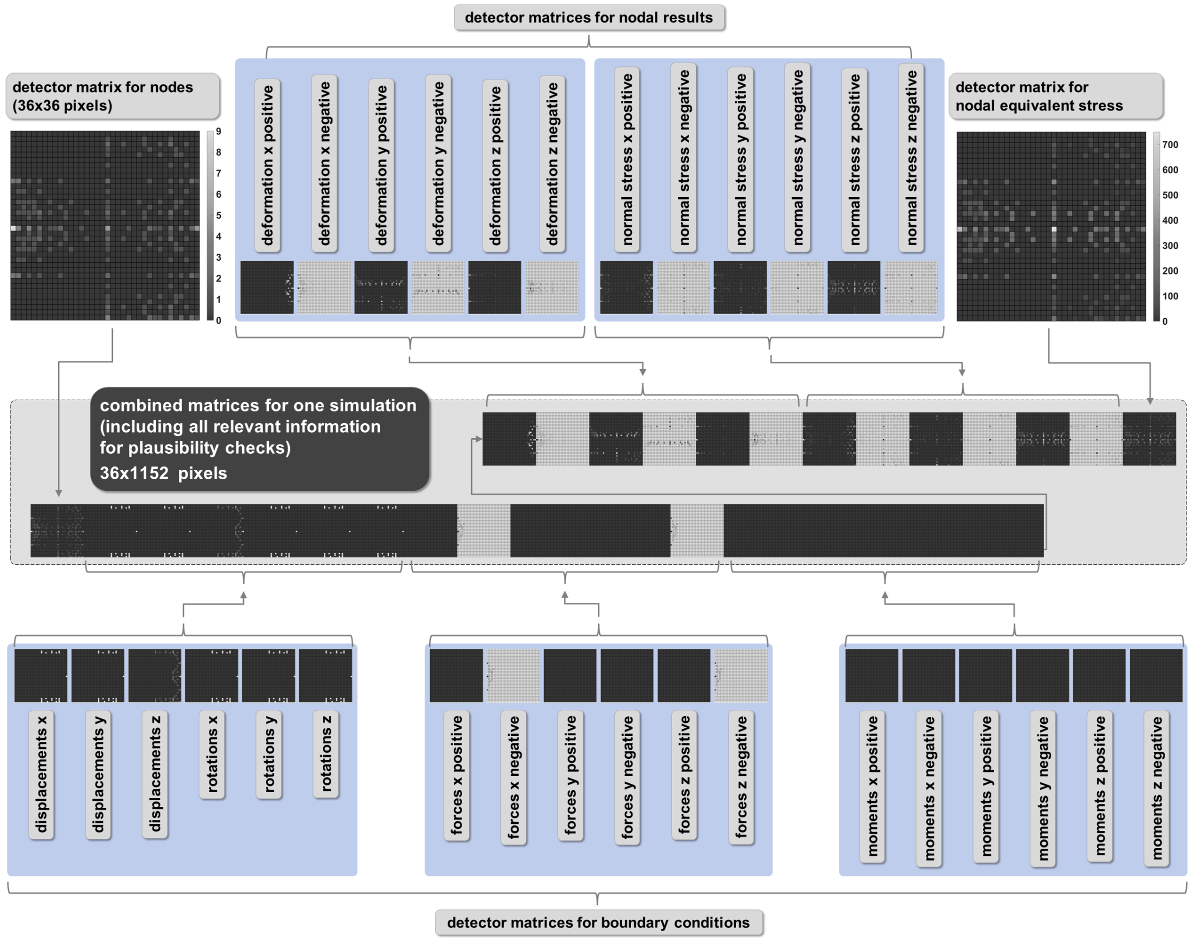


Figure 5. Generated Matrix (“DNA of an FE simulation”) for one FE simulation

In Figure 5 the different matrices for nodes, supports (displacement and rotation), loads (forces, moments) and results (stresses and deformations) can be seen. Each of these matrices has the same size as the number of pixels from the detector surface (i. e. 36x36). All the numerical matrices from one simulation can be combined to form one big numerical matrix (size 36x1,152 pixels), as shown in the middle of Figure 5. As all the values from the different nodes are accumulated, the range in values is very different. For example the equivalent stress matrix normally contains much higher values than the deformation matrix in X-direction. Different Normalization strategies can be applied to transform the numerical matrices in the typical range of -1 to +1. This is important, as most neural networks demand normalized input and output information.

For each of the 36x36 matrices different normalization strategies are applied. The stresses are normalized according to the multiple yield strength and the number of projected nodes at each pixel. Typical min max normalization is not suitable as the same part with fine and coarse meshes would lead to the same normalized matrix. This is not suitable for the presented methodology.

During the creation of the data-base (see Figure 1 light grey circle) each matrix of each simulation must be labeled. For example validated simulations could be labeled "plausible", whereas simulations with multiple iterations could be labeled "non-plausible". In this context, many different options for different labels are imaginable.

3.5 Training, testing and application of a machine learning classifier

For the training and testing of a machine learning classifier, a large data-base of labeled data is mandatory. Large suppliers of the automotive industry perform several thousand FE simulations a day. These simulations can be easily used to form a large database, containing both plausible and non-plausible simulations. In academia, large parameter studies can be used to generate data-set with several ten thousand simulations. The data-base can then be used to train a typical classifier, such as a Convolutional Neural Network (CNN). After the training of the network the performance is evaluated using data-sets that were not used during the training to evaluate the quality of the trained networks. Usually goodness-of-fit parameters such as the „accuracy" (derived from the confusion matrix; Powers, 2011) is calculated. Good CNNs should reach 90% or more. After the training the CNN can be used to classify a new simulation into the categories "plausible" or "non-plausible". In Figure 1 this process is displayed with the dark grey cycle. Starting from the FE information from the post-processing, the mesh is oriented and then the corresponding matrices (nodes, boundary conditions, results) are calculated and transformed and normalized in one big matrix. This matrix represents the whole FE simulation and is the input for the previously trained CNN. According to the input matrix, the simulation is classified into the pre-trained categories (i. e. "plausible" or "non-plausible") by the CNN.

4 Conclusion and outlook

Data-driven product development has large potential to eliminate unnecessary iterations during product development. Already existing simulation data can be used to recognize non-plausible FE simulation. This is essential as FE simulations are not only performed by experienced simulation engineers with many years of expertise. Machine Learning algorithms are very well suited to classify nonlinear data in different categories, but the question arises how to transfer heterogeneous simulation data form different simulations to one classification algorithm. In this paper a methodology is presented that is able to transfer linear structural mechanical FE simulation data to Convolutional Neural Networks. The methodology uses spherical detector surfaces to calculate a specific numerical matrix that represents one simulation. The different sub-steps of the methodology are explained in detail and visualized with a simple geometry.

In the future a large data-base with approximately 100,000 FE simulations of different parts with different boundary conditions will be generated. Afterwards different machine learning algorithms will be applied and adapted for the given data-set. The performance of the classifier will be tested with unknown data-sets. Goodness-of-fit parameters such as the accuracy should reach above 90% to get a reliable classifier.

Acknowledgement

The authors would like to thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Tesla GPU.

This research work is part of the FAU “Advanced Analytics for Production Optimization” project (E|ASY-Opt) and funded by the Bavarian program for the “Investment for growth and jobs” objective finance by the European Regional Development Fund (ERDF), 2014-2020. It is managed by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology. The authors are responsible for the content of this publication.

References

Bengio, Y. (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning.

Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Chearer, C., & Wirth, R. (1999). CRISP-DM 1.0. Step-by-step Data Mining guide. CRISP-DM consortium.

Chen, M.-S.; Han, J., & Yu, P. (1996). Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering Vol. 8, No. 6.

Dayhoff, J., & DeLeo, J. (2001). Artificial Neural Networks. Opening the Black Box. Conference on Prognostic Factors and Staging in Cancer Management: Contributions of Artificial neural Networks and Other Statistical Methods.

Deng, L. (2012). Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. Microsoft Research. Redmond. USA.

Ding, S.; Yin, S.; Zhang, P.; Ding, E., & Naik, A. (2009). An approach to data-driven adaptive residual generator design and implementation. Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes. Spain.

Ding, S. (2012). Data-Driven Design of Fault-Tolerant Control Systems. 8th IFAC Symposium on Fault Detection, Mexico.

Ermel, C.; Gall, J.; Lambers, L. & Taentzer, G. (2011). Modeling with Plausibility Checking: Inspecting Favorable and Critical Signs for Consistency between Control Flow and Functional Behavior. International Conference on Fundamental Approaches to Software Engineering (FASE). Saarbrücken, 26. March – 3. April 2011.Summary of Proceedings, Springer, Berlin. pp. 156-170. DOI 10.1007/978-3-642-19811-3_12.

Kestel, P., Schneyer, T., & Wartzack, S. (2016). Feature-based approach for the automated setup of accurate design-accompanying Finite Element Analyses. In Proceedings of the 14th International Design Conference. Dubrovnik.

Kruse, R.; Borgelt, C.; Braune, C.; Klawonn, F.; Moewes, C., & Steinbrecher, M. (2011). Computational Intelligence. Vieweg+Teubner Verlag, Springer Fachmedien Wiesbaden GmbH.

LeCun, Y.; Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature Review Article.

Müller-Sommer, H. & Straßburger, S. (2010). Methods for the Plausibility Check of Input Data for Intra-Logistic-Simulations. Integration Aspects of Simulation: Equipment, Organization and Personnell. Karlsruhe, 7.-8. October 2010. KIT Scientific Publishing. Karlsruhe. pp. 61-68.

Nestler, A. (2007). Bestimmung der Messunsicherheit für die Verfahren und Methoden zur Bodenanalytik des Anhanges 1 der Bundes-Bodenschutz- und Altlastenverordnung. Dissertation. Freie Universität Berlin.

Powers, D.M.W. (2011). "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation". Journal of Machine Learning Technologies. Vol. 2 No.11. pp. 37-63.

Quin, Z., & Tang, Y. (1995). Uncertainty Modeling for Data Mining. A Label Semantics Approach. Springer.

Qin, Z., & Tang, Y. (2014). Uncertainty Modeling for Data Mining, A Label Semantics Approach. Springer, Berlin Heidelberg. DOI 10.1007/978-3-642-41251-6.

Runkler, T. (2010). Data Mining. Methoden und Algorithmen intelligenter Datenanalyse. Vieweg+Teubner. Wiesbaden.

Sauer, C., Küstner, C., Schleich, B., & Wartzack, S. (2017). Einsatz von Deep Learning zur ortsaufgelösten Beschreibung von Bauteileigenschaften. In Krause, D.; Paetzold, K.; Wartzack, S. (Eds.), Design for X. Beiträge zum 28. DfX-Symposium (pp. 49-60). Bamberg, DE: Hamburg: TuTech Verlag.

Schleich, B.; Anwer, N.; Mathieu, L., & Wartzack, S. (2015). Contact and Mobility Simulation for Mechanical Assemblies Based on Skin Model Shapes. Journal of Computing and Information Science in Engineering, 15(2), 021009-1-021009-7. https://dx.doi.org/10.1115/1.4029051

Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. The Swiss AI Lab IDSIA. Switzerland.

Spruegel, T.; Kestel, P., & Wartzack, S. (2016). FEA-Assistenzsystem – Plausibilitätsprüfung für Finite-Elemente-Simulationen mittels sphärischer Detektorflächen. In Krause D., Paetzold K., Wartzack S. (Eds.), Beiträge zum 27. DfX-Symposium. Jesteburg.

Spruegel, T.C.; Schröppel, T. & Wartzack, S. (2017). Generic Approach to Plausibility Checks for Structural Mechanics with Deep Learning. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 1: Resource-Sensitive Design | Design Research Applications and Case Studies, Vancouver, Canada, 21.-25.08.2017.

Tischler, K. (2013). Informationsfusion für die cooperative Umfeldwahrnehmung vernetzter Fahrzeuge. Dissertation. Karlsruher Institut für Technologie (KIT).

Tröger, K.A. (2009). Analyse der elektrischen Durchschlagmechanismen in syntaktischen Schäumen. Dissertation. RWTH Aachen University.

Yin, S.; Ding, S.; Xie, X., & Luo, H. (2014). A Review on Basic Data-Driven Approaches for Industrial Process Monitoring. IEEE Transactions on industrial electronics, Vol. 61, No. 11.

Zhang, G.; Patuwo, B., & Hu, M. (1997). Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14 (1998) 35–62.

Zhang, Q.; Shariff, R.; Smith, D.; Cudrak, A., & Stanley, S. (2007). Artificial neural network. real-time process control system for small utilities. Journal AWWA.

Zhou, Z.-H. (2002). Three perspectives of Data Mining. Elsevier Science B.V.